Diss. ETH No. 15233

Evaluating Performance in Systems with Heavy-Tailed Input A Quantile-based Approach

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZURICH

for the degree of
DOCTOR OF TECHNICAL SCIENCES

presented by
ULRICH FIEDLER
Dipl. Physiker ETH
born January 12, 1965
citizen of Germany

accepted on the recommendation of
Prof. Dr. B. Plattner, examiner
Dr. M. Maechler, co-examiner
Prof. Dr. P. Huang, co-examiner

2003

Abstract

One of the key invariants in computer and communication systems is that important characteristics follow long- or heavy-tailed distributions. This means that the tail of these distributions declines according to a power law. Hence, the probability for extremely large values is non-negligible. For example, such distributions have been found to describe the size of web objects or the processing latencies in computer and communication systems. As a consequence, there is a need to employ such distributions when evaluating such systems with synthetic workloads. However, sampling from such distributions to generate workloads implies that the system under evaluation remains in transient state over all periods of time that are feasible for performance evaluations. Consequently, frequently-used statistics for performance evaluation, such as the average of the system output, do not converge.

In this thesis we move away from evaluation using statistics such as the average, which describe the expected behavior of the system in all cases, and take the step towards evaluation using statistics such as quantiles, which describe the behavior in a given percentage of cases. Such quantiles of the system output do not depend on the extreme tail of the output distribution. We therefore address the problem of whether employing quantiles can enable performance evaluations within periods of time that are feasible in practice for performance evaluations.

Quantiles have a natural interpretation to statistically characterize the system performance. If e.g. a system offers a web service, the 99-th percentile of the latencies of downloads can statistically characterize the system performance from a user's view, since 99% of downloads terminate within times smaller than this quantile. If converged, such quantiles can be used to derive statistical guarantees for the system performance. Similar statements hold for system components such as servers and networks.

Applying probability theory, we show that statistics of quantiles converge considerably faster than other frequently-used performance evaluation statistics if the underlying distribution is long- or heavy-tailed. Based on this theory, we give a method which enables to evaluate system performance under long- or heavy-tailed input within periods of time that are practically feasible. We validate the proposed method by applying it to a simulation-based evaluation of the network performance of systems that offer web services.

We show that the proposed method has further applications to other problems that require performance evaluation with synthetic workloads which are generated by sampling long- or heavy-tailed distributions. These applications include capacity provisioning, benchmarking of new hard- and software, as well the evaluation of protocols that rely on the request/reply paradigm such as HTTP, IMAP, FTP, or NFS. Further applications can be found in the field of computer systems where CPU requirements of tasks show a heavy-tailed distribution. This includes the evaluation of migration policies in a network of workstations, as well as task assignment policies for distributed servers.

Kurzfassung

Die Langschwänzigkeit der Verteilungen wichtiger Kenngrössen ist eine der bekannten Invarianten in Computer- und Kommunikationssystemen. Langschwänzigkeit einer Verteilung bedeutet, dass die Verteilung nach einem Potenzgesetz zerfällt. Dies wiederum hat zur Folge, dass die Wahrscheinlichkeit sehr grosse Werte zu beobachten nicht vernachlässigbar ist. Unter anderem wurde festgestellt, dass z.B. die Grösse von Web-Objekten oder die Verarbeitungslatenz gewisser Systeme mit langschwänzigen Verteilungen beschrieben werden können. Daher gibt es bei der Evaluation von Systemen mit synthetischer Last ein Bedürfnis solche Verteilungen zu modellieren. Allerdings impliziert das Generieren von Last durch ein Abtasten solcher Verteilungen, dass das zu evaluierende System sich in einem transienten Zustand befindet während aller Zeitspannen, die für Performance-Evaluationen machbar sind. Als Folge konvertieren gebräuchliche Statistiken zur Performancemessung, wie der Mittelwert von Outputgrössen, nicht.

In dieser Dissertation bewegen wir uns weg von Performance-Evaluationen mit Statistiken, wie dem Mittelwert, der das erwartete Verhalten des Systems in allen Fällen wiedergibt, und betrachten Statistiken wie Quantile, die das Verhalten des Systems in einem gegebenen Prozentsatz von Fällen wiedergeben. Solche Quantile der Outputgrössen hängen nicht vom äussersten Schwanz der Output-Verteilung ab. Deshalb untersuchen wir das Problem, ob das Betrachten von Quantilen eine Performance-Evaluation innerhalb von Zeiten ermöglicht, die für solche Evaluationen praktisch machbar sind.

Quantile haben eine natürliche Interpretation um die Performance von Systemen zu charakterisieren. Wenn z.B. ein System einen Web-Dienst anbietet, kann das 99-te Perzentil der Download-Latenz die System Performance aus

Benutzersicht charakterisieren, da 99% der Downloads innerhalb einer Zeit enden, die kleiner als dieses Quantil ist. Falls ein solches Quantil konvergiert hat, kann es benutzt werden, um statistische Garantien für die Performance des Systems abzugeben. Aehnliche Aussagen können für Systemkomponenten, wie Server und Netzwerke, gemacht werden.

Durch Anwendung von Wahrscheinlichkeitstheorie zeigen wir, dass Statistiken wie Quantile signifikant schneller als andere bei der Performance-Evaluation gebräuchliche Statistiken konvergieren, wenn die unterliegende Verteilung langschwänzig ist. Basierend auf dieser Theorie geben wir eine Methode an, die es ermöglicht, die System Performance unter langschwänzigem Input innerhalb von Zeiten zu evaluieren, die praktisch machbar sind. Wir validieren die vorgeschlagene Methode, indem wir sie auf eine simulationsbasierte Performance-Evaluation von Netzwerken in Systemen, die Web-Dienste anbieten, anwenden.

Wir zeigen, dass diese Evaluationsmethode weitere Anwendungen bei Problemen hat, die eine Performance-Evaluation mittels synthetischer Last erfordern, welche durch Abtasten von langschwänzigen Verteilungen generiert wird. Die Anwendungen beinhalten die Dimensionierung von Kapazitäten, das Benchmarking neuer Hard- und Software, sowie die Evaluation von Protokollen die auf dem Request/Reply Paradigma beruhen, wie HTTP, IMAP, FTP, oder NFS. Weitere Anwendungen können im Gebiet der Computersysteme gefunden werden, da der CPU-Bedarf von Tasks ebenfalls langschwänzig verteilt ist. Dies beinhaltet die Evaluation von 'Migration Policies' in einem Netzwerk von Workstations sowie von 'Task Assigment Policies' für verteilte Server.

Contents

1 Introduction **1**

1.1 Problem Statement 4

1.2 Our Approach 6

1.3 Validation 8

1.4 Research Contributions 9

1.5 Outline 10

2 Background and Related Work **13**

2.1 Probability Distributions 14

2.1.1 Moments of a Distribution 16

2.1.2 Confidence Interval and Accuracy 16

2.1.3 Exponential and Heavy-tailed Distributions 17

2.1.4 Pareto Distributions 20

2.2 Modeling Heavy Tails 21

2.2.1 Heavy Tails in Network Traffic Generation 21

2.3 Implications of Heavy-tailed Input 26

2.4 Performance Evaluations with Heavy-Tailed Input 30

2.4.1 Network and Server Performance 30

2.4.2 Task Scheduling in Operating Systems 33

2.5 Performance Evaluation with Quantiles 34

2.6 Summary 34

3 Theory 37

3.1 Definition of Quantiles 37

3.2 Distribution of a Sample's Quantile 40

3.3 The Impact of Correlations on Convergence 42

3.3.1 Convergence of Quantiles 47

3.4 Summary 51

4 A Method for Quantile-based Evaluation 53

4.1 Estimation of Minimal Sample Sizes 56

4.1.1 Evaluating Convergence of Input Quantiles 57

4.1.2 Large Sample Approximation 59

4.1.3 Special Cases 60

4.2 Evaluating Quantiles in System Output 60

4.2.1 Visually Assessing Convergence 61

4.2.2 Testing for Convergence 62

4.2.3 Inferring Estimates of Quantiles 63

4.2.4 Rate of Convergence 63

4.3 Generalizations of the Method 64

4.4 Summary 66

5 Simulation Environment 67

5.1 Performing Simulations of Web Downloads 67

5.2 ns-2 69

5.2.1 TCP Support 70

5.2.2 HTTP support 71

5.2.3 A Short Review of HTTP/1.1's Connection Management 72

5.3 Our Implementation of Web Downloads in ns-2 75

5.3.1 Resource-related Input Variables 76

5.3.2 User-related Input Variables 77

5.3.3 Protocol-related Input Variables 77

5.3.4 Our Implementation of HTTP 78

5.3.5 Connection Setup 80

5.3.6 Interface for Testing 80

5.4 Summary 81

6 Evaluation of Simulation Input 83

6.1 Properties of the Workload Generation Process 84

6.2 Object Size Distribution 85

6.3 Object Size Quantiles 86

6.3.1 Sample Sizes to Converge Object Size Quantiles . . . 86

6.3.2 Sample Sizes to Converge the Running Mean 88

6.3.3 Introducing System Limits 89

6.3.4 Sensitivity to Small Changes in Tail Index 91

6.4 Special Cases 92

6.5 Summary 92

7 Evaluation of Simulation Output 95

7.1 Simulation Study 95

7.1.1 Workload Generation 95

7.1.2 Topology 97

7.1.3 Simulation Duration 98

7.1.4 Limitations 99

7.1.5 Evaluation of Network Latency and Latency Quantiles 99

7.2 Results 100

7.2.1 Low Utilization 100

7.2.2 High Utilization 106

7.2.3 Medium Utilization 110

7.3 Summary 112

8 Applications 115

8.1 Network Performance Evaluation 117

8.2 Server Performance Evaluation 118

8.3 Computer System Performance Evaluation 119

8.4 Outside Computer and Communication Systems 119

8.5 Summary 120

9 Conclusions and Further Work 121

9.1 Review of Contributions 122

9.2 Further Work 125

A E-mail Exchange with Jan Beran 139

B Examining Normal Plots 143

C Normality Tests 149

D Details of Simulation Evaluation 153

List of Figures

1.1 System Model 6

2.1 Distribution of a Random Variable 14

2.2 CCDF of a Exponential and a Pareto Distribution 20

2.3 Levels of Activity 22

2.4 Self-similarity of Web Traffic 23

2.5 Convergence to Normality vs. Convergence to α-stable (from [Crovella and Lipsky(2000)]) 28

3.1 Relation CDF and Quantile 38

3.2 Sample's CDF 39

4.1 System Model 54

4.2 CCDF of Samples from a Heavy-Tailed Distribution 65

5.1 System Model of the Simulation 68

5.2 Interactions of Download with HTTP/1.0 73

5.3 Interactions of Download with HTTP/1.1 74

5.4 Method Invocations of a Download (Example) 78

5.5 Visualizing Connection Management 80

7.1 Validation Topology 97

7.2 Normal Plot (99-th NLP, Coarse Model, Low Util.) 101

7.3 Normality Test (99-th NLP, Coarse Model, Low Util.) 102

7.4 Consistency of Convergence (Coarse Model, Low Util.) 103

7.5 Normality Test (99-th NLP, Accurate Model, Low Util.) 105

7.6 Consistency of Convergence (Accurate Model, Low Util.) . . . 105

7.7 Normality Test (Coarse Model, High Util.) 107

7.8 Consistency of Convergence (Coarse Model, High Util.) . . . 108

7.9 Estimation of H and Sigma (Coarse Model, High Util.) 110

7.10 Normal Plots for the 99-th NLP (Coarse Workload Model, 2560Kb/s) 111

B.1 Example: Normal Plot for Data with Outliers [statguide(1997)] 143

B.2 Normal Plots: Possible Deviations from Normality [statguide(1997)] 144

D.1 Convergence of NLPs, OSPs and avg. NL, avg OS (Coarse Model, Low Util.) 154

D.2 Consistency of Convergence (Coarse Model, Low Utilization) 155

D.3 Convergence of NLPs, OSPs and avg. NL, avg OS (Coarse Model, Medium Utilization) 156

D.4 Consistency of Convergence (Coarse Model, Medium Utilization) 157

D.5 Convergence of NLPs, OSPs and avg. NL, avg OS (Coarse Model, High Utilization) 158

D.6 Consistency of Convergence (Coarse Model, High Utilization) 159

D.7 Convergence of NLPs, OSPs and avg. NL, avg OS (Accurate Model, Low Util.) . . . 160

D.8 Consistency of NLPs, OSPs (Accurate Model, Low Util.) . . . 161

D.9 Object Size / Network Latency Relation (Coarse Model) . . . 161

D.10 Histogram of Network Latencies around the 99-th NLP (Coarse Model) . . . 162

D.11 Normal Plots for the 99-th NLP (Coarse Workload Model, 2560Kb/s) . . . 163

List of Tables

2.1 Required Sample Size to Estimate the Average from the Running Mean [Crovella and Lipsky(2000)] 30

4.1 Critical Values for Normality Test [Rice(1995)] 63

5.1 Resource Related Parameters to Generate Web Traffic 76

5.2 User Related Random Variables to Generate Web Traffic . . . 77

6.1 Parameters for the ParetoII Object Size Distribution 85

6.2 Percentiles of the Object Size Distribution 86

6.3 Sample Size Required to Estimate the 99-th Object Size Percentiles 88

6.4 Sample Size Required to Estimate further Object Size Percentiles 88

6.5 Sample Size Required to Estimate the Average Object Size (adapted from [Crovella and Lipsky(2000)]) 89

6.6 Sample Size Required to Estimate the Average Object Size (2.1GB limit) 91

6.7 Sample Sizes at Various Tail Indices α (5% Accuracy) 91

7.1 Probability Distributions to Generate Web Traffic 96

7.2 Avg. Utilization of the Bottleneck Link 97

7.3 Sample Size Required to Estimate the 99-th NLP (Coarse Model, Low Util.) 104

7.4 Sample Size Required to Estimate the 99-th NLP (Accurate Model, Low Util.) 106

7.5 Approx. Sample Size Required to Estimate the 99-th NLP (Coarse Model, High Util.) 110

D.1 Estimated NLPs (Coarse Model, Low Utilization, 500k DLs) . 153

D.2 Estimated NLPs (Coarse Model, High Utilization, 500k DLs) . 153

D.3 Estimated NLPs (Accurate Model, Low Utilization, 120k DLs) 155

D.4 Population around 99-th NLP (Coarse Model, 30x150k DLs) . 155

Chapter 1

Introduction

One of the key invariants in computer and communication systems is that important characteristics follow long- or heavy-tailed distributions. This means that the tail of these distributions declines according to a power law. Hence, the probability for extremely large values is non-negligible. For example, such distributions have been found to describe the size of web objects or the processing latencies in computer and communication systems. As a consequence, there is a need to employ such distributions when evaluating such systems with synthetic workloads. However, sampling from such distributions to generate workloads implies that the system under evaluation remains in transient state over all periods of time that are feasible for performance evaluations. Consequently, frequently-used statistics for performance evaluation, such as the average of the system output, do not converge.

In this thesis we move away from evaluation using statistics such as the average, which describe the expected behavior of the system in all cases, and take the step towards evaluation using statistics such as quantiles, which describe the behavior in a given percentage of cases. Such quantiles of system output do not depend on the extreme tail of the output distribution. We therefore address the problem of whether employing quantiles can enable performance evaluations within periods of time that are practically feasible for performance evaluations.

We illustrate the problem and our approach to a solution with an example of a capacity provisioning problem for web services. The infrastructure for these services typically consists of a network, servers, and clients. The service is engaged by triggering a browser at the client to download a web page. The browser therefore requests web objects from servers over the network. Upon arrival, these objects are displayed in the browser.

The main property of the traffic of web services in the context of performance evaluation is the stochastic self-similarity of traffic patterns. This means that the burstiness of traffic patterns is preserved on different time scales which span at least four to five magnitudes [Park and Willinger(2000)]. Such self-similarity is known to have significant negative impact on performance and stability in performance evaluations. A related property is the long-tailed distribution of the size of downloaded objects. This long-tailed distribution implies a very large variability of object sizes. Thus, extremely large object sizes occur with significant probability in addition to many small objects.

To evaluate the performance of networks and servers for capacity provisioning, benchmarking, or protocol evaluation, it is common practice to employ synthetically generated workloads rather than to replay traffic traces [Krishnamurthy and Rexford(2001)]. The main reasons for this are that traffic traces may not reflect the full variability inherent to web traffic and that traffic traces cannot account for effects caused by load adaptation mechanisms such as TCP's flow control in the network. Moreover, traffic traces do not account for future trends in workload developments. The object sizes in the synthetic workload generation are usually obtained by sampling long- or heavy-tailed distributions. It can be shown that generating workload by sampling a heavy-tailed object size distribution induces a self-similarity in the generated traffic which is comparable to the self-similarity of web traffic observed in existing infrastructure [Willinger et al.(1995)]. As a consequence such traffic has similar performance and stability properties.

[Crovella and Lipsky(2000)] states that performance evaluations are difficult when workload is generated by sampling a heavy-tailed distribution. Such workloads cause the evaluated system to remain in a transient state for all times that are practically feasible for performance evaluations. For the performance evaluation of web services, this implies that frequently used statistics such as the average download latency do not converge. Thus, these statistics are not suitable for performance evaluation.

A similar statement can be made for capacity provisioning, benchmarking, or protocol evaluation of any service that employs request/reply transactions or downloads of objects that have a long- or heavy-tailed distribution. Examples for such services are e-mail and file transfer which base on protocols like IMAP, FTP, and NFS.

Clearly, there is a need to research whether meaningful statistics can be found in order to statistically evaluate the performance of services within times that are of practical interest. This thesis proposes a quantile-based approach, which implies that the evaluation is restricted to account for a large fraction of downloads instead of all downloads. For the example of performance evaluation of a web service, this means that we propose to evaluate network and server performance with the statistics of download latency percentiles such as the 99-th latency percentile. By definition 99% of the downloads have a latency smaller than the

99-th latency percentile. Thus this percentile has a natural interpretation in statistical evaluation of quality of service. Performance evaluation with such latency quantiles does not account for the latencies of extremely large downloads. Therefore performance evaluation with such statistics can lead to meaningful results within periods of time that are practically feasible.

To validate this hypothesis, we show that a quantile-based approach can be employed to evaluate the network performance in simulations of web services. Results presented in this thesis indicate that network latency quantiles, such as the 99-th latency quantile, converge during periods of time that are practically feasible. This in turn leads to useful quality of service guarantees. We further show that a similar convergence can be expected for other systems when input for performance evaluation is generated by sampling from long- or heavy-tailed distributions. Therefore it is possible to give statistical quality of service guarantees in a number of resource allocation problems.

1.1 Problem Statement

For the example of performance evaluation of a web service with a synthetic workload the research problem that this thesis addresses can be stated as follows:

We assume that object sizes for the performance evaluation are generated by sampling a heavy-tailed distribution. This generally prevents frequently-used statistics such as the average download latency from converging at sample sizes that are of practical use to engineer quality of service guarantees. A similar statement can be made for other statistics that depend on objects of all size. Hence, there is a need to research *whether it is possible to evaluate the performance of the service with statistics that* *do not depend on objects of all size*. These statistics then need to have the following properties:

1. These statistics need to have a meaningful interpretation in the context of quality of service.
2. These statistics need to converge at sample sizes that are of practical use to evaluate quality of service.

The reason that statistics that depend on objects of all size do not converge at sample sizes that are of practical use can be given as follows: In order for the object size distribution in the sample to converge to the heavy-tailed distribution used for generation of the objects, a necessary condition is that the running average of the generated object sizes converges to the average of the heavy-tailed distribution. This requires sample sizes larger than 10^{14} if we assume convergence to a 5% relative accuracy and heavy-tailed object size distribution with $\alpha = 1.1$. This large sample size can be explained with convergence properties of the sample's average from a heavy-tailed distribution that has infinite variance. This average does not converge to a normal distribution at the typical $n^{-1/2}$ rate, where n is the sample size. Instead, the average converges to a α-stable distribution at a rate which is considerably slowed down for α between 1.0 and 1.2, which is typical for measured object size distributions. Truncating the object size distribution at 2.1GB, which is equivalent to assuming that object sizes are represented by signed 32 bit integers since $2^{31} = 2.1 \cdot 10^9$, reduces the sample size to 10^{12} objects. This is to be explained with the very large variance of the truncated heavy-tailed distribution.

1.2 Our Approach

In this thesis, we abstract from the performance evaluation of a web service with synthetic workload. We introduce a system model which is depicted in Figure 1.1 and propose a quantile-based method to enable performance evaluation of systems that follow this model at amounts of input that are of practical use.

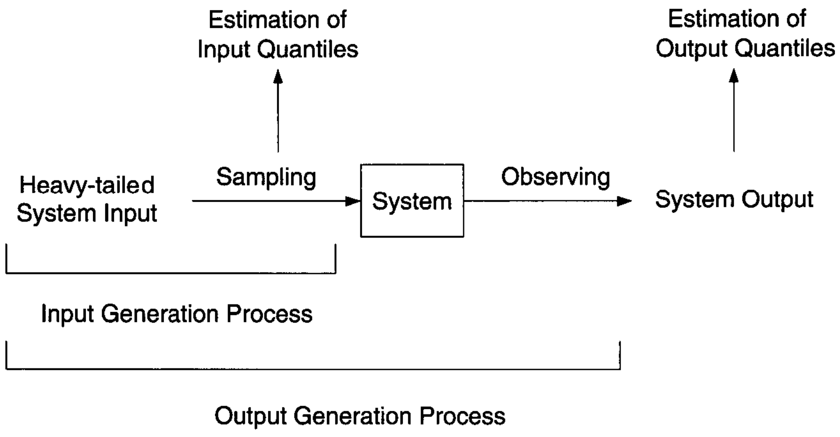

Estimation of Input Quantiles
Estimation of Output Quantiles
Heavy-tailed System Input
Sampling
System
Observing
System Output
Input Generation Process
Output Generation Process

Figure 1.1: *System Model*

For the example of web services, the long- or heavy-tailed distribution in the input generation process is the size distribution of downloaded objects. The observed system output is the latency of the downloads. The system consists of the infrastructure for which we evaluate performance, i.e. network, servers, etc. Alternatively, simulations which mimic the infrastructure can be used for performance evaluation.

We assume that system performance can be characterized with converged quantiles of system output, such as the 95-th, 98-th, or 99-th percentile. For the example of web services this means assuming that the performance of a web service can be characterized with converged 95-th, 98-th, or 99-th percentiles of user-perceived download latency. These percentiles reflect the amount of time required to complete the corresponding percentage of web downloads and therefore naturally reflect a statistical quality of a web service when converged.

We assume that a lower bound for the amount of input necessary to converge a quantile in system output can be estimated from the convergence of the corresponding quantile of system input. This lower bound estimates the initial phase of the convergence of the output quantile. Due to effects that come from the adaptivity of the system, much more of the input may need to be consumed to converge the quantile in system output.

Hence, the method is based on estimation of quantiles such as the 95-th, 98-th, or 99-th percentile in system in- and output. The method exploits the fact that quantiles from a heavy-tailed distribution usually converge at sample sizes that are of practical use. For the example of a web service, the 99-th object size quantile converges to a 5% relative accuracy at sample sizes larger than 10^5 if the object sizes are obtained by sampling a heavy-tailed object size distribution with $\alpha = 1.1$. This comes from the fact that quantiles of a heavy-tailed distribution converge to a normal distribution at a $n^{-1/2}$ rate, where n is the sample size. This normal distribution usually has a comparably small variance. The convergence solely depends on the order of the quantile and the probability density of the heavy-tailed distribution in the vicinity of the quantile. This convergence is fundamentally different from the convergence of the running average in the sample from the heavy-tailed distribution. The running average converges to an α-stable distribution at a rate which is significantly slower than $n^{-1/2}$. As a consequence of this difference in convergence, quantiles of input from a heavy-tailed distribution can be estimated at sample sizes that are of practical use. This means that the corresponding quantile in system output may also converge at sample sizes that are of practical use.

Assuming that the system can contain adaptive mechanisms that prevent an evaluation with simple input/output correlation schemes, we propose to apply convergence test procedures to estimate quantiles of system output. Inferences from probability theory suggest that quantiles in system output generally converge to a normal distribution whenever the output distribution is sufficiently regular in the vicinity of the quantile. The details of this convergence depend on the correlation structure of system output. If the correlations decay fast enough to be consistent with a weakly dependent correlation structure, quantiles generally converge to a normal distribution at a $n^{-1/2}$ rate [Hampel et al.(1986)]. Stronger correlations result in a long-range dependent correlation structure, which typically manifests in a slow down of the rate of convergence [Beran(1994)]. We thus propose to apply standard normality test procedures which are based on normal probability plots to check whether the p-th quantile in system output is consistent with a normal distribution. In case of convergence, the test additionally checks the rate of convergence and provides an accurate estimation of the quantile.

1.3 Validation

We validate the proposed method with a simulation study of the network performance of a web service using ns-2 [ns-2(2000)]. We derive the convergence of the 95-th, 98-th, and 99-th percentile in simulation input. We employ this convergence to determine the minimal number of downloaded objects required to estimate the corresponding 95-th, 98-th, and 99-th network latency percentile. We then test the convergence of these latency quantiles. We find that in our simulations under low utilizations, quantiles of interest are weakly correlated and converge to a normal distribution at a $n^{-1/2}$ rate. With low utilization, we mean utilizations that are comparable to what Odlyzko's study of utilization patterns [Odlyzko(2000)] reports as average in private networks. Under high utilizations, we also find convergence of latency quantiles to a normal distribution. This convergence is slower than a $n^{-1/2}$ rate. With high utilization, we mean utilizations that are comparable to what France Telecom and others consider as a maximum which is acceptable during the busiest period [Ben Fredj et al.(2001)]. In both cases we can accurately estimate confidence intervals for the latency quantiles that can be employed to engineer guarantees for the web service.

Finally, we argue that both the estimation of the minimal sample size to converge the p-th quantile in system output as well as the test method can be applied to evaluate system output of other systems with heavy-tailed input.

1.4 Research Contributions

The research contribution of this thesis is to provide a method that enables performance evaluation of systems with synthetic workloads that were generated by sampling heavy-tailed distributions. In detail this means

1. **Quantiles are suitable statistics for performance evaluation**
   We give evidence that quantiles are suitable statistics for performance evaluation of systems with synthetic workloads that were generated by sampling heavy-tailed distributions.
2. **A priori bounds for evaluation duration**
   We provide lower bounds that estimate the initial phase in the convergence of quantiles in system output.

3. **Estimation of quantiles in system output**
   We give a method to test whether quantiles of simulation output have converged. In case of convergence this method additionally provides accurate estimates for the quantiles.

4. **Practicability of the method**
   We show that the test method can be employed to evaluate the network performance of web services in terms of latency quantiles.

5. **Versatility of the method**
   We show that our method is not limited to the performance evaluation of network performance of web services. Instead, the method has a series of further applications which need not be related to the evaluation of network performance of web services.

These research contributions are revisited and assessed in Chapter 9.

1.5 Outline

This thesis is structured as follows:

- **Chapter 2** gives backgrounds of probability theory. The chapter explains why it is important to model long- and heavy-tails in performance evaluations in the generation of synthetical workloads and why this leads to a stability problem. Finally, the chapter reviews how this stability problem is addressed in related work.

- **Chapter 3** introduces the theory behind our quantile-based method.

- **Chapter 4** introduces our quantile-based method for performance evaluation, which is comprised of estimation of minimal sample sizes required to converge the p-th quantile in system output and testing this quantile for convergence.
- **Chapter 5** describes the simulation environment of ns-2 which we have employed to study network latencies of web downloads to validate our method, and includes a description of our HTTP implementation.
- **Chapter 6** introduces the validation study and evaluates minimal sample sizes required to converge the p-th latency quantile in a simulation of a web service.
- **Chapter 7** elaborates on the validation study and applies the proposed test method to download latency quantiles in a simulation of a web service.
- **Chapter 8** presents further application scenarios and protocols to which our evaluation method can be applied.
- **Chapter 9** concludes this thesis with a summary of results, reviews our research contributions, and provides a number of starting points for further research.

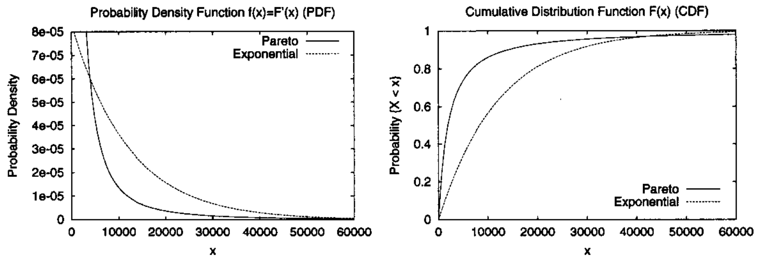
Chapter 2

Background and Related Work

In this chapter we review the background and related work for this thesis. We start with statistics and introduce the notion of long- and heavy-tailed distributions. We explain the long memory property of long- or heavy-tailed distributed characteristics. We explain why it is important to model this property when performance is evaluated with synthetical workload. We illustrate this with the example of performance evaluations with self-similar network traffic. We explain why performance evaluations of systems with synthetical workloads which are generated by sampling long- or heavy-tailed distributions inherently suffer from a stability problem. Finally, we review how this stability or convergence problem is addressed in research work.

2.1 Probability Distributions

A *random variable* X is described by the probability p that certain values x can be observed. The set of values is called the *sample space*.



(a) Probability Density Function

(b) Cumulative Distribution Function

Figure 2.1: *Distribution of a Random Variable*

A *discrete random variable* is a random variable which can take a finite or countably infinite number of values. Examples for a discrete random variable are the number of downloaded web objects in a simulation or the number of currently active users in a simulation. A discrete random variable has a certain probability to take the value x. This probability is denoted by the *probability mass function PMF* $p(x_i) = p(X = x_i)$ for which

$$\sum_i p(x_i) = 1$$

.

A *continuous random variable* is a random variable which can take any real number. An example for a continuous random variable that is related to the topic of this thesis is the download time of a web object. For a continuous random variable, the role of the probability mass function is taken by a *probability density function* (PDF), $f(x)$, which has the properties that $f(x) \geq 0$, f is piecewise continuous, and

$$\int_{-\infty}^{\infty} f(x)dx = 1 \tag{2.1}$$

Then for any $a < b$ the probability that X falls in the interval $]a, b[$ is given by

$$p(a < X < b) = \int_{a}^{b} f(x)dx \tag{2.2}$$

For examples for PDFs see Figure 2.1 (a).

The probability $p(X \leq x)$ is denoted by the *cumulative distribution function (CDF)*

$$F(x) = \int_{-\infty}^{x} f(u)du \tag{2.3}$$

It follows that

$$\lim_{x \rightarrow -\infty} F(x) = 0 \tag{2.4}$$

$$\lim_{x \rightarrow \infty} F(x) = 1 \tag{2.5}$$

and

$$p(a < x < b) = F(b) - F(a) \tag{2.6}$$

For an example for CDFs see Figure 2.1 (b).

2.1.1 Moments of a Distribution

The properties of a distribution of a continuous random variable can be described with moments of the distribution. The *r-th moment* of a distribution is defined as

$$E(X^r) = \int_0^\infty x^r f(x)dx \tag{2.7}$$

Examples:
The first moment of a distribution is the *expected value* $E(X)$.
The *r-th central moment* of distribution is defined as

$$E([X - E(X)]^r) = \int_0^\infty (x - E(X))^r f(x)dx \tag{2.8}$$

Examples:
The second central moment of a distribution is the *variance of the distribution* $Var(X)$, the third central moment is the skewness, and the fourth central moment the kurtosis.

Analog definitions can be made for the distribution of discrete random variables by replacing integrals with sums.

2.1.2 Confidence Interval and Accuracy

A *confidence interval* for a random variable is a interval $[L, U]$ that contains the variable with a specified probability. We call this probability the significance level of the confidence interval. For example, let X be a random variable that follows a normal distribution with expectation value μ and variance σ^2 which is given by

$$\mathcal{N}(\mu, \sigma)(x) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{x} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dt \tag{2.9}$$

Then the confidence interval at significance level 95%, which is symmetric around the expected value μ, is given by $[U, L]$ with $U = \mu - 1.96 * \sigma$ and $L = \mu + 1.96 * \sigma$. Unless explicitly stated otherwise, all confidence intervals in this thesis assume a significance level of 95%.

The relative *accuracy* with which we can estimate a random variable from observations that follow a given distribution can now be defined as

$$Accuracy = max\{|\frac{L-\mu}{\mu}|, |\frac{U-\mu}{\mu}|\} \qquad (2.10)$$

2.1.3 Exponential and Heavy-tailed Distributions

We now introduce distributions that are frequently used when modeling computer- and communication systems.

We begin with the *exponential distribution*. The PDF f of a exponential distribution is given with:

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ \frac{1}{a} e^{-\frac{x}{a}} & \text{if } x \geq 0 \end{cases} \qquad (2.11)$$

The CDF is given by integration. For an example see Figure 2.1.

The exponential distribution has the following properties:

1. The expected value is $E(x) = a$. The variance is $Var(x) = a^2$ which is comparably small. This leads to a standard deviation which equals the expected value.
2. The probability that a exponentially distributed random variable takes values which are magnitudes larger than the expected value is negligible. Neglecting a very small fraction ($< 1\%$) of the very largest values when sampling the

distribution has negligible impact on the statistical properties of the sample.

3. The exponential distribution is the only memoryless random distribution. Thus if for example the age of processes were exponentially distributed, the remaining lifetime of a process is independent of its current age since $e^{a+b} = e^a * e^b$.

Examples for variables in the field of computer and communication systems that are usually modeled with exponential distributions are:

- the session inter-arrival times in web workload generation [Krishnamurthy and Rexford(2001)]
- the arrival rate of tasks in a computer systems [Harchol-Balter et al.(1999)].

Next we introduce long- and heavy-tailed distributions. We follow [Crovella and Lipsky(2000)] and define *heavy-tailed distributions* with

$$1 - F(x) \sim x^{-\alpha} \quad \alpha \in]0, 2] \tag{2.12}$$

where $a(x) \sim b(x)$ means

$$\lim_{x \to \infty} \frac{a(x)}{b(x)} = 1$$

We call α the tail index. We note that more general definitions are possible (see e.g. [Goldie and Kluppelberg(1997)]). For an example see the Pareto distribution in Figure 2.1.

A heavy-tailed distribution has the following properties:

1. For $\alpha \leq 1$, the expected value does not exist. Therefore this thesis assumes $\alpha \in]1, 2]$ unless explicitly stated otherwise. For all $\alpha \in]0, 2]$ the variance is infinite, i.e. the

second moment does not exist. When sampling from such a distribution, the variance of the sample grows without limit.

2. The probability that a heavy-tailed distributed random variable takes values which are magnitudes larger than the expected value is not negligible. Neglecting a very small fraction ($< 1\%$) of the very largest values when sampling the distribution has significant impact on the statistical properties of the sample. The tail of a heavy-tailed distribution manifests a straight line in a log-log plot of the *complementary cumulative distribution function* (CCDF) $1 - F$ (see Figure 2.2).

3. A heavy-tailed distribution has long memory. Thus if age of processes were heavy-tailed distributed, old age of a process would imply that a large remaining lifetime is to be expected. If e.g. the tail index of the heavy-tailed distribution were 1, a process of age x seconds had a probability $\frac{1}{2}$ that the remaining lifetime is more than another x seconds. In other words, the median remaining lifetime of the process is equal to the current age.

Examples for variables in the field of computer and communication systems that are usually modeled with heavy-tailed distributions are:

- the object size and number of embedded images in web workload generation [Krishnamurthy and Rexford(2001)]
- the CPU requirements in a computer systems [Harchol-Balter et al.(1999)].

We define a *long-tailed distribution* as a heavy-tailed distribution which is truncated several orders of magnitude beyond the expected value. Thus long-tailed distributions inherit most of the

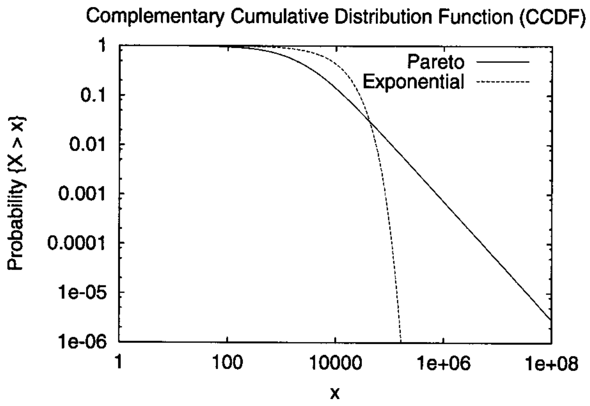


Figure 2.2: *CCDF of a Exponential and a Pareto Distribution*

statistical properties of heavy-tailed distributions. However, the variance of long-tailed distributions is finite. We note that more general definitions for long-tailed distributions are possible.

2.1.4 Pareto Distributions

Pareto and ParetoII distributions (see [Johnson et al.(1994)]) are the simplest class of representants of heavy-tailed distributions and thus frequently used in modeling.

For a Pareto distribution the CDF is given by

$$F(x) = 1 - (\frac{k}{x})^{\alpha} \quad for \quad x \in [k, \infty) \tag{2.13}$$

with parameters *minimal value* k and *shape parameter* α. The shape parameter of the Pareto distribution is equal to its tail index.

For a ParetoII distribution the CDF is given by

$$F(x) = 1 - \frac{1}{(1+\frac{x}{s})^{\alpha}} \quad x \in [0, \infty[\tag{2.14}$$

The ParetoII distribution has two free parameters: the average a, and the shape parameter α which equals its tail index. $s = a * (\alpha - 1)$ is a dependent parameter.

2.2 Modeling Heavy Tails

As we have seen in the previous section, heavy-tailed distributed characteristics have long memory. In many systems this long memory contributes to long-range dependence or self-similarity in performance characteristics, which has significant negative impact on system performance. A similar statement can be made for long-tailed distributed characteristics. Thus, modeling long- or heavy-tails in the distributions of system input is inevitable when generating workload for performance evaluations with synthetic workload.

2.2.1 Heavy Tails in Network Traffic Generation

We elaborate this statement for the generation of synthetic network traffic for performance evaluation.

Generally characterizing network traffic is a difficult problem given that both network technologies and applications that generate traffic vary from site to site and evolve over time. To cope with this high variability, [Paxson and Floyd(1997)] proposes to identify invariants that affect resource allocation and system performance. These invariants have been studied on various activity

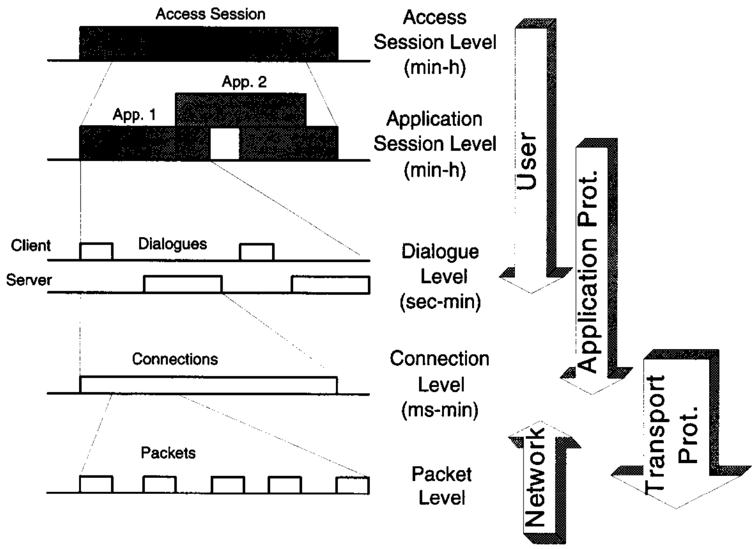


Figure 2.3: *Levels of Activity*

levels (see [Charzinski(2002)] and references therein, and Figure 2.3 (left) for an illustration of activity levels). These activity levels can be linked to user, application, transport protocol, and network behavior (see Figure 2.3 (right)). In addition to that, the relation of invariants on different activity levels has been studied. Probably the most well known invariant of network behavior that affects resource allocation and system performance across studies is the self-similarity of network traffic on the network layer (see Figure 2.4).

However, this *self-similarity* of traffic on the network layer is unlike the distributional self-similarity known from deterministic fractals for which the parts exactly resemble their parts in all detail. This self-similarity is more general in a sense that the shape of the graph depicting the throughput is preserved under aggregation if suitably normalized. The visual impression of

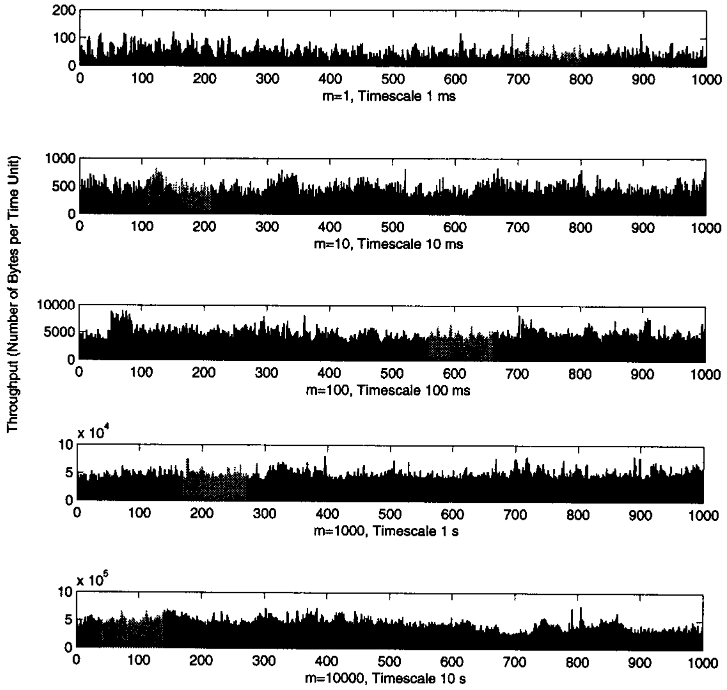


Figure 2.4: *Self-similarity of Web Traffic*

burstiness under aggregation does not change. This phenomenon is called burstiness preservation. Burstiness or variability are formally captured by the statistics of variance, which is related to the second moment of the throughput distribution and therefore called second order statistics. Hence, *second order self-similarity*, which is observed as an invariant for network traffic on the network layer, means that the variance of throughput statistics is preserved under aggregation if suitably normalized. The normalization, which completely defines second order self-similarity, has a direct relation to the correlation structure of statistics. This normalization is completely described with a single parameter, the Hurst parameter. Independent or short-range dependent statistics all lead to a Hurst parameter of 0.5. However, measurements of network traffic suggest an invariant Hurst parameter around 0.9. This invariant Hurst parameter in turn implies a strong long-range-dependent correlation structure among observations of throughput in a network [Leland et al.(1994)].

This in turn has significant negative impact on the performance of services that use the network such as web, e-mail, or FTP. The effect of this long-range dependence in the statistic of network throughput is that relatively long periods of low throughput follow relatively long periods of high throughput, which, in statistics literature, is known as the *Joseph effect*[1]. This leads to drastic reductions in the effectiveness of deploying "buffers" in network components in order to absorb transient increases in traffic load [Erramilli et al.(1996)]. This in turn has a considerable negative effect on system performance.

A series of measurements, simulations, and theoretical considerations have been made to track down the causes for self-similarity in network traffic and explore its effects on network performance.

[Willinger et al.(1995)] theoretically shows that the self-similarity of network traffic can be explained by the size of transfered objects following a heavy-tailed distribution. [Park et al.(1996)] employs simulations to confirm these theoretical results. The work shows that the relation between self-similarity of network traffic and the heavy-tailedness of the ob-

[1] Mandelbrot chose this term in reference to the biblical "seven years of great abundance" and "seven years of famine" account of the irrigation capacities of the river Nile in ancient Egypt. [Mandelbrot and Wallis(1968)]

ject size distribution is not significantly affected by changes in network resources, topology, traffic mixing, the distribution of inter-arrival times. The work also shows that transport layer mechanisms such as TCP's reliability and flow control preserve self-similarity in network traffic. Hence, given that there is evidence that size distributions of objects transfered in real systems possess long tails (see [Arlitt and Williamson(1996)] [Crovella and Bestavros(1996)] [Paxson and Floyd(1994)]), this explanation gives rise to a new understanding of network dynamics.

[Park et al.(1997)] performs a simulation to study the adverse impact of self-similarity in network traffic on performance. This work considers various implementations of TCP's congestion control for reliable, flow-controlled packet transport. In addition to confirming that the explanation that self-similar network traffic with the size distribution of objects transfered is long- or heavy-tailed, [Park et al.(1997)] finds that network performance measured by packet loss and retransmission rate declines smoothly as self-similarity is increased under reliable, flow-controlled packet transport. Queueing delay is increased more drastically. When traffic is highly self-similar, as measured in real networks, queueing delay grows nearly proportionally to the buffer capacity present in the system. From these observations this work infers that provisioning for QoS is a difficult problem at the presence of self-similar traffic.

We thus infer that modeling the the long or heavy tail in the object size distribution is inevitable when synthetically generating network traffic.

2.3 Implications of Heavy-tailed Input

Generating input by sampling a heavy-tailed distribution for synthetical workload generation has severe implications for the stability in performance evaluations. A heavy-tailed distribution has infinite variance or, more formally, non-existing second and higher moments. As a consequence, the convergence of performance statistics that depend on these moments significantly differs from the convergence of the same statistics in systems that sample from commonly used light-tailed distributions with finite variance. If the tail index α of the heavy-tailed distribution is close to one, the performance statistics that depend on these moments cannot even be evaluated at sample sizes that are of practical use.

We illustrate this with the example of a simple statistic that depends on the second moment of the distribution: the running average in a sample from a distribution in system input. Let $X_1, ..., X_n$ be a sample of n independent observations of a random variable in system input. The running sample average can then be defined as

$$\overline{X}_n = \frac{1}{n} \sum_{i=1}^{n} X_i \tag{2.15}$$

If the $X_1, ..., X_n$ are from a light tailed distribution with finite first and second moments, the well known *central limit theorem* (CLT) can be applied. The CLT states that the distribution of $\overline{X}_n$ converges in distribution to a normal distribution $\mathcal{N}$ at rate $n^{-\frac{1}{2}}$.

$$n^{1/2}(\overline{X}_n - \mu) \xrightarrow[in\ distribution]{} \mathcal{N}(0, \sigma^2) \tag{2.16}$$

This convergence implies the following:

1. μ is the most probable value for $\overline{X}_n$ at all sample sizes n. Hence, confidence intervals for $\overline{X}_n$ around μ are symmetric.

2. The distribution of $\overline{X}_n$ has fast decaying exponential tails. Hence, confidence intervals are well centered around μ.

3. The distribution of $\overline{X}_n$ converges at a $n^{1/2}$ rate. Hence, confidence intervals for $\overline{X}_n$, which are given by $\mu \pm 1.96 * \sigma * n^{-1/2}$, lead to convergence of $\overline{X}_n$ at small sample sizes.

This is illustrated in Figure 2.5 (top) which is from [Crovella and Lipsky(2000)]. The figure shows histograms of the distribution of $A_n = \overline{X}_n$ around $\mu = 1$ for increasing sample sizes n. The X_i from which the $\overline{X}_n$ have been computed were drawn from a exponential distribution.

If the $X_1, ..., X_n$ are from a heavy-tailed distribution for which the second moment does not exist (see section 2.1.3), we need to refer to a *generalized central limit theorem* (GCLT) [Nolan(2002)] to describe the convergence of $\overline{X}_n$. This GCLT states that the distribution of $\overline{X}_n$ converges in distribution to a α-stable distribution $\mathcal{S}_\alpha$ at rate $n^{1/\alpha - 1} < n^{-\frac{1}{2}}$.

$$n^{1-1/\alpha}(\overline{X}_n - \mu) \xrightarrow[in\ distribution]{} \mathcal{S}_\alpha \qquad (2.17)$$

α-stable distributions are a superset of normal distributions which have four parameters. α, the index of stability, equals to the shape parameter of the heavy-tailed distribution used for sampling. The special case where $\alpha = 2$ is the set of normal distributions. α-stable distributions have a skewness parameter in addition to the location parameter (a generalization of the parameter μ in the normal distribution), and the scale parameter (a generalization of the parameter σ in the normal distribution). α-stable distributions are known to have no closed form representation except for three special cases. One of these cases is the normal distribution. Usually α-stable distributions are described with their characteristic functions (see [Nolan(2002)]). The convergence of the distribution of $\overline{X}_n$ to an α-stable distribution implies the following:

1. μ can be far from the most probable value for $\overline{X}_n$ since α-stable distributions are usually skewed. Hence, particularly at small sample sizes, confidence intervals for $\overline{X}_n$ around μ can be highly asymmetric.

2. The distribution of $\overline{X}_n$ has at least one slowly decaying

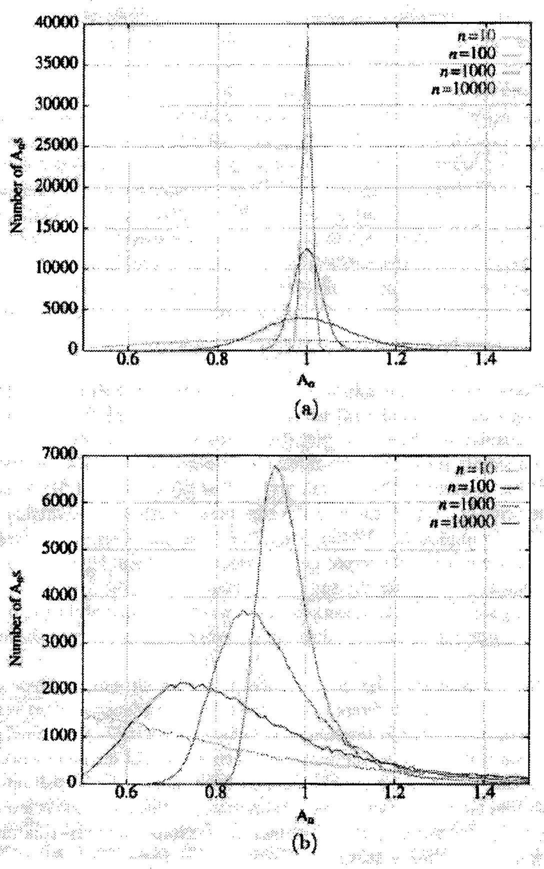


Figure 2.5: *Convergence to Normality vs. Convergence to α-stable (from [Crovella and Lipsky(2000)])*

heavy tail. This tail has the same index α as the heavy-tailed distribution used for generating the X_i. Hence, confidence intervals cannot be well centered around μ.

3. The distribution of $\overline{X}_n$ converges at a $n^{1/\alpha-1}$ rate. Hence, for $\alpha \rightarrow 2$ the convergence is almost as fast as the convergence to a normal distribution. However, for $\alpha \rightarrow 1$, this convergence is severely slowed.

This is illustrated in Figure 2.5 (bottom) which is from [Crovella and Lipsky(2000)]. The figure shows histograms of the distribution of $A_n = \overline{X_n}$ around $\mu = 1$ for increasing sample sizes n. The X_i from which the $\overline{X_n}$ have been computed were drawn from a strictly positive heavy-tailed distribution with tail index $\alpha = 1.4$.

[Crovella and Lipsky(2000)] give the following rough approximation to estimate the sample size required to estimate μ with the running sample average $\overline{X}_n$. The convergence relation 2.17 implies

$$|\overline{X}_n - \mu| \approx c_1 n^{1/\alpha-1} \tag{2.18}$$

They then define the k digit accuracy with which they want to estimate the running sample average as

$$\frac{|\overline{X}_n - \mu|}{\mu} \leq 10^{-k} \tag{2.19}$$

Hence, the sample size n to estimate μ from $\overline{X}_n$ can be obtained by inserting Equation 2.18 into Equation 2.19 which yields in

$$\frac{c_1}{\mu} * n^{1/\alpha-1} \leq 10^{-k} \tag{2.20}$$

Before solving for n, Crovella and Lipsky suggest that approximating $\frac{c_1}{\mu}$ with 1 is sufficient to obtain the order of magnitude of n. Thus, the sample size required for estimating μ from the running average $\overline{X}_n$ with k digit accuracy is given by

$$(\frac{\mu}{c_1} * 10^{-k})^{-\frac{1}{1-1/\alpha}} \leq n \qquad (2.21)$$

Evaluating this equation, [Crovella and Lipsky(2000)] list the sample sizes given in Table 2.1 for two digit accuracy. We note that the definition of k digit accuracy which Crovella and Lipsky use here is a special case of the definition of relative accuracy that is used throughout this thesis (see Equation 2.10). The two digit accuracy corresponds to 1% relative accuracy in our terms.

α	n
2.0	$1.0 \cdot 10^4$
1.7	$7.2 \cdot 10^4$
1.5	$1.0 \cdot 10^6$
1.2	$1.0 \cdot 10^{12}$
1.1	$1.0 \cdot 10^{22}$

Table 2.1: *Required Sample Size to Estimate the Average from the Running Mean [Crovella and Lipsky(2000)]*

2.4 Performance Evaluations with Heavy-Tailed Input

2.4.1 Network and Server Performance

Despite the stability or convergence problem discussed in the previous section, most research works on network and server performance continue to assess performance with the average. They circumvent the convergence problem in two common ways. The first approach is to limit the variability of traffic by tightly truncating the tail of the heavy-tailed distribution. The second and less common approach is to restrict to reporting trends.

Recent works pursuing the first approach are [Christiansen et al.(2000)], [Nahum et al.(2001)], and [Crovella et al.(1999)]. [Christiansen et al.(2000)] employs synthetic workload generation in an experiment to study the effect of RED vs. FIFO queues in routers on network performance. They find that for HTTP/1.0, RED in router queues only leads to minimal improvements on the download latencies of web objects. This work employed [Mah(1997)]'s model for workload generation which includes an empirical distribution of object sizes of web downloads from measurements at UC Berkeley in 1997. The empirical distribution is long-tailed. However, reviewing the distribution, which is available as a table in the code of [ns-2(2000)], we have seen that the largest object is 1.6MB. In addition to this rather small implicit object size limit, [Christiansen et al.(2000)] focuses on optimizing RED for small download latencies below two seconds. Presumably, this leads to a circumvention of the convergence problem.

[Nahum et al.(2001)] employs synthetic workload generation in an experiment to study the effects of wide-area conditions on web server performance. They find that packet loss can reduce the maximum server throughput by as much as 50 percent and increase the server response time required to deliver web objects. The workload in this experiment is generated with the SURGE workload generator [Barford and Crovella(1998)]. This generator produces workload by sampling analytic distributions. The generator employs a long-tailed distribution function to determine the size of web objects. To address the convergence problem, [Nahum et al.(2001)] test the convergence of a "typical" data point. They repeat their experiment 35 times, and use a normal plot to show that the average number of HTTP operations per second converges to normal distribution. To prevent a convergence problem they set the largest object size in the experiment to 3.2MB[Nahum(2002)].

[Crovella et al.(1999)] employs synthetic workload generation with SURGE [Barford and Crovella(1998)] in an experiment to study the scheduling of concurrent downloads of static web objects in web servers. They compare size independent scheduling to smallest-object-first scheduling for the typical web workload in which the object size distribution is long-tailed. The main finding is that a smallest-object-first scheduling policy can allow web servers to significantly lower average download latencies without severely penalizing downloads of large objects. The SURGE workload generator in workload generation has been configured to generate objects with 2000 distinct sizes between 186 bytes and 121MB. This limit circumvents a severe convergence problem. To ensure convergence of results, experiments were run long enough that they were "not strongly influenced by transients". Moreover, web downloads were grouped into 40 bins according to object size, for which average values where taken when studying the dependence of download latencies on the object size.

Recent work pursuing the second approach to restrict to reporting trends is [Fiedler(2001)]. This work performs simulations to investigate how to provision a differentiated services (DiffServ) Intranet which serves three classes of traffic, i.e., voice, real-time and best-effort data. All data traffic is modeled as web traffic. Real-time data traffic is web traffic with a download time requirement. The main finding is that deploying DiffServ is advantageous if the amount of best-effort traffic in the network is high. The workload in this experiment is generated with a SURGE like analytic model. The sizes of web objects are determined by sampling a long-tailed distribution. The largest object in the simulation is around 2GB. The problem of convergence of results is not explicitly addressed in this work. Results are reported as trends.

2.4.2 Task Scheduling in Operating Systems

A number of simulation studies with long- and heavy-tailed job size distributions were conducted to explore various aspects of task scheduling in operating systems.

[Harchol-Balter and Downey(1997)] studies migration policies in a network of workstations. They employ trace-driven simulation to show that preemptive migration outperforms non-preemptive remote execution even when memory-transfer costs for preemptive migration are high. They show that this finding is a consequence of the long- or heavy-tail property in the job size distribution in the trace they used to drive the simulation. The use of a trace prevents any convergence problem.

[Harchol-Balter et al.(1999)] studies task assignment policies for a distributed job server if the job size distribution is long-tailed. They employ simulation and analysis to compare size based policies such as SITA-E to dynamic policies that send the job to the host with the least current load. In SITA-E (size based interval task assignment with equal load), a job size range is associated with each host in the distributed server. Host 1 serves all jobs that have a size between x_0 and x_1, host 2 serves all jobs that have a size between x_1 and x_2, and so on. The cutoff points for the size intervals are chosen such that the load is equally distributed. Hence, in SITA-E, the variability of job sizes arriving at a host is limited, which is shown to be the cause that SITA-E outperforms dynamic policies under highly variable long-tailed job size distributions for statistics such as mean waiting time and mean slowdown. The workload in this study is generated from an analytic model. Job sizes in this model are long-tailed with an upper limit of 10^{10} time units. Results are obtained after arrival of $4 \cdot 10^5$ jobs and averaging over 400 simulation runs. A deviation of simulation results from analytic results, which are "similar in trend", is explained with a hint to the convergence problem of simulations with long- or heavy-tailed input. However, in this study no further analysis is devoted to the convergence problem.

2.5 Performance Evaluation with Quantiles

In the next chapter, we will explain why quantiles have much better convergence properties that the average when the underlying distribution is long-tailed. Despite this fact, the notion of quantiles (and percentiles) is rarely used in the evaluation of network and server performance. To our knowledge only [Raunak et al.(2000)] use quantiles to investigate the potential of proxy caching to improve performance of web downloads. However, this work uses trace driven simulation presumably to avoid a convergence problem. The main finding of the work is that web caching may reduce average network and server latency by up to 60%. However, the 99-th latency percentiles are only reduced by 15-20%. The work thus infers that the benefit of proxy caching to enhance capacity of web services is limited. This finding is explained with the poor locality of large web objects. As a consequence authors have recommended rethinking the cost-benefit tradeoffs of deploying web proxies.

2.6 Summary

In this chapter we have reviewed the background and related work of this thesis. We have introduced the notion of long- and heavy-tailed distributions and have explained the long memory property of long- or heavy-tailed distributed statistics. We have explained why it is important to model this property when performance is evaluated with synthetical workloads. We have illustrated this with the example of performance evaluations with self-similar network traffic. We have explained why performance evaluations of systems with synthetical workloads which are generated by sampling long- or heavy-tailed distributions inherently suffer from a stability problem. Finally, we have reviewed how this stability or convergence problem is addressed in research work.

Chapter 3

Theory

In this chapter we introduce the theory behind the quantile-based method which we develop in Chapter 4. We introduce the notion of quantiles and explain why quantiles are promising statistics when evaluating the performance of systems which generate input by sampling long- or heavy-tailed distributions. We generally explain the impact of correlation in system output on convergence and show that limit theorems for M-estimators can be employed for quantiles. We show that under reasonable assumptions we can generally expect that quantiles in system output converge at sample sizes that are feasible in the practice.

3.1 Definition of Quantiles

Definition 1 (p-th Quantile of a Distribution) *We define the p-th quantile x_p of the distribution F as the smallest value x_p for which $P(X < x_p) \leq p$.*

It follows from this definition that $p \in [0, 1]$.

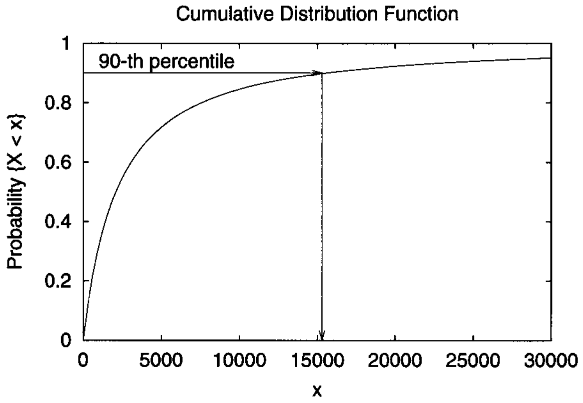


Figure 3.1: *Relation CDF and Quantile*

If p is represented by a percentage value we call the p-th quantile a $p * 100$-*th percentile*.

An example for this is the 99-th object size percentile from a known object size distribution. By definition 99% of the object have a smaller size than this object size percentile.

Theorem 1 *Suppose that F is the CDF of a continous random variable and is strictly increasing on some interval I, and that $F = 0$ to the left of I, and that $F = 1$ to the right of I. I may be unbounded. Then the inverse F^{-1} exists and the p-th quantile x_p of F is given by*

$$x_p = F^{-1}(p) \tag{3.1}$$

Proof is by the definition of CDF (see [Rice(1995)]). Figure 3.1 illustrates this theorem.

To enable quantile-based evaluation of system performance, we need to determine the convergence of quantile in a sample

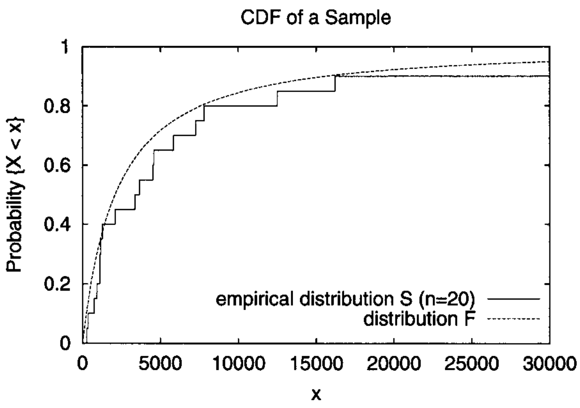


Figure 3.2: *Sample's CDF*

$X_1, .., X_n$ of observations of a random variable X. We thus introduce the notion of order statistics and empirical distribution in order to define the quantile of a sample. The *order statistic* of the sample can be obtained by arranging the observations $X_1, .., X_n$ in increasing order allowing repetitions. We denote the order statistic with

$$X_{(1)} \leq X_{(2)} \leq ... \leq X_{(n)} \tag{3.2}$$

The order statistic can now be employed to define the *empirical distribution* S_n of the random variable X in the sample $X_1, .., X_n$

$$S_n(x) = \begin{cases} 0 & \text{if } x < X_{(1)} \\ \frac{i}{n} & \text{if } X_{(i)} \leq x < X_{(i+1)} \text{ and } i < n \\ 1 & \text{if } x \geq X_{(n)} \end{cases} \tag{3.3}$$

If the sample has been obtained by independently sampling a distribution $F(x)$, the weak law of large numbers implies that for any fixed x

$$S_n(x) \xrightarrow[n \to \infty]{} F(x) \tag{3.4}$$

(See Figure 3.2 for an illustration of this convergence). Therefore F is sometimes called the *limit* or *asymptotic distribution.*

Hence, the definition of the p-th quantile in a sample can be stated as:

Definition 2 (p-th Quantile of a Sample) *The p-th quantile of a sample is the p-th quantile of the corresponding empirical distribution S_n.*

This definition implies that $X_{(k)}$ is the p-th sample's quantile with $k = \lceil np \rceil$.

3.2 Distribution of a Sample's Quantile

In this section, we give a first insight why quantiles are promising statistics for the performance evaluation of systems with heavy-tailed input. We start with assuming independent sampling and review the distribution of a sample's p-th quantile and the corresponding limit theorem. This limit theorem says that a sample's p-th quantile converges to a normal distribution at a $n^{-1/2}$ rate. This also holds for heavy-tailed distributions that are sufficiently regular, i.e. smooth, in the neighborhood of the quantile. For heavy-tailed distributions, this convergence then is fundamentally different from the convergence of the sample's average to α-stable at a rate slower than $n^{-1/2}$, which leads to impractical sample size for performance evaluation.

Under the assumption of independent sampling, the distribution of the p-th sample quantile is given by the following theorem:

Theorem 2 (Distribution of the p-th Sample Quantile) *Let $X_1, .., X_n$ be n independent observations on a random variable*

X that follow the distribution F. Let F be such that two regularity conditions hold: (i) $F(x)$ admit a continuous PDF $f(x)$ for all x and (ii) let the p-th quantile x_p of the distribution F be unique and $f(x_p) > 0$. Then, the probability density $f_k(x)$ of the p-th sample's quantile $X_{(k)}$ with $k = \lceil np \rceil$ is given by

$$f_k(x) = n\binom{n-1}{k-1}(F(x))^{k-1}(1 - F(x))^{n-k}f(x) \qquad (3.5)$$

Proof:

The event $x \leq X_{(k)} \leq x + dx$ occurs if $k - 1$ observations are smaller than x, one observation is the interval $[x, x + dx]$, and $n - k$ observations are larger than $x + dx$. Under assumption of independence, the probability of any particular arrangement of this type is $F^{k-1}(x)f(x)[1 - F(x)]^{n-k}dx$. By the multinomial theorem, there are $n\binom{n-1}{k-1}$ such arrangements. Hence, $f_k(x)$ is given by equation 3.5. For details see [Rice(1995)], section 3.7, p. 101.

The corresponding limit theorem for the distribution of the p-th sample quantile for sample size $n \rightarrow \infty$ can be given as follows.

Theorem 3 (Limit Theorem for the p-th Sample Quantile)
Let $X_1, .., X_n$ be n independent observations on a random variable X that follow the distribution F. Let F be such that two regularity conditions hold. Let (i) $F(x)$ admit a continuous PDF $f(x)$ for all x and (ii) let the p-th quantile x_p of the distribution F be unique and $f(x_p) > 0$. Let $k = \lceil np \rceil$ when $n \rightarrow \infty$. Then

$$\sqrt{n}(X_{(k)} - x_p) \rightarrow \mathcal{N}(0, \sigma^2) \quad for\, n \rightarrow \infty\, with\, \sigma = \frac{\sqrt{p(1-p)}}{f(x_p)} \qquad (3.6)$$

We note that this theorem is the quantile's equivalent of the central limit theorem for the running sample average. Convergence properties in terms of rate of convergence and the limit distribution in this theorem are identical to the ones in the central limit theorem. For a proof of the theorem, which is essentially straightforward from Theorem 2, see [Rao(1973)], section 6f.2, p.423.

We now get a first insight why quantiles are promising statistics to evaluate the performance of systems that generate input by sampling long- or heavy-tailed distributions. Theorem 3 says that quantiles in this input converge to a normal distribution at a $n^{-1/2}$ rate if generated by independent sampling. This is fundamentally different to the average of the sample or other statistics that depend on moments of the heavy-tailed distribution. These statistics converge to α-stable distributions at a rate $n^{1/\alpha-1}$ (see Section 2.3). As a consequence, quantiles of system input from long- or heavy-tailed distributions such as the 99-th percentile can be evaluated at sample sizes that are feasible in practice. We only have to make sure that the long- or heavy-tailed distribution in system input is such that the regularity conditions (i) and (ii) in Theorem 3 are fulfilled and the probability density at the quantile is significantly different from zero.

In the next section we show that similar statements can be made for the convergence of quantiles in system output where observations are not independent.

3.3 The Impact of Correlations on Convergence

The assumption that observations in the sample are independent does not hold for system output. Limit theorems such as Theorem 3 need modifications to cover situations in which observations in a sample are correlated. Before reviewing theory that covers the convergence of quantiles of correlated observations, we generally discuss the impact of correlation to convergence. We define and explain the different impact of weak and long-range dependence. We use the example of the variance of the running sample average for this discussion and assume two finite moments for the limit distribution of observations.

We recall that under assumption of independence, the variance of the sample average is given by:

Theorem 4 *Let $X_1, .., X_n$ be a i.i.d. (independent and identically distributed) sample from a distribution F which has average μ and variance σ^2. Then the variance of the sample's average is given by:*

$$var(\overline{X}_n) = \sigma^2 n^{-1} \tag{3.7}$$

Proof:

Since the X_i are i.i.d.

$$var(\overline{X}_n) = \frac{1}{n^2} \sum_{i=1}^{n} Var(X_i) = \sigma^2 n^{-1} \tag{3.8}$$

since $Var(X_i) = \sigma^2$.

We now define covariance and autocorrelation to enable a discussion of required modifications of Theorem 4 that account for correlated observations.

Definition 3 (Covariance and Autocorrelation) *Let $X_1, .., X_n$ be a sample from a distribution F which has average μ and variance σ^2. The covariance of two observations X_i and X_j in the sample is then given by:*

$$Cov(X_i, X_j) = E[(X_i - \mu)(X_j - \mu)] \tag{3.9}$$

The autocorrelation between two observations X_i and X_j in the sample is then given by:

$$\rho(i,j) = \frac{1}{\sigma^2} Cov(X_i, X_j) \tag{3.10}$$

We denote that a more sophisticated definition of autocorrelation is required if the limit distribution $F(X)$ has a non existing second moment (see [Beran(1994)] chapter 11).

Definition 3 leads to the following identity for $var(\overline{X}_n)$:

$$var(\overline{X}_n) = n^{-2}\sigma^2 \sum_{i,j=1}^{n} \rho(i,j) \tag{3.11}$$

since

$$var(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} Cov(X_i, X_j) \tag{3.12}$$

The special case where all correlations for $i \neq j$ sum up to zero leads to equation 3.7 since

$$\sum_{i=1}^{n} \rho(i,i) = n \tag{3.13}$$

For other cases the variance is given by:

Theorem 5 *Let $X_1, .., X_n$ be a sample from a distribution F which has average μ and variance σ^2. Let $\rho(i,j)$ be the autocorrelations of $X_1, .., X_n$. Then the variance of the sample's average is given by:*

$$var(\overline{X}_n) = \sigma^2 c(\rho) n^{-1} \tag{3.14}$$

with

$$c(\rho) = [1 + \delta_n(\rho)] \tag{3.15}$$

and

$$\delta_n(\rho) = n^{-1} \sum_{i \neq j} \rho(i,j) \tag{3.16}$$

Proof is with equation 3.11 and equation 3.13.

This leads to the following definition of weak dependence:

Definition 4 (Weakly dependent Correlation Structure) *We define the correlation structure in the sample as weakly dependent if correlations are summable, i.e.*

$$\sum_{i \neq j} \rho(i,j) < \infty$$

.

If now the generation process is stationary, i.e. $E(X_i) = \mu$ for all i and correlations depend on the lag $|i - j|$ only, the relation between the decay of correlations and the dependence structure can be stated as follows:

Theorem 6 *For a stationary generation process the correlation structure is weakly dependent (WD) if and only if correlations decay exponentially with the lag $k = |i - j|$ or faster, i.e.*

$$|\rho(k)| \leq ba^k \; where \; 0 < b < \infty, 0 < a < 1 \tag{3.17}$$

Combining Theorem 5 and Theorem 6 leads to the following implication if the generation process is stationary: Exponential or faster decay of autocorrelations with lag $k \rightarrow \infty$ lead to a weakly dependent correlation structure. The implication on convergence of this correlation structure is limited to an enlargement of the variance in the asymptotic distribution by a constant factor compared to the i.i.d. case. This enlargement depends on the sum of all correlations and not a single or a few correlations. The rate of convergence remains unchanged.

However, from observations in a number of scenarios it is known that the rate of convergence can be slowed down (see [Beran(1994)]). We therefore model the slowed-down rate of convergence with the simplest possible approach and establish the relation to the convergence structure. We exchange n^{-1} in Equation 3.14 by $n^{-\alpha}$. Then

$$var(\overline{X}_n) \approx \sigma^2 c(\rho) n^{-\alpha} \ where\, 0 < \alpha < 1 \tag{3.18}$$

with

$$c(\rho) = \lim_{n \to \infty} n^{\alpha - 2} \sum_{i \neq j} \rho(i, j) \tag{3.19}$$

It can be shown that, under assumption of a stationary generation process, the sum of the correlations diverges for $n \to \infty$, i.e.

$$\sum_k \rho(k) = \infty \tag{3.20}$$

With the following definition of long-range dependence from [Beran(1994)] it immediately follows that this slowed-down rate is a consequence of long range dependence.

Definition 5 (Long Range Dependence) *We define the correlation structure in the sample as long range dependent if there exists some* $0 < \alpha < 1$ *for which*

$$\sum_{i \neq j} \rho(i, j) = \infty$$

and

$$\lim_{n \to \infty} n^{\alpha-2} \sum_{i \neq j} \rho(i,j) < \infty$$

.

We can now establish the relation between a long range dependence structure and the decay of correlations:

Theorem 7 *For a stationary generation process the correlation structure is long-range dependent (LRD) if and only if correlations decay polynomially with the lag $k = |i - j|$, i.e.*

$$\rho(k) \approx c_\rho |k|^{-\alpha},\ with\ c_\rho > 0\ for\ |k| \to \infty \tag{3.21}$$

Combining Equation 3.18 and Theorem 7 leads to the following implication if the generation process is stationary: Polynomial decay of autocorrelations with lag $k \to \infty$ leads to a long-range dependent correlation structure. The implication on convergence of this correlation structure is, in addition to an enlargement of the variance, a slow-down in the rate of convergence compared to the i.i.d. case. The degree of slow-down depends on the exponent in the polynomial decay.

3.3.1 Convergence of Quantiles

We now review generalizations of the limit theorem for the p-th sample quantile (Theorem 3) to justify the expectation that sample's quantiles in system output usually converge to a normal distribution. These generalizations of the limit theorem account for correlations among observations. These generalizations are limit theorems for M-estimators since quantiles are a special case of a generalized maximum likelihood estimator (M-estimator). M-estimators are popular in location estimation since they enable to prove limit theorems for entire classes of statistics rather than for a single statistics such as average, median, or other quantiles.

We introduce M-estimators as follows: We recall that a *maximum likelihood estimator* (mle) of a statistic θ is the value of θ that makes the observed data most probable, i.e. maximizes the likelihood of this data. Assuming that the X_i are from different independent estimation attempts, the joint probability density for the observed data is given by the product of the marginal densities.

$$lik(\theta) = \prod_{i=1}^{n} f(X_i|\theta) \tag{3.22}$$

where f is the probability density of the marginal distribution. This is often called likelihood. Estimation of θ then implies the maximization of this product. However, rather than maximizing the product itself, maximum likelihood estimation suggests to maximize its natural logarithm.

$$l(\theta) = \sum_{i=1}^{n} log[f(X_i|\theta)] \tag{3.23}$$

This is equivalent and usually easier to compute.

Maximum likelihood estimators are frequently used since it can be shown that mles have a number of desirable properties. The most important one for this thesis is that, under minimal conditions on regularity for the underlying distribution F, maximum likelihood estimators converge to a normal distribution.

The generalization of maximum likelihood estimators to M-estimators can now be explained with the following example: The location of the normal distribution is μ. Thus, estimating μ with the sample average $\overline{X}$ is equivalent to minimizing the corresponding negative log likelihood, or

$$\sum_{i=1}^{n} \left(\frac{X_i - \mu}{\sigma}\right)^2 \tag{3.24}$$

Outliers, which have non negligible probability if the underlying distribution $F(X)$ is long-tailed, have significant effects on this estimate. When $F(X)$ is heavy-tailed, this estimate is not even defined since the variance σ^2 is infinite. In contrast, the median is known to minimize

$$\sum_{i=1}^{n} \left|\frac{X_i - \mu}{\sigma}\right| \tag{3.25}$$

Hence, outliers have much less weight and the median leads to a much more robust estimate for μ. Abstracting from Equation 3.24 and Equation 3.25, [Huber(1981)] started to study the properties of classes of statistics. He introduced *M-estimators*, which are the minimizers of

$$\sum_{i=1}^{n} \Psi(\frac{X_i - \mu}{\sigma}) \tag{3.26}$$

where the weight function Ψ is a compromise between the weight functions for the sample average and the sample median. Statistics literature frequently works with $\psi = \Psi'$ rather than with Ψ and assume that σ is known and equal to 1.

Other quantiles than the median can also be expressed with the M-estimator. The p-th quantile is obtained by setting

$$\psi(x) = \begin{cases} p-1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ p & \text{if } x > 0 \end{cases} \tag{3.27}$$

where c is a constant that depends on the parameter σ in the normal distribution. We denote that $\psi(x)$ has an irregularity at $x = x_p$.

This implies that we have to carefully review the conditions in limit theorems for M-estimators before we make any inferences for the convergence of quantiles. For the case of a weakly dependent correlation structure among observations [Hampel et al.(1986)] reports in section 8.3 that M-estimators generally converge to normal. The implications of these weakly dependent correlations are the same as in the example discussed in section 3.3. The variance is enlarged, the rate of convergence remains at $n^{-1/2}$. The relation between correlations and enlargement can be formalized with so called influence functions. The regularity condition required for this convergence is that F can be approximated with its Taylor expansion around the quantile. This is comparable to the regularity condition required for convergence in the i.i.d. case (see Theorem 2). For the problem of performance evaluation of web traffic, it seems reasonable to assume such regularity in system output. This in turn implies that we can expect that quantiles in system output converge to normal at a $n^{-1/2}$ rate as long as the correlation structure in system output is consistent with weak dependence.

For the convergence of quantiles in system output that shows a long-range dependent correlation structure, we have to refer to limit theorems for so called reference processes with long memory. These reference processes are Gaussian processes such as fractional Gaussian noise and fractional ARIMA (auto regressive integrated moving average). We refer to Gaussian processes and processes that are derived from Gaussian processes. [Beran(1994)] reports in section 8.3 that M-estimators of every process that can be derived from a Gaussian processes converge to normal. The impact of the long-range dependent correlations are the same as in the example discussed in section 3.3: The variance of the normal distribution is enlarged and the rate of convergence is slowed down. The enlargement can be evaluated with Hermite polynomials. The relation between the convergence of the average and the convergence of any other statistic that can be expressed with M-estimators, can also be established with Hermite polynomials. The rate of convergence for all statistics that can be expressed with M-estimators is slowed down to n^{H-1} where H is the Hurst parameter.

Strictly speaking, the regularity conditions for a quantile's ψ as expressed in Equation 3.27 are not sufficient that the results of section 8.3 in [Beran(1994)] apply for quantiles. However, in an e-mail exchange that is documented in Appendix A, the author argues that the quantile's ψ can be approximated arbitrary closely with regular ψ_n. As a consequence, results in section 8.3 of [Beran(1994)] also hold for quantiles.

For our problem of performance evaluation this implies that quantiles in system output converge to normal at a n^{H-1} rate whenever the system is comparable to a reference process with long memory. Whenever system output is long-tailed, the convergence relation between average and other statistics lets us expect that quantiles such as the 99-th percentile converge faster than the average.

3.4 Summary

In this chapter we have introduced the theory behind the quantile-based method which we develop in Chapter 4. We have introduced the notion of quantiles and have explained why quantiles are promising statistics when evaluating the performance of systems which generate input by sampling long- or heavy-tailed distributions. We have generally explained the impact of correlation in system output on convergence and have shown that limit theorems for M-estimators can be employed for quantiles. We have shown that under reasonable assumptions we can generally expect that quantiles in system output converge at sample sizes that are feasible in practice.

Chapter 4

A Method for Quantile-based Evaluation

In this chapter, we present the quantile-based method which aims at evaluating the performance of systems with heavy-tailed input at sample sizes that are feasible in practice. The system model which we assume is depicted in Figure 4.1.

An example for such a system is the simulation of the network of a web service. The input generation process in such a simulation involves a heavy-tailed object size distribution with tail index α between 1.0 and 1.2. The system output of interest is the latency of the web downloads. Output statistics that depend on the moments of this object size distribution, such as the average latency, cannot be evaluated at sample sizes that are feasible in practice (see Section 2.3 for details).

In order to statistically evaluate the performance of such systems, we propose to evaluate quantiles in system output such as the 99-th, 98-th, and 95-th percentile. This is due to three reasons.

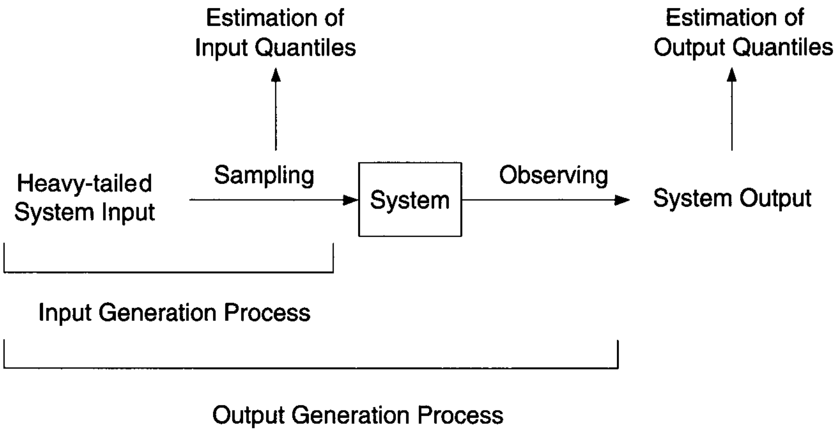


Figure 4.1: *System Model*

1. First, these quantiles have a natural interpretation in statistical quality of service. For the example of the 99-th latency percentile of a web service, this interpretation can be stated as follows: 99% of the downloads have a latency smaller than the 99-th latency percentile. Hence, the statistical performance of the web service can be characterized with download latency quantiles in the high nineties such as the 95-th, 98-th, or 99-th percentile. A similar statement can be made for network and server latencies of web downloads. For details of this concept see [Fiedler(2001)].

2. Quantiles of system output do not necessarily depend on the extreme tail of the input distribution and hence not on moments of this distribution. Therefore, quantiles may converge at sample sizes for which statistics that depend on moments of the input distribution do not converge. This in turn enables statistical performance evaluation. Moreover, we know that, when mild regularity conditions apply for the limit distribution of system output, quantiles in system out-

put converge to a normal distribution (see Section 3.3 for details). With mild regularity conditions for the limit distribution we mean for example that this distribution has a continuous first derivative which is different from zero on the interval that includes all output values. This convergence to a normal distribution can be assessed with standard procedures such as normal plots and convergence tests. In case of convergence, this additionally provides an accurate estimate of the p-th quantile in system output. This estimate can then be used to state statistical QoS guarantees for the system.

3. Lower bounds to the initial phase in the convergence of output quantiles can be inferred from the convergence of input quantiles. We exploit the fact that under mild regularity conditions on the heavy-tailed distribution in system input, input quantiles converge to a normal distribution. In this case, the convergence can be established with probability theory (see Section 3.2). This leads to lower bounds of sample sizes which are feasible for performance evaluation. The assumption behind this is that for any statistic in system output to converge, the corresponding statistic in system input has to converge. We think that this assumption, which is comparable to the assumption in [Crovella and Lipsky(2000)] stating that system stability requires convergence of the average in system input, is reasonable for the applications intended by this thesis.

The method for performance evaluation with quantiles thus pertains to (i) an estimation of the lower bounds for the initial phase of convergence of quantiles in system output and (ii) an assessment of the convergence of output quantiles including estimation of quantiles. We first develop the two parts of the method for ParetoII distributed input with tail index close to 1 for systems which can be evaluated based on the 99-th percentile. Then we generalize the method to other heavy-tailed distributions and quantiles other than the 99-th percentile. We start with listing the assumptions about system properties upon which the method is based. In Section 4.1 we develop the estimation of lower bounds. In Section 4.2 we develop the assessment of convergence and the estimation of quantiles in system output. Finally, in Section 4.3 we discuss generalization of the method.

The assumptions on system properties which the method makes can be listed as follows:

1. System performance can be evaluated based on the statistics of the 99-th percentile of system output.
2. This statistic must have converged to enable QoS evaluation. Presumably for this statistic to converge the corresponding statistic in system input has to converge.
3. Input to the system is obtained by independently sampling probability distributions.
4. The heavy-tailed distribution in system input is represented with a ParetoII distribution. The tail index of this ParetoII distribution is close to one, e.g. $\alpha = 1.2$.
5. The system is such that the system output has finite variance due to system limits. Mild regularity conditions hold for the limit distribution of system output: The limit distribution has a continuous first derivative which is different from zero on the interval that includes all output values.

4.1 Estimation of Minimal Sample Sizes

We assume that for a statistic in system output to converge the corresponding statistic in system input has to converge (System property 2). Hence, for a quantile such as the 99-th percentile of system output to converge the corresponding quantile of the heavy-tailed distribution in system input has to converge. Sample sizes that are required to converge quantiles in system input to a desired accuracy can be evaluated based on Theorem 2 and Theorem 3. Theorem 2 gives the probability density f_n of the p-th quantile in a sample. Theorem 3 says that this density converges to the density of a normal distribution at rate $n^{-1/2}$ for sample size $n \rightarrow \infty$. These theorems can be exploited for evaluation since the condition of independent sampling and the two conditions on regularity made in both theorems are fulfilled: The independent sampling is by system property 3. The regularity conditions follows from system property 4 since a ParetoII distribution has a continous first derivative $f(x)$ for all $x > 0$ and is strictly monotonous which leads to $f(x) > 0$ for all $x > 0$ which in turn leads to unique quantiles.

4.1.1 Evaluating Convergence of Input Quantiles

We assume that the accuracy to which a quantile in system input has converged at sample size n is given with

$$Accuracy_n = max\{|\frac{L_n - E_n}{E_n}|, |\frac{U_n - E_n}{E_n}|\} \tag{4.1}$$

where E_n are the expected value and $[L_n, U_n]$ are the lower and upper bound of the confidence interval of the quantile (see Section 2.1.2). Hence, the method to evaluate the sample size required to converge the p-th quantile from the heavy-tailed distribution in system input to a given accuracy can be stated as follows: For all sample sizes n, the expected value E_n for the p-th sample quantile is equal to the quantile of the distribution independent of n since any heavy-tailed distribution represented by a ParetoII distribution is continuous (System property 4). Hence,

$$E_n = F^{-1}(p) \tag{4.2}$$

At fixed sample size n, the confidence interval bounds L_n and U_n, and thus the accuracy, can be evaluated by successive numerical integration of the p-th quantile probability density f_n. This density is given in Equation 3.5 of Theorem 2 (employs System property 3+4).

$$f_n(x) = n\binom{n-1}{k-1}(F(x))^{k-1}(1-F(x))^{n-k}f(x) \qquad (4.3)$$

with $k = \lceil np \rceil$. The relations to determine the confidence intervals bounds are

$$\int_0^{L_n} f_n(x)dx \doteq \frac{1 - confidence\, level}{2} \qquad (4.4)$$

for the lower bound L_n and

$$\int_0^{U_n} f_n(x)dx \doteq 1 - \frac{1 - confidence\, level}{2} \qquad (4.5)$$

for the upper bound U_n. Thus the required sample size to converge the p-th quantile to a desired accuracy can be evaluated by iterating the sample size n and evaluating the numerical integrations for the confidence interval bounds. We denote that this numerical integration cannot be directly performed with standard software such as Mathematica [Mathematica(1999)] since the first factor $n\binom{n-1}{k-1}$ in f_n can become very large compared to the remaining factors. This problem can be solved by evaluating f_n indirectly via

$$log(f_n) = log(fac_1) + .. + log(fac_m) \qquad (4.6)$$

instead of

$$f_n = fac_1 \cdot .. \cdot fac_m \qquad (4.7)$$

4.1.2 Large Sample Approximation

A good starting point for the iteration which is close to the final result can be obtained from Theorem 3. The theorem implies that in large samples the distribution of the p-th sample quantile can be approximated by a normal distribution. This normal distribution is centered around the p-th quantile of the heavy-tailed distribution in system input which can be evaluted with

$$x_p = F^{-1}(p) \tag{4.8}$$

The variance of this normal distribution follows from Theorem 3

$$\sigma^2 = \frac{p(1-p)}{nf^2(x_p)} \tag{4.9}$$

where $f(x_p)$ is the probability density of the ParetoII distribution in system input. The confidence interval of this normal distribution is $x_p \pm 1.96\,\sigma$. Hence, the sample size can be derived by evaluating the accuracy relation

$$\frac{1.96\,\sigma}{x_p} \leq accuracy \tag{4.10}$$

Inserting Equation 4.9 in Equation 4.10 and solving for n yields in an estimation for the sample size

$$\left(\frac{1.96}{x_p}\right)^2 \frac{p(1-p)}{f^2(x_p)} \frac{1}{accuracy^2} \leq n \tag{4.11}$$

From this equation it can be seen that the sample size required to converge quantiles of interest in system input will remain feasible for performance evaluations as long as the probability density f at the quantile x_p of the heavy-tailed distribution is not extremely small. This is the case for the 99-th from a ParetoII distribution if the tail index is close to one (System property 4).

4.1.3 Special Cases

In special cases, this estimated sample size also approximates the the sample size required to converge the corresponding quantile in system output. Such a special case is given when the relation between in- and output around the quantile can be well approximated as linear. The reason for this is the preservation of a normal distribution under a linear transformation. In the context of a simulation of web services this may apply when utilization is low enough that effects from TCP's congestion control and retransmission algorithms which are triggered by packet losses become statistically negligible.

4.2 Evaluating Quantiles in System Output

Assessing the convergence of output quantiles to a normal distribution and inferring estimates to evaluate system performance in case of convergence can be achieved as follows:

1. Visually assessing convergence. This can be done with normal plots.
2. Testing the hypothesis that the p-th quantile in system output converges to a normal distribution. This can be achieved with assessing the linearity in the normal plot or other standard procedures, i.e. frequently-used normality tests (see Appendix C).
3. Estimating the p-th quantile in system output and its accuracy under the assumption that this quantile is normally distributed. This can be achieved with standard procedures.

4. Checking the rate of convergence. This can be achieved by checking whether the decay of the variance for increasing sample size n is as expected from probability theory.

4.2.1 Visually Assessing Convergence

Visually assessing the convergence of quantiles of system output can be achieved with normal plots. Normal plots[1] are a extremely useful graphical tool for qualitatively assessing the fit of of data to a normal distribution.

We assume that we have m samples of system output with size n that we obtained by independent performance evaluations of the system. The p-th sample quantiles of these samples, were obtained by ordering the n observations in each sample

Sample #1: $1 \rightarrow Y_{(1),1} \leq .. \leq Y_{(k),1} .. \leq Y_{(n),1}$

...

Sample #m: $m \rightarrow Y_{(1),m} \leq .. \leq Y_{(k),m} .. \leq Y_{(n),m}$

and setting $k = \lceil np \rceil$.

To produce the normal plot, we arrange the output quantiles $Y_{(k),1} \ldots Y_{(k),m}$ in ascending order:

$Y_{(k),1} \ldots Y_{(k),m} \rightarrow Y_{(k),(1)} \ldots Y_{(k),(m)}$.

Then we exploit that if this ordered set is consistent with normality, the expected value of $Y_{(k),(i)}$ is the $\frac{i}{m+1}$ quantile of a normal distribution with unknown parameters μ and σ:

$$E(Y_{(k),(i)})) = \mathcal{N}^{-1}(\mu, \sigma^2)(\frac{i}{m+1}) \qquad (4.12)$$

[1] Normal plots are sometimes also called normal probability plots or normal quantile-quantile plots

Not knowing the parameters μ and σ of the normal distribution $\mathcal{N}(\mu, \sigma^2)$, we can exploit that any quantile of a normal distribution can be related to the corresponding quantile of the standard normal distribution $\mathcal{N}(0,1)$. The relation is (see [Rice(1995)]):

$$\mathcal{N}^{-1}(\mu, \sigma^2)(\frac{i}{m+1}) = \sigma * \mathcal{N}^{-1}(0,1)(\frac{i}{m+1}) + \mu \qquad (4.13)$$

Therefore a *normal plot* plots the $Y_{(k),(i)}$ against the $\frac{i}{m+1}$ quantile of the standard normal distribution.

If the data in the set is close to normal distributed, the result of the plot is close to a straight line. Any deviation in the data from normality such as skewness or subexponential tails can be visually inspected (see Appendix B for a discussion of typical deviations from linearity in normal plots).

4.2.2 Testing for Convergence

However, care needs to be taken in classifying a sample quantile as converged to a normal distribution. To enhance reliability of the classification we need to extend the normal plot to a hypothesis test. The simplest way to perform a test is employing linear regression to evaluate the deviation from linearity in the normal plot. The correlation coefficient r from this linear regression, which is a quantitative measure for the deviation from linearity, can then be compared against the critical values at given significance level (see Table 4.1). For e.g. sample size $i = 30$ [Rice(1995)] reports that, if the data is consistent with normality, 10% of plots have a correlation coefficient below 0.9707, 5% have a coefficient below 0.9639, and 1% have a coefficient below 0.9490. Values for $i = 40$ are 0.9767, 0.9715, and 0.9597. The critical values, originally given in [Filliben(1975)], were obtained from Monte Carlo simulations to determine the null sampling distribution of r under normality.

m	10%	5%	1%
10	0.9347	0.9180	0.8804
15	0.9506	0.9383	0.9110
20	0.9600	0.9503	0.9290
30	0.9707	0.9639	0.9490
40	0.9767	0.9715	0.9597
50	0.9807	0.9764	0.9664
60	0.9836	0.9799	0.9710

Table 4.1: *Critical Values for Normality Test [Rice(1995)]*

Alternatively, we can employ frequently-used normality tests (see Appendix C for an overview).

4.2.3 Inferring Estimates of Quantiles

Intercept and slope of this linear regression determine the estimates of the parameters μ and σ of the normal distribution. This can be see from Equation 4.13. We call these estimates m_n and s_n. Hence, $m_n \pm 1.96 * s_n$ is the confidence interval for the estimate m_n of the quantile in system output at given sample size n. The accuracy of this estimation then is

$$accuracy = \frac{1.96 * s_n}{m_n} \tag{4.14}$$

4.2.4 Rate of Convergence

In order to further enhance the reliability of this test method we propose to successively test the convergence for increasing sample sizes n and to check whether the rate of convergence is as expected. We know that under mild regularity conditions on the distribution of system output (System property 5) this rate has to be $n^{-1/2}$ where n is the sample size if the correlation structure of system output is consistent with weak dependence (see Section 3.3). If the correlation structure is consistent with long-range dependence, we can expect this rate to be n^{H-1}, where H is the Hurst parameter. Thus, the estimated variance s_n of a normally distributed quantile should decay with a $n^{-1/2}$ or n^{H-1} rate. We therefore propose to check whether

- $s_n * \sqrt{n}$ is constant. This is consistent with a weak dependent correlation structure in system output
- $log(s_n)$ is linear in $log(n)$. This is consistent with a long-range dependent correlation structure in system output.
- $log(s_n)$ is not linear in $log(n)$.

4.3 Generalizations of the Method

We note that the proposed method is not limited to performance evaluations with ParetoII distributed input that are evaluated based on the 99-th quantile in system output. The method can usually also be employed for evaluation with other heavy-tailed distributions in system input and other fixed quantile in system output.

However, three requirements need to be fulfilled.

1. The output quantile of interest, which is used to evaluate the system performance, must be chosen such that all p-quantiles of input that impact the quantile of interest in system output must have an order p which is not extremely close to one. This is the case for the 99-th input percentile but is not the case for the 99.999-th input percentile. Without this requirement the p-th input quantile cannot converge

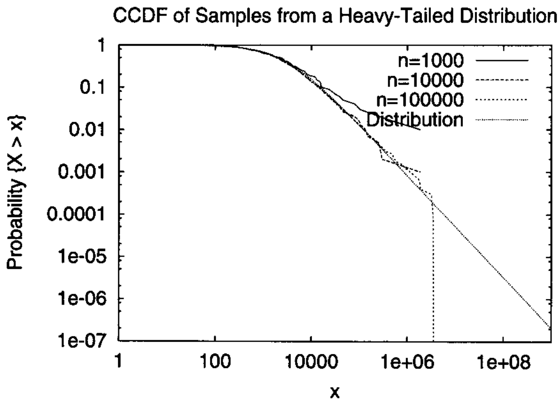


Figure 4.2: *CCDF of Samples from a Heavy-Tailed Distribution*

at sample sizes of interest. This can be inferred from Equation 4.11 and the fact that $f(x_p) \longrightarrow 0$ for $p \longrightarrow 1$. For an illustration see Figure 4.2.

2. Presumably the heavy-tailed distribution in system input must be such that the regularity conditions in Theorem 3 are fulfilled. This implies that the distribution is such that quantiles are unique, and have a nonzero probability density. Moreover, the probabiltiy density is continuous for all values. Without this requirement input quantiles may not converge at sample sizes of interest.

3. The system must be such that minimal regularity conditions such as the uniqueness of the quantile of interest hold for the limit distribution of system output. Without this system property output quantiles cannot converge to a normal distribution.

4.4 Summary

In this chapter, we have presented a method which aims at statistically evaluating the performance of systems with heavy-tailed input at sample sizes that are feasible in practice. The method is based on an evaluation of quantiles such as the 95-th, 98-th, or 99-th percentile in system output. The method pertains to (i) estimation of the minimal sample sizes required for performance evaluation and (ii) a method to evaluate quantile of system output that employs standard normality test procedures. The evaluation method (ii) pertains to a visual pre-test of convergence of quantiles with normal plots, testing convergence with a hypothesis test, estimation of the confidence intervals for the quantiles out of the hypothesis test and checking the rate of convergence.

Chapter 5

Simulation Environment

In this chapter, we describe the simulation environment which we employ in the validation study of the proposed method. The focus of this study is to show that the method can be employed to evaluate web services based on network latency quantiles.

We first discuss how to perform simulations of web downloads. Second, we introduce the simulation engine ns-2. Third, we review ns-2's support to simulation of TCP connections. Fourth, we describe how we implement and drive hyper text transfer protocol (HTTP) interactions on top of TCP.

5.1 Performing Simulations of Web Downloads

Accurately simulating web downloads to obtain realistic latency distributions is difficult given that web traffic patterns exhibit great variability. We thus assume that the system model underlying to the simulation includes the strategies described in [Paxson and Floyd(1997)], [Barford and Crovella(1998)],

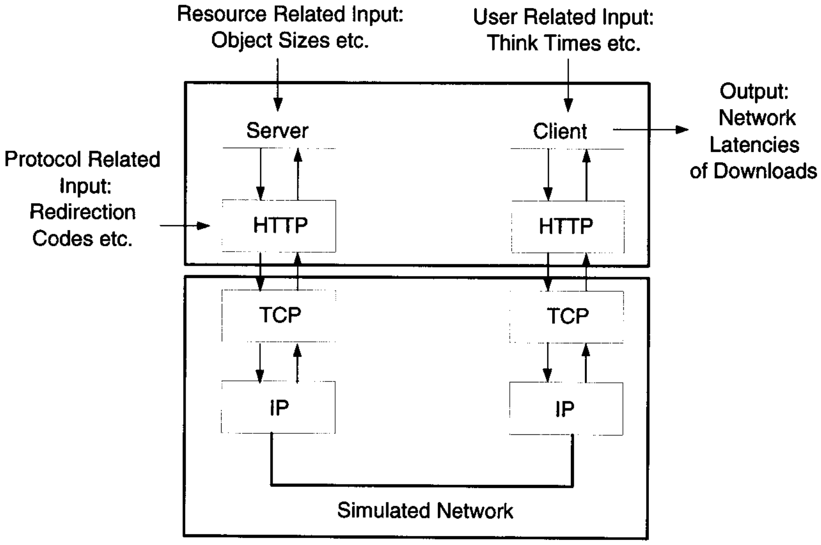


Figure 5.1: *System Model of the Simulation*

[Krishnamurthy and Rexford(2001)] to cope with this problem. In detail, this means that

- We assume that all input which drives the simulation is on the level of the application [Paxson and Floyd(1997)]. We assume that this input can be grouped into resource-related, protocol-related, and user-related input [Krishnamurthy and Rexford(2001)]. An example for resource-related input is the distribution of object sizes. An example for protocol-related input is the frequency of HTTP redirects. An example for user-related input is the distribution of think times.
- We assume that this input is modeled by sampling distributions which are given with a analytic formulae [Barford and Crovella(1998)]. These analytic formulae

capture well known invariants of the corresponding input variable. An example for such an invariant is heavy-tailedness with tail index α.

- We assume that interactions are explicitly modeled on the level of the application protocol [Paxson and Floyd(1997)]. Hence, HTTP interactions are explicitly modeled.

This leads to the system model depicted in Figure 5.1.

Moreover, we assume that implementation of the model accounts for the improvements of HTTP/1.1 [Fielding et al.(1999)] over HTTP/1.0 [Berners-Lee et al.(1996)] that optimize performance in terms of network latency. These improvements concern the TCP connection management of HTTP.

5.2 ns-2

We use ns-2 (network simulator version 2) [ns-2(2000)] as our simulation environment which we have enhanced with a HTTP implementation and instrumented that we can measure the network latencies of downloads. ns-2 is the open source, freely available discrete event simulator targeted at academic networking research [ns-2 Research)]. ns-2 can perform simulations of wired and wireless networks on on the level of IP packets and provides procedures to create and manage network topologies. ns-2 supports both shared media such as Ethernet as well as point-to-point connections. At connection endpoints, so called agents construct or consume IP packets which are transfered from source to destination. These agents simulate simple applications and can be enhanced to model application layer protocols such as HTTP. ns-2 offers substantial support for the simulation of routing and TCP.

Technically, ns-2 is an object-oriented simulator, written in C++ with an object-oriented tool command language (OTcl) interpreter as a front-end. Core low-level routines such as packet forwarding are implemented in C++ since this code rarely changes and needs to be run fast. High level routines such as the configuration and topology definition which change frequently and need not run fast are implemented in OTcl.

We implemented HTTP and started our simulations with ns-2.1b6a which was released in May 2000. Later, we adapted our HTTP implementation to ns-2.1b9a [ns-2 Change Log (2003)] which was released in July, 2002. The main reason for this adaptation was the integration of a "better" random number generator. In detail, the Park-Miller LCG16807 random number generator has been replaced with Pierre L'Ecuyer's MRG32k3a.

5.2.1 TCP Support

ns-2 offers substantial support for the simulation of TCP. In addition to the widely-used Reno version of TCP, ns-2 also supports SACK, Tahoe, and New Reno variants of TCP. The functionality implemented for each of these variants captures the essence of TCP's congestion and error control behavior which affects transmission latency. For our implementation we employ Reno TCP [Jacobson(1988)] which is based primarily on the 4.4BSD TCP implementation. Reno TCP in ns-2 is "bug-fixed" by Kathie Nichols and Van Jacobsen who authored the original BSD implementation (see comments in the code).

Technically, the endpoints of a TCP connection are modeled with so called TCP agents which emulate simple applications that use TCP. Every such agent has a C++ base class with an OTcl interface for configuration. This class contains a collection of routines for sending packets, processing ACKs, managing the send window, and handling timeouts. The base TCP agent solely supports unidirectional data transfer. The Reno 2-way FullTCP agent, which is derived from this agent, additionally supports bidirectional data transfer and delayed method invocation upon receipt of transfered data. We employ this delayed method invocation to explicitly implement the interactions of HTTP.

Further important properties and settings of FullTCP which are related to our simulation of web downloads are the following: FullTCP implements a complete 3-way-handshake for connection establishment. However, FullTCP does not support FIN bits for explicit connection tear down. FullTCP uses a signed integer to account for sequence numbers. Sequence number wrapping is not supported which results in a limitation of 2.1GB to the maximum transfer size. We have configured the maximum segment size (MSS), i.e. maximum amount of data the sender may send in a single packet, to 1000 bytes. FullTCP implements delayed acknowledgment. I.e. the receiver sends one acknowledgment for every two data packet it receives. The initial retransmission timeout used to estimate the round trip time (RTT) to the receiver is set to 3 seconds. This initial setting is used for the retransmission timer and defines the tradeoff between slow recovery from a lost TCP SYN packet against possible spurious transmissions. The time granularity to calculate RTT and retransmission timeout (RTO) is 100 ms. The initial congestion window size is set to two segments of size MSS. The receiver buffer size is set large enough that it does not restrict throughput.

5.2.2 HTTP support

Version ns-2.1b6a contains very limited support for simulation of web downloads. The support does not model the complete set of improvements in connection management of HTTP/1.1 [Fielding et al.(1999)] over HTTP/1.0 [Berners-Lee et al.(1996)]

that optimize download latencies. HTTP interactions are not driven with an analytic model as required in our system model but distributions which determine the size of web objects and think time between successive downloads are taken from a hard-coded table. This table represents the distributions from a limited set of measurements taken 1995 at Berkeley [Mah(1997)]. The largest object in these measurements is 1.6 MB. As a consequence, the traffic generated with this implementation is not very bursty thus misses a significant fraction of variability. Moreover, the implementation uses ns-2's default one-way TCP connection module which does not model the 3-way handshake for connection setup. This leads to inaccurate results for download latencies as losses of SYN packets in the 3-way handshake trigger long timeouts. We have thus decided to enhance ns-2 with our own implementation of web downloads.

5.2.3 A Short Review of HTTP/1.1's Connection Management

Before presenting this implementation we generally introduce the improvements in connection management of HTTP/1.1 over HTTP/1.0 which are not completely modeled in ns-2.1b6a. We use the example of a web page that consists of a container objects and five embedded objects. One of these embedded objects is located on a remote server. Figure 5.2 depicts the typical set of HTTP/1.0 interactions required to perform a download of this web page.

In the first phase of this download, the web client sends a request for the container object to the web server after establishing a TCP connection. The server typically responds with a reply that includes the container object after receiving and processing the request. In the second phase, after the client has received and processed the container object, the client starts sending further

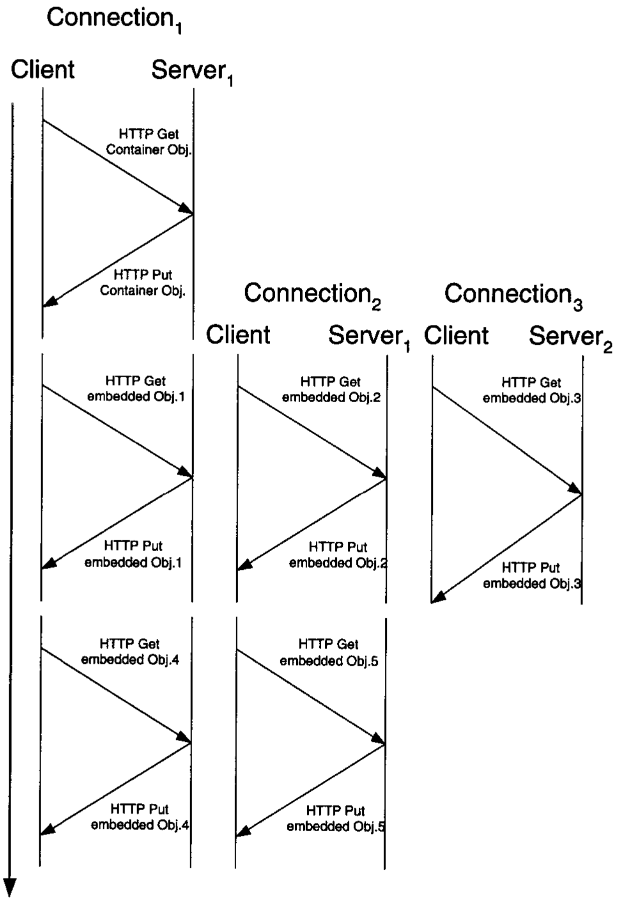


Figure 5.2: *Interactions of Download with HTTP/1.0*

requests for embedded objects of the web page. To handle the request, the client establishes a new TCP connection for each embedded object. The new connection is established although the embedded object, in many cases, is located on the same server as the container object or other embedded objects. To minimize the network latency of the download, a limited number of such TCP connections can be established in parallel. In our example this limit, which is to maintain fairness between different users, is set to three. Upon receipt of the request, the servers reply by sending the requested objects. Once one of these objects is received, the client sends a request for the next embedded object until all objects are requested.

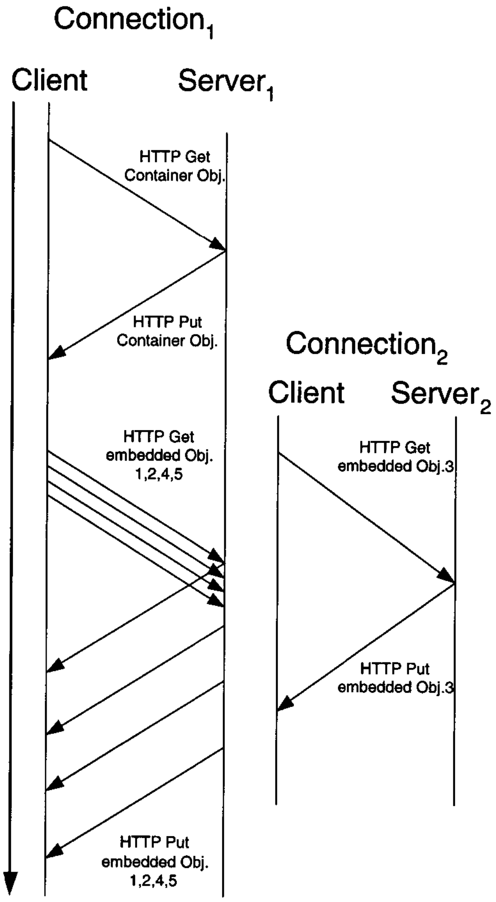


Figure 5.3: *Interactions of Download with HTTP/1.1*

Figure 5.3 depicts the typical set of HTTP/1.1 interactions to download the same web page. The first phase of the download, in which the contained object is downloaded, is the same as in HTTP/1.0. In the second phase, HTTP/1.1 implements two optimizations for connection management that affect the download latency:

- *Persistent connections*
  TCP connections between the client and a server that have been established for a HTTP request/reply interaction are kept open for reuse in further HTTP request/reply interactions. This prevents unnecessary TCP slow starts and thus inhibits the possibility of time consuming events such as the loss of a SYN packet triggering a time-out.
- *Pipelining*
  TCP connections between client and servers are immediately established. All requests for embedded objects are immediately send off after processing the container object.

5.3 Our Implementation of Web Downloads in ns-2

The key features of our implementation to support web downloads with HTTP/1.1 in ns-2 are:

1. We follow the system model depicted in Figure 5.1 and drive the HTTP interactions on the application layer by sampling analytic distributions.
2. We implement the complete set of HTTP/1.1 improvements in connection management that affect network latencies. This includes pipelining and persistent connections.

3. We employ ns-2's FullTCP which accurately models 3-way handshake for connection setup.

The goal of this implementation is to research network latencies of web downloads during times of peak usage.

The important random variables which drive the simulation and their distribution including parameters were chosen as follows:

5.3.1 Resource-related Input Variables

The resource-related variables were chosen such that the full hierarchical structure of web resource can be modeled. Hence, resource-related variables pertain to

- size of container objects
- size of embedded objects
- number of embedded objects

Default values for distributions including parameters are listed in Table 5.3.1. The parameters in the functions have the same values as in [Feldmann et al.(1999)]. Real numbers obtained by the random number generator are truncated to integer.

Parameter	Distribution	Average	Shape
size of container obj.	ParetoII	12KB	1.2
size of embedded obj.	ParetoII	12KB	1.2
number of embedded obj.	ParetoII	3	1.5

Table 5.1: *Resource Related Parameters to Generate Web Traffic*

No special effort is taken to model the locality of objects which may considerably affect server performance since the focus of the implementation is research of network latencies. Instead, we follow the approach of [Fiedler(2001)] and chose servers randomly.

5.3.2 User-related Input Variables

For user-related variables, our implementation restricts to modeling the think time of successive downloads after which the next transaction is triggered. The default values for distribution of think times including parameters are listed in Table 5.3.2. The shape parameter of the ParetoII distribution of think time, which is the significant parameter that impacts the traffic characteristic, has the same values as in [Feldmann et al.(1999)]. The parameter for the average in this function, which can be used to adjust the network utilization, has a default value of 40 seconds.

Parameter	Distribution	Average	Shape
Think Time	ParetoII	40 sec	2.0

Table 5.2: *User Related Random Variables to Generate Web Traffic*

More coarse grained user-related variables such as session duration, which exceed the scope of investigating latencies during time of peek usage, are not modeled.

5.3.3 Protocol-related Input Variables

Protocol-related variables which primarily affect error handling, redirection etc. are not modeled.

5.3.4 Our Implementation of HTTP

To implement HTTP interactions we exploit that ns-2's FullTCP module supports bidirectional data transfer with the possibility to invoke a command at the receiver side upon receipt of all data. We thus employ mutual method invocation between client and server to implement HTTP interactions. The connection management of HTTP/1.1 including persistent connections and pipelining can then be added by method derivation.

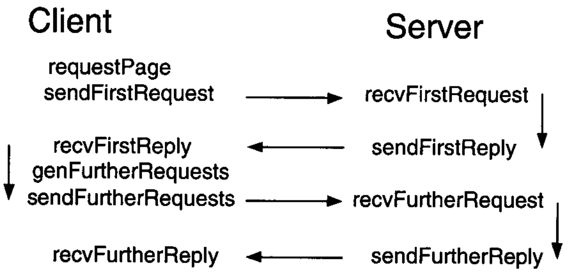


Figure 5.4: *Method Invocations of a Download (Example)*

Our implementation can be explained as follows: All bookkeeping associated with the download of a web page is done in a dynamic object which is created at the start of a download and freed upon termination of the download. This bookkeeping includes the status of the download as well as locations where embedded objects have to be requested which is known as soon as the container object has been received. The sequence of mutual method invocations that implement the interactions depicted in Figure 5.3 are listed in Figure 5.4. When the think time of this client is over the client method *requestPage* calls the client method *sendFirstRequest* which simulates the sending of the HTTP GET of the container object. This *sendFirstRequest* method employs FullTCP to simulate a data transfer to the server and calls the server method *recvFirstRequest* when the server has received the data. Then the server determines the size of the container object and number, size, and location of embedded objects by sampling the distribution of the corresponding variables. Moreover, the server calls its *sendFirstRequest* method which simulates the sending of the HTTP PUT of the contained object. This *sendFirstRequest* method in turn employs FullTCP to simulate a data transfer to the client and calls the client method *recvFirstReply*. This client method *recvFirstReply* calls *genFurtherRequests* which updates the bookkeeping and calls *sendFurtherRequests* to simulate the sending of HTTP GETs for the embedded objects. This *sendFurtherRequests* method employs FullTCP to simulate the data transfers to the server(s) and calls the server method *recvFurtherRequest* when the corresponding server has received the data. Each call of the *recvFurtherRequest* method in turn calls *sendFurtherReply* which simulates the sending of a HTTP GET of an embedded object. This *sendFurtherReply* method in turn employs FullTCP to simulate a data transfer to the client and calls the client method *recvFurtherReply* which updates the bookkeeping.

The connection management of HTTP/1.1 is implemented in two steps with deriving a set of methods at the client's side. The first step adds persistent connections. The second step adds pipelining of requests. In the first step, a container that enables reuse of previously opened connections is added to each client to keep track of persistent connections. A derived client method *genFurtherRequests* searches this container before sending a request. Methods at the server side do not need to be modified to implement persistent connections. The server just reuses the connection of the request to send the reply. In the second step, pipelining, i.e the immediate sending off of requests for embedded objects, is added by deriving the client method *genFurtherRequests*.

5.3.5 Connection Setup

Employing FullTCP solves the problem of accurately modeling the 3-way handshake at connection setup.

5.3.6 Interface for Testing

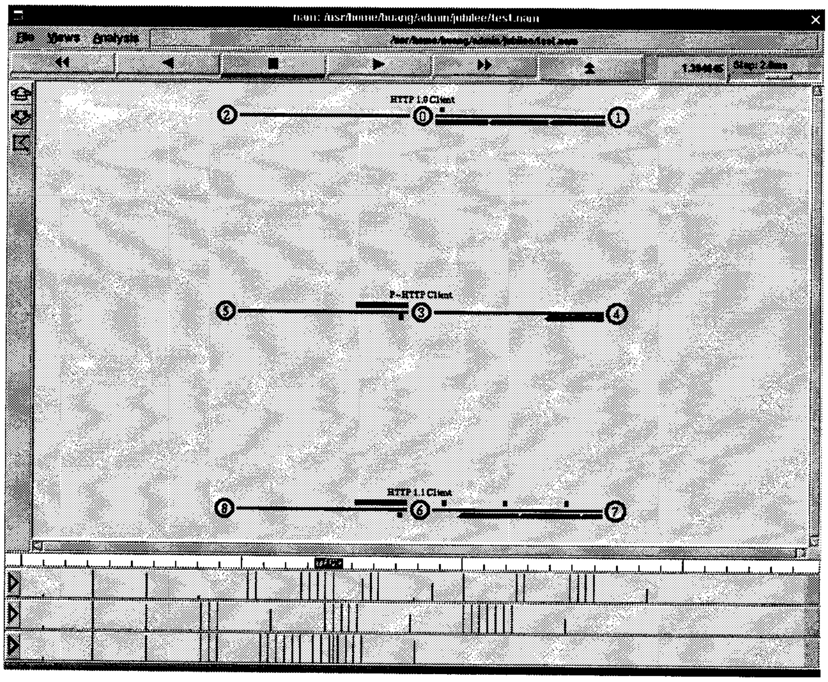

nam: /usr/home/huang/admin/jubilee/test.nam
File
Views
Analysis
/usr/home/huang/admin/jubilee/test.nam
Step: 2.0ms
HTTP 1.0 Client
0
1
2
P-HTTP Client
3
4
5
HTTP 1.1 Client
6
7
8

Figure 5.5: *Visualizing Connection Management*

For test purposes the HTTP interactions in our implementation can be visualized with the network animator (nam) [nam(2003)]. We have analyzed a number of downloads in different setups to verify the correctness of our implementation. Figure 5.5 depicts a screen shot that has been made when testing the download of a single web page with multiple embedded objects with HTTP/1.0, persistent HTTP, and HTTP/1.1.

5.4 Summary

In this chapter we have described the simulation environment which we employ to verify the applicability of the evaluation method proposed in chapter 4. We have discussed the underlying system model of our simulation of web services. We have described our implementation on top of ns-2 to simulate web services with HTTP/1.1 which is targeted at research of network latencies.

Chapter 6

Evaluation of Simulation Input

In this chapter, we start with the validating the proposed method and analyze the convergence of input in a simulation of web services. We justify that it is sufficient to focus this analysis on the input which is obtained by sampling the long- or heavy-tailed object size distribution in workload generation. We show how the method, which is described in Section 4.1, can be employed to evaluate the minimal sample size required to converge the 99-th, 98-th, and 95-th object size percentiles in simulation input. We show that for our simulation these sample sizes are magnitudes smaller than the sample size required to converge the running average of object sizes in simulation input. We further show that this difference in sample sizes also holds under small changes to the parameters of the object size distribution used in workload generation. Moreover, we discuss under which conditions these sample sizes also approximate the sample size required to converge the 99-th, 98-th, and 95-th network latency percentile of web downloads in simulation output.

6.1 Properties of the Workload Generation Process

This analysis of convergence of simulation input exploits that underlying model of the simulation is as depicted in Figure 5.1. In detail this means:

1. The variables in web traffic generation process pertain to resource-related variables such as the size of container and embedded objects, and the number of embedded objects in a web page, user-related variables such a the think time between successive downloads, and HTTP-protocol related variables which are not modeled in our implementation.
2. All input variables, including the object sizes, are obtained by independently sampling analytic distributions. These analytic distributions reflect the important characteristics of web traffic such as heavy-tailedness of object size distributions.

We assume that it is the heavy-tailed distributed variables in simulation input which determine the convergence properties of simulation input. This assumption can be justified with [Willinger et al.(1997)] [Park et al.(1996)] reporting that heavy-tailed distributed variables are the essential cause for the great variability and the self-similarity of web traffic. We further make the technical assumption that heavy-tailed distributed system input is such that is has a continuous first derivative and is strictly monotonous for all $x > 0$. This is e.g. the case for a Pareto or ParetoII distribution. As a consequence, the sample size required to converge quantiles from such distributions can be evaluated with the method proposed in section 4.1 which is based on Theorem 2 and Theorem 3.

The heavy-tailed distributed variables in our simulation are

- the distribution of object size for container and embedded objects.
- the distribution of think times between successive downloads.

[Park et al.(1996)] reports that effects of variability from the heavy-tails of the object size distributions clearly dominate the effects from the heavy-tail in the think time distribution. This can be explained with the fact that the tail indices of object size distributions are significantly smaller than the tail index of the think time distribution. Hence, we focus our analysis of convergence of simulation input on the distributions of object sizes.

6.2 Object Size Distribution

We exploit that both embedded and container objects are modeled with the same ParetoII object size distribution which is given with

$$F(x) = 1 - \frac{1}{(1+\frac{x}{s})^{\alpha}} \qquad x \in [0, \infty[\tag{6.1}$$

The parameters of this distribution are listed in Table 6.1. We denote that average and shape parameters are independent parameters where as $s = a * (\alpha - 1)$ is a dependent parameter.

Parameter	Value
average a	12KB
shape parameter α	1.2
s	2400B

Table 6.1: *Parameters for the ParetoII Object Size Distribution*

6.3 Object Size Quantiles

We assume that object sizes in simulation input are independently sampled from this ParetoII distribution. Hence, we can employ part one of the proposed method, which is described in section 4.1, to evaluate the minimal sample sizes required to converge then 95-th, 98-th, 99-th, 99.9-th, and 99.99-th object size percentiles in simulation input.

Percentile	Object Size
95-th	27 KB
98-th	60 KB
99-th	110 KB
99.9-th	760 KB
99.99-th	5.2 MB

Table 6.2: *Percentiles of the Object Size Distribution*

As a first result we can give percentiles of the object size distribution x_p which are evaluated by setting $x_p = F^{-1}(p)$. For the ParetoII object size distribution given in Equation 6.1 the percentiles can be evaluated from

$$F^{-1}(p) = s * \left(\frac{1}{(1-p)^{1/\alpha}} - 1 \right)$$

Values for the 95-th, 98-th, 99-th, 99.9-th, and 99.99-th object size percentiles are listed in Table 6.2

6.3.1 Sample Sizes to Converge Object Size Quantiles

Next, we can give the minimal sample sizes required to converge these object size quantiles to a given accuracy.

We apply the proposed method and start with estimating these sample sizes with

$$\left(\frac{1.96}{x_p}\right)^2 \frac{p(1-p)}{f^2(x_p)} \frac{1}{accuracy^2} \leq n \tag{6.2}$$

This inequation follows from a large sample approximation of the distribution of sample's quantiles (see section 4.1.2 for a derivation).

We improve this estimation with the accurate evaluation as proposed in Section 4.1.1. We thus iterate the sample size n and evaluate the following integral relations which determine the confidence interval for the sample's quantile until the accuracy relation 2.10 for the sample's quantile leads to the given accuracy.

$$\int_0^{L_n} f_n(x)dx \doteq \frac{1 - confidence\,level}{2} \doteq 0.025 \tag{6.3}$$

for the lower bound L_n of the confidence interval and

$$\int_0^{U_n} f_n(x)dx \doteq 1 - \frac{1 - confidence\,level}{2} \doteq 0.975 \tag{6.4}$$

for the upper bound U_n. Here f_n is the probability density of the sample's quantile as given in Equation 3.5, the confidence level is assumed to be 95%. E_n in the accuracy relation is given with $F^{-1}(p)$ since the sample's quantile's expected value equals to the corresponding quantile of the distribution the samples. Values for estimation of the 95-th, 98-th, 99-th, 99.9-th, and 99.99-th object size percentiles at a 5% accuracy are listed in Table 6.4. Values for estimation of the 99-th object size percentile at various accuracies are listed in Table 6.3. These values maximally differ by 0.2 in the mantissa from the values estimated with Inequation 6.2. Hence, the error that comes from approximating the sample's quantile's distribution with a normal distribution is negligible.

Accuracy of the 99-th percentile	Sample Size
1%	$2.8 \cdot 10^6$
2%	$7.2 \cdot 10^5$
3%	$3.2 \cdot 10^5$
5%	$1.2 \cdot 10^5$
10%	$3.1 \cdot 10^4$

Table 6.3: *Sample Size Required to Estimate the 99-th Object Size Percentiles*

Percentile (5% Acc.)	Sample Size
95-th	$2.6 \cdot 10^4$
98-th	$6.0 \cdot 10^4$
99-th	$1.2 \cdot 10^5$
99.9-th	$1.2 \cdot 10^6$
99.99-th	$1.1 \cdot 10^7$

Table 6.4: *Sample Size Required to Estimate further Object Size Percentiles*

6.3.2 Sample Sizes to Converge the Running Mean

These sample sizes to converge the 95-th, 98-th, and 99-th object size quantile are magnitudes smaller than the sample sizes required to converge the running average of the object size distribution. This can be seen by adopting the argument of [Crovella and Lipsky(2000)] which we review in section 2.3. This argument yields in the following inequation for estimation of the sample size required to converge the running average from a heavy-tailed distribution with tail index α with k digit accuracy

$$(\frac{\mu}{c_1} * 10^{-k})^{-\frac{1}{1-1/\alpha}} \leq n \tag{6.5}$$

for which [Crovella and Lipsky(2000)] suppose that $\frac{\mu}{c_1} \approx 1$. Hence, to estimate sample size for any accuracy not just k digit accuracy the inequation can be stated as:

$$accuracy^{-\frac{1}{1-1/\alpha}} \leq n \tag{6.6}$$

Values for estimation of the running sample average from the object size distribution at various accuracies are listed in Table 6.5.

Accuracy	Sample Size
1%	$1.0 \cdot 10^{12}$
2%	$1.5 \cdot 10^{10}$
3%	$1.4 \cdot 10^{9}$
5%	$6.4 \cdot 10^{7}$
10%	$1.0 \cdot 10^{6}$

Table 6.5: *Sample Size Required to Estimate the Average Object Size (adapted from [Crovella and Lipsky(2000)])*

These values for the sample size required to estimate the running average are magnitudes larger than the values required to estimate the 95-th, 98-th, and 99-th percentile from the object size distribution. This large difference can be explained with the fundamental difference in convergence of the running average to a α-stable distribution at a $n^{1/\alpha-1}$ rate versus the convergence of the quantiles to a normal distribution at a $n^{-1/2}$ rate. Here n is the sample size and α the tail index of the object size distribution.

6.3.3 Introducing System Limits

This large difference continues to hold at the presence of realistic bounds to the object size distribution inherent to common operating systems. We show that this is the case with a 2.1GB limit to the object size distribution. The choice of this limit to 2.1GB can be explained with the fact that signed integers in our simulation are represented with 32 bits and $2^{31} = 2.1 \cdot 10^9$.

The sample size required to converge the 95-th quantile remains unchanged by introducing a 2.1GB limit. This can be explained with the fact that $1 - F(2.1GB) = 10^{-7}$. Hence, the sample size to required converge the 95-th percentile is practically equal to the sample size required to converge the $[100 * (0.95 + 10^{-7}])$-th percentile. A similar statement can be made for the 98-th and 99-th quantile.

The sample size required to converge the running average after introducing a limit can be estimated with the central limit theorem (CLT). CLT states that the running average in i.i.d samples of size n from a distribution with two finite moments converge to a normal distribution

$$n^{1/2}(\overline{X}_n - \mu) \xrightarrow[in\ distribution]{} \mathcal{N}(0, \sigma^2) \tag{6.7}$$

Thus, at sample size n the confidence interval for the running average is given with

$$\mu \pm 1.96 \frac{\sigma}{\sqrt{n}} \tag{6.8}$$

and the accuracy relation is given by

$$accuracy_n = \frac{1.96 \frac{\sigma}{\sqrt{n}}}{\mu} \tag{6.9}$$

Thus solving for n yields in

$$\left(\frac{1.96 * \sigma}{\mu * accuracy} \right)^2 \leq n \tag{6.10}$$

The parameters σ and μ of the asymptotic normal distribution can be evaluated with the standard formulae. Evaluating these formulae with numeric integration leads to $\sigma^2 = 4.9 \cdot 10^{11}$ and $\mu = 11067$ for the ParetoII object size distribution with a 2.1GB limit. Inserting these values for σ and μ in Inequation 6.10 leads to the sample sizes listed in Table 6.6.

Accuracy of the sample's average	Sample Size
1%	$1.5 \cdot 10^8$
2%	$3.8 \cdot 10^7$
3%	$1.7 \cdot 10^7$
5%	$6.1 \cdot 10^6$
10%	$1.5 \cdot 10^6$

Table 6.6: *Sample Size Required to Estimate the Average Object Size (2.1GB limit)*

6.3.4 Sensitivity to Small Changes in Tail Index

The difference in the sample size required to converge the 99-th quantile and the running average gets more emphasized for tail index $\alpha \to 1$ (See Table 6.7 which has been obtained by repeating the evaluation of section 6.3.1 and section 6.3.2 for 5% accuracy). From this table we infer that the sample size required to converge the 99-th percentile from a ParetoII distribution do not significantly change when the tail index α varies between 1.1 and 1.3 This is not the case for the running average. The sample size required to converge the running average largely increases for tail index $\alpha \to 1$.

α	99-th Percentile	Average (w/o limit)	Average (2.1GB limit)
1.1	$1.3 \cdot 10^5$	$2.0 \cdot 10^{14}$	$1.4 \cdot 10^{12}$
1.2	$1.2 \cdot 10^5$	$6.4 \cdot 10^7$	$6.1 \cdot 10^6$
1.3	$9.1 \cdot 10^4$	$4.3 \cdot 10^5$	$2.9 \cdot 10^6$

Table 6.7: *Sample Sizes at Various Tail Indices α (5% Accuracy)*

6.4 Special Cases

The sample sizes to converge the p-th object size quantile in simulation input, which are listed in Table 6.3 and Table 6.4, is not only necessary but also sufficient to converge the corresponding p-th latency quantile if network bandwidth is large enough to prevent packet losses. This can be explained with the fact that a normal distribution is conserved under the linear transformation which in this case describes the object size latency relationship in the vicinity of the p-th latency quantile.

Assuming that TCP's congestion control and retransmission algorithm have an adverse impact on convergence implies that it is reasonable to assume that the sample sizes listed in Table 6.3 and Table 6.4 are lower bounds to the sample sizes required to converge the corresponding p-th quantile in simulation output.

6.5 Summary

In this chapter, we have started to validate the proposed method. We have analyzed the convergence of input a simulation of web services. We have justified that it is sufficient to focus this analysis on the input which is obtained by sampling the long- or heavy-tailed object size distribution in workload generation. We have determined the minimal sample size required to converge the 99-th, 98-th, and 95-th quantile in the sample from the object size distribution in simulation input. For simulations such sample sizes are feasible in practice. We have found that the sample sizes do not highly dependent on small changes in the tail index α of the object size distribution such as $\alpha = 1.1$ instead of $\alpha = 1.2$. We have shown that this is in contrast to the sample size required to converge the running average from the object size distribution.

This sample size largely grows if the tail index α decreases from 1.2 to 1.1.

Chapter 7

Evaluation of Simulation Output

In this chapter, we continue with the validation study of the proposed method and analyze the convergence of output in a simulation of a web services. We show that download latency quantiles such as the 99-th, 98-th, and 95-th percentile can converge within sample sizes that are feasible for simulations and can thus be used to evaluate network performance. The system model and environment for the simulation study is as described in Chapter 5.

7.1 Simulation Study

7.1.1 Workload Generation

We take two steps to show that the method can be employed for evaluation. We first show that the method allows to evaluate network performance under the assumption that each download consists of a single object before we remove this simplifi-

Name of Set	Object Size Distribution	Embedded Objs. Per Page Dist.	Think Time Distribution
Coarse Model	ParetoII Average 12 KB Shape 1.2	None	ParetoII Average 10s Shape 2.0
Accurate Model	ParetoII Average 12 KB Shape 1.2	ParetoII Average 3 Shape 1.5	ParetoII Average 40s Shape 2.0

Table 7.1: *Probability Distributions to Generate Web Traffic*

cation in a second step. This single object assumption allows us to directly associate object sizes with download latencies which facilitates the analysis of results. The specific choice of parameters for web traffic generation under this assumption is depicted in Table 7.1.1, first row. We refer to the underlying model for workload generation as the "coarse model". In the second step we remove this single object assumption and chose parameters such that they reflect the full structure inherent to web pages (see Table 7.1.1, second row). We refer to the underlying model for workload generation as the "accurate model". To obtain comparable results with both models, we have adjusted the parameters for the coarse model in the table such that the network utilization for traffic generated with both models are comparable. With *network utilization* we mean the amount of traffic transported over the network per time unit in proportion to the capacity of the bottleneck link. In detail, we have adjusted the average think time in the coarse model so that the ratio

$$\frac{average\ number\ of\ objects\ *\ average\ object\ size}{average\ think\ time} \qquad (7.1)$$

which determines the average network utilization remains constant.

7.1.2 Topology

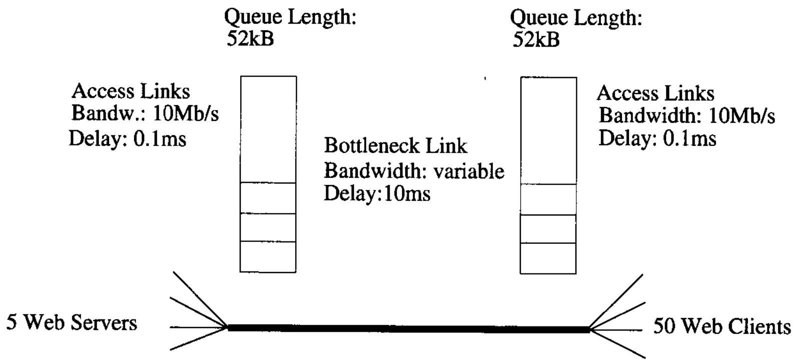


Figure 7.1: *Validation Topology*

	Capacity	Avg. Utilization	Loss Rate
High utilization	640Kb/s	64%	0.8%
Medium utilization	2560Kb/s	17%	≤ 0.1%
Low utilization	6400Kb/s	7.0%	≤ 0.1%

Table 7.2: *Avg. Utilization of the Bottleneck Link*

The topology in the validation study and the location of web clients and servers is as depicted in Figure 7.1.2. The bottleneck link in this topology represents some critical link in a network. The bottleneck link can also be viewed in more general sense, since it has been argued that there is always a single bottleneck link on any network path [Bajaj et al.(1998)]. All access links to the bottleneck have a capacity of 10Mb/s and a propagation delay of 0.1ms. We later vary this propagation delay of access links to explore whether the evaluation method can be applied without assuming that the physical propagation delay between all clients and servers is equal. The bottleneck link has a propagation delay of 10ms. This can be seen as an abstraction from multiple links in a well provisioned network where delay adds up to 10ms or as a worst case of a transcontinental or transoceanic link. Queue sizes are set to 52KB. We vary the capacity at the bottleneck link to explore under which link utilizations our method can be applied to estimate network performance. We consider three cases for the capacity of the bottleneck link: 640Kb/s, 2560Kb/s, and 6400Kb/s. The 640Kb/s case leads to a average utilization of slightly more than 60% (see Table 7.2). This average utilization is known as an upper limit to what's acceptable during the busiest period (see [Ben Fredj et al.(2001)] on provisioning procedures). We then lower the average link utilization to values that are typical for data networks [Odlyzko(2000)]. We refer to the 640Kb/s case as *high utilization*, to the 2560Kb/s case as *medium utilization*, to the 6400KB/s case as *low utilization.*

Given that the focus is on network performance, we randomly chose the server for each web request and assume that embedded objects are located on the same server as the container objects. Default values are used for all other configurations of the simulation (see chapter 5 for details).

7.1.3 Simulation Duration

We run very long simulations. When employing the coarse workload model simulations terminate after the first 500,000 requested objects have been completely downloaded, which corresponds to 28 hours of simulation time. Typically more than 500,000 objects have been completely downloaded during this period. This period is sufficient to converge the 95-th, 98-th, and 99-th object size percentiles in simulation input such that they can be estimated with a 3.6% accuracy or better (c.f. table 6.3). When employing the accurate workload model simulations terminate after the first 120,000 requested web pages have been completely downloaded. These web pages contain approximately 500,000 web objects. Hence, this period is also sufficient to converge the 95-th, 98-th, and 99-th object size percentiles in simulation input such that they can be estimated with a 3.6% accuracy or better.

7.1.4 Limitations

The object size distribution which determines traffic burstiness is truncated at 2.1GB since object sizes in our simulation environment are represented with 32 bit signed integers. We denote that this differs from the 4GB limitation in common operating systems which represent object sizes with unsigned integers. Moreover, the simulation environment imposes a 2.1GB limit on the amount of data that can be transported over a TCP connection to avoid a wrap over of the sequence number which is also represented with a 32 bit integer. Therefore, given that we implement downloads with HTTP/1.1 which uses persistent TCP connections, we have to limit the total size of web pages including container and embedded objects to 2.1GB. We have verified that we have chosen simulation duration and number of simulation runs large enough that objects and pages of this size occur in our simulations.

7.1.5 Evaluation of Network Latency and Latency Quantiles

We assume that the *network latency* of a web download is defined as the time that has elapsed between the following start and stop event. The start event is the begin of the send process of the first TCP *syn* packet associated with the first corresponding HTTP request. The stop event is the end of the reception process of the last TCP packet containing relevant data associated with the any of the corresponding HTTP replies.

The p-th network latency quantile (NLP) associated with each of the simulation runs is then computed as follows: We take the network latency of downloads in the order with which the corresponding web requests were issued. All latencies are sorted from shortest to longest, i.e. $NL_1, .., NL_n \rightarrow NL_{(1)}, .., NL_{(n)}$. The p-th latency quantile is then given by $NL_{(k)}$ with $k = \lceil np \rceil$.

7.2 Results

In this section, we present the results from applying the method described in Section 4.2 to assess the convergence of 99-th, 98-th, and 95-th network latency percentiles in simulation output.

We recall that the method exploits that it is reasonable to assume that latency quantiles in simulation output generally converge to a normal distribution which we have explained in Section 3.3.1. When observed latencies have a weakly dependent correlation structure this convergence is at a $n^{-1/2}$ rate where n is the sample size. When observed latencies have a long-range dependent correlation structure this convergence is at a n^{H-1} rate where $1/2 \leq H < 1$ is the Hurst parameter.

Hence, the focus of investigation is whether we can observe a convergence of the 99-th, 98-th, and 95-th network latency percentiles at sample sizes which occur in our simulations.

7.2.1 Low Utilization

At low utilization we can observe that all latency quantiles investigated converge to a normal distribution at rate $n^{-1/2}$ for both the coarse and the accurate workload generation model. The sample size required to estimate a latency quantile from simulation output is comparable to the sample size required to estimate the corresponding object size quantile in simulation input.

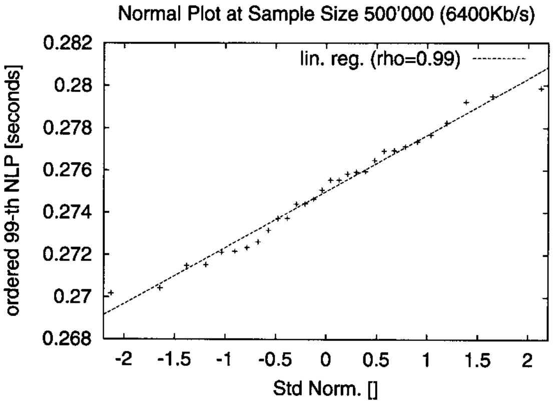


Figure 7.2: *Normal Plot (99-th NLP, Coarse Model, Low Util.)*

We start with presenting results from the evaluation of 30 simulation runs obtained with the coarse model for workload generation. This model allows us to directly compare the convergence of latency quantiles and corresponding object size quantiles. Figure 7.2 depicts a normal plot produced from the 99-th network latency percentiles of the 30 simulation runs. This normal plot results in a straight line and shows no indications that the distribution of these latency percentiles deviates from a normal distribution. This is also reflected in the correlation coefficient which has been obtained by applying linear regression to the points of the plot. The correlation coefficient is 0.99. The intercept and slope of the linear regression, which are 0.275 seconds and 0.0027 seconds, lead to estimations for the parameters of the underlying normal distribution. These parameters in turn lead to an estimation of the 99-th latency quantiles which is $0.275 \pm 1.96 * 0.0027$ seconds which is accurate up to 1.9%.

In order to investigate the minimum sample size required for

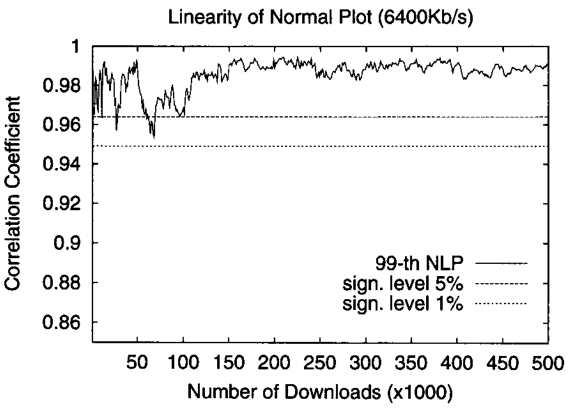


Figure 7.3: *Normality Test (99-th NLP, Coarse Model, Low Util.)*

convergence we produce normal plots for increasing sample size. We successively employ linear regression to the points of these normal plots to test whether the normal plots result in a straight line which indicates that the latency quantiles follow a normal distribution. Figure 7.3 depicts the correlation coefficients from these normal plots together with critical values for the hypothesis that the latency quantiles follow a normal distribution. The upper line depicts the critical value for the correlation coefficient at a 5% significance level given in Table 4.1. This means that if quantiles are normally distributed, we expect 5% of the correlation coefficients from normal plots to be smaller than this value. The lower line reflects a 1% significance level given in Table 4.1. This means that if quantiles are normally distributed, we expect 1% of the correlation coefficients from normal plots to be smaller than this value. From Figure 7.3 we come to the conclusion that the 99-th latency percentiles in this simulation converge to a normal distribution for sample sizes larger than 110'000.

From Figure 7.4 we infer that the rate of this convergence is $n^{-1/2}$. The figure plots the product $s * \sqrt{n}$ vs. n with n being the

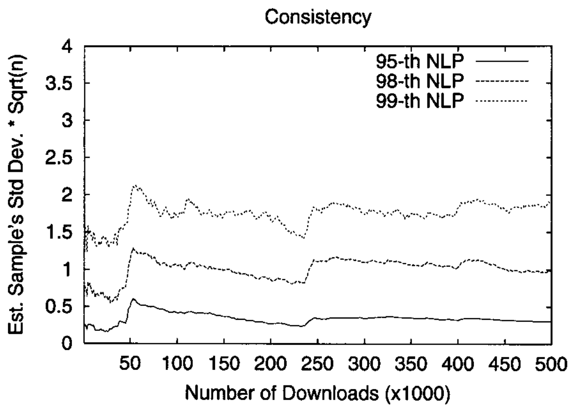


Figure 7.4: *Consistency of Convergence (Coarse Model, Low Util.)*

sample size and s being the estimated standard deviation, which is given by the slope of the linear regression in the normal plot. Comparing this figure to the corresponding figure for simulation input (see appendix D Figure D.1) leads to the conclusion that the product is constant within the accuracy of the simulation. The relative deviation around sample size $240,000$ is not larger than the corresponding relative deviations in simulation input which by probability theory must be constant. Hence, we infer that this convergence is consistent with a weakly-dependent correlation structure in observed latency quantiles.

Finally, we list minimal sample sizes required to estimate the 99-th latency quantile at a given accuracy (see Table 7.3). These sample sizes were determined as follows: The definition for accuracy is given in Equation 2.10. The normal plot at 500k downloads leads to an estimation of the expected 99-th latency quantile of $\mu = 0.275$ seconds. This estimation equals up to three digits to the estimation from any other normal plot between 110k and 500k downloads. Based on the consistency check we can get a robustified estimation of the radius of the confidence interval in

Accuracy of the 99-th percentile	Sample Size
1%	$3 \cdot 10^6$
2%	$7 \cdot 10^5$
3%	$3 \cdot 10^5$
5%	$1 \cdot 10^5$
10%	$3 \cdot 10^4$

Table 7.3: *Sample Size Required to Estimate the 99-th NLP (Coarse Model, Low Util.)*

which we expect the latency quantile. Exploiting that the convergence is to a normal distribution at rate $n^{-1/2}$, the radius of the confidence interval can be estimated with:

$$confidence\,interval\,radius = 1.96\,\frac{\sigma}{\sqrt{n}} \tag{7.2}$$

where the parameter σ of the normal distribution can be conservatively estimated from the consistency check as $\sigma \leq 2.3$ seconds. Hence, inserting Equation 7.2 into the accuracy relation given in Equation 2.10 and solving for n yields in

$$\left(\frac{1.96 * \sigma}{\mu}\right)^2 \frac{1}{accuracy^2} \leq n \tag{7.3}$$

We can thus determine the sample sizes required to estimate the 99-th latency quantile from simulation output. Results are listed in Table 7.3. The sample sizes turn out to be comparable to the sample sizes required to estimate the corresponding object size quantiles in simulation input (see Table 6.3).

Similar observations can be made for the convergence and consistency of convergence of the 98-th and 95-th latency percentile (see appendix D).

We therefore repeat the simulations for low utilization with the accurate model for web traffic generation. We perform 30

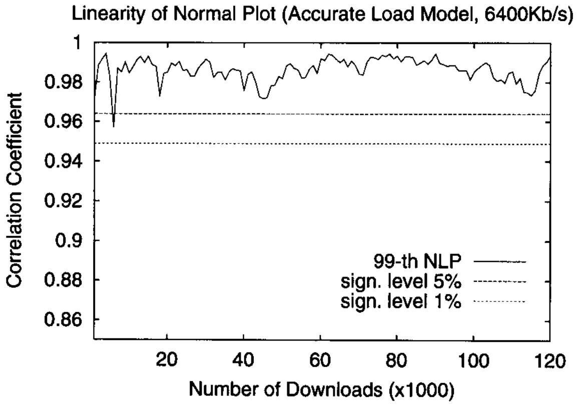


Figure 7.5: *Normality Test (99-th NLP, Accurate Model, Low Util.)*

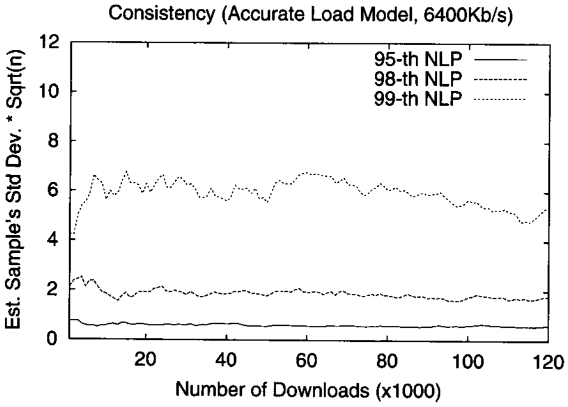


Figure 7.6: *Consistency of Convergence (Accurate Model, Low Util.)*

simulation runs with different seeds and terminate after the first 120,000 requested web pages have been completely downloaded. Results for the convergence of 99-th latency quantiles and consistency of the convergence are depicted in Figure 7.5 and Figure 7.6. The normal plot at 120,000 downloads leads to an estimation of the 99-th latency percentile of 0.77 ± 0.03 seconds which is accurate up to 4%. Conservatively estimating $\sigma \leq 6.8$ seconds from the consistency check (see Figure 7.6) and applying Equation 7.3, we can determine the samples sizes required to estimate the 99-th latency percentile from this simulation (see Table 7.4). The magnitude of the sample sizes listed in Table 7.4 is comparable the magnitude of sample sizes required to estimate the corresponding object size quantiles in simulation input (see Table 6.3).

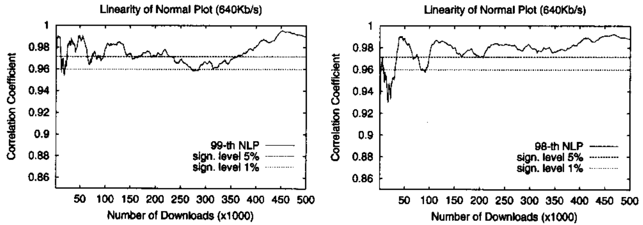
Accuracy of the 99-th percentile	Sample Size
1%	$3 \cdot 10^6$
2%	$8 \cdot 10^5$
3%	$3 \cdot 10^5$
5%	$1 \cdot 10^5$
10%	$3 \cdot 10^4$

Table 7.4: *Sample Size Required to Estimate the 99-th NLP (Accurate Model, Low Util.)*

Similar observations can be made for the convergence and consistency of convergence of the 98-th and 95-th latency percentile (see appendix D).

7.2.2 High Utilization

At high utilization we observe for the coarse workload generation model that the 99-th and 98-th latency percentile converge to a normal distribution at a rate which is no more consistent with weak dependence. The 95-th latency quantile does not converge during simulation times that we have done. The order of magnitude of the sample sizes required to estimate a latency quantile from simulation output is still comparable to the order of magnitude of the sample size required to estimate the corresponding object size quantiles in simulation input.



a) 99-th NLP b) 98-th NLP

Figure 7.7: *Normality Test (Coarse Model, High Util.)*

Results presented here are from the evaluation of 40 simulation runs. Figure 7.7 depicts the correlation coefficients from normal plots for increasing sample size together with critical values for the hypothesis that the latency quantiles follow a normal distribution. The upper line depicts the critical value for the correlation coefficient at a 5% significance level. This means that if quantiles are normally distributed, we expect 5% of the correlation coefficients from normal plots to be smaller than this value. The lower line reflects a 1% significance level. This means that if quantiles are normally distributed, we expect 1% of the correlation coefficients from normal plots to be smaller than this value. From Figure 7.7 we come to the conclusion that the 98-th latency percentiles in this simulation converge to a normal distribution for sample sizes larger than 100’000. The 99-th latency percentile presumably converges for sample sizes larger than 350’000.

Estimating the 99-th and 98-th latency quantiles from the intercept and slope of the linear regression of the normal plot at sample size 500’000 lead to 7.37 ± 0.45 seconds for the 99-th latency percentile and 5.11 ± 0.68 seconds for the 98-th percentile. However, the convergence of the 99-th and 98-th latency per-

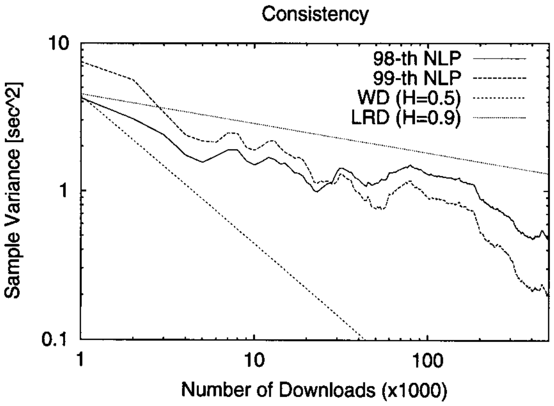


Figure 7.8: *Consistency of Convergence (Coarse Model, High Util.)*

centile is not consistent with a $n^{-1/2}$ rate (see Figure 7.8 which depicts a log-log plot of sample variance of the quantiles vs. sample size). Convergence at a $n^{-1/2}$ rate would result in a line parallel to the reference line entitled with Hurst parameter $H = 0.5$ which is clearly not what we observe. The convergence is also not completely consistent with a slower rate n^{1-H} with Hurst parameter $1/2 < H < 1$ which is expected for a long-range dependent correlation structure among the observations of latency quantiles. Such a correlation structure would result in a straight line with smaller slope (see e.g. the reference line for Hurst parameter $H = 0.9$ which is to be expected for the corresponding on/off process with unbound heavy-tailed input with tail index $\alpha = 1.2$ (see [Willinger et al.(1997)])). Moreover, Figure 7.8 shows that the 99-th and 98-th latency percentiles converge at different rates which is to be explained with the fact that the simulation has not yet reached stability. Nevertheless, we argue that it is possible to estimate latency quantiles from these simulations by additionally estimating confidence intervals for the rate of convergence. Such an estimation can be obtained by grouping simulations and evaluating the variance of latency quantiles at sample size n for each group. In our case this implies to perform e.g. 20 times 40 simulation runs instead of 40 to estimate upper However, for practical applications some approximated upper bound for the latency quantile which can be estimated from the intercept of normal plots and Figure 7.8 may be sufficient. We roughly estimate this bound to compute the magnitude of the minimal sample size required for such an estimation. The normal plot at 500k downloads leads to an estimation of the expected 99-th latency quantile of $\mu = 7.37$ seconds. This estimation equals up to two digits to estimations from any other normal plot between 210k and 500k downloads. The radius of the confidence interval in which we expect the latency quantile can now be estimated with the long-range dependent equivalent of Equation 7.2

$$confidence\, interval\, radius = 1.96\, \sigma\, n^{H-1} \qquad (7.4)$$

However, both, H and σ, have to be estimated from the data of the simulation. Some rough estimate can be obtained by visually 7.8 fitting a line to the log-log plot of variances [1] (see Figure 7.9). This leads to $H = 0.55$ and $\sigma^2 = 30,000$ seconds^2 excluding the first 100,000 downloads which fall in the initial phase of convergence. Based on these values the order of magnitude of the minimal sample size for estimation a quantile can be estimated with

$$\left(\frac{1.96 * \sigma}{\mu * accuracy}\right)^{\frac{1}{1-H}} \leq n \qquad (7.5)$$

Values for the 99-th latency percentile in this simulation are listed in Table 7.5.

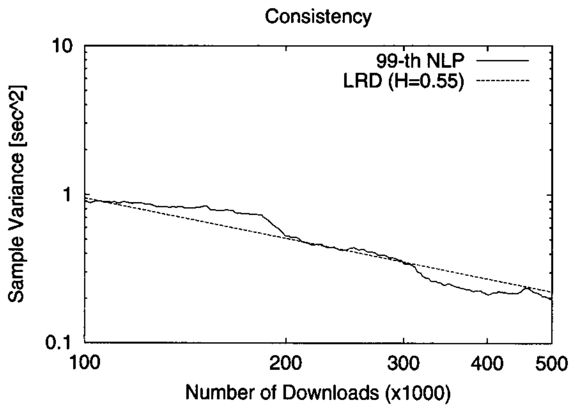


Figure 7.9: *Estimation of H and Sigma (Coarse Model, High Util.)*

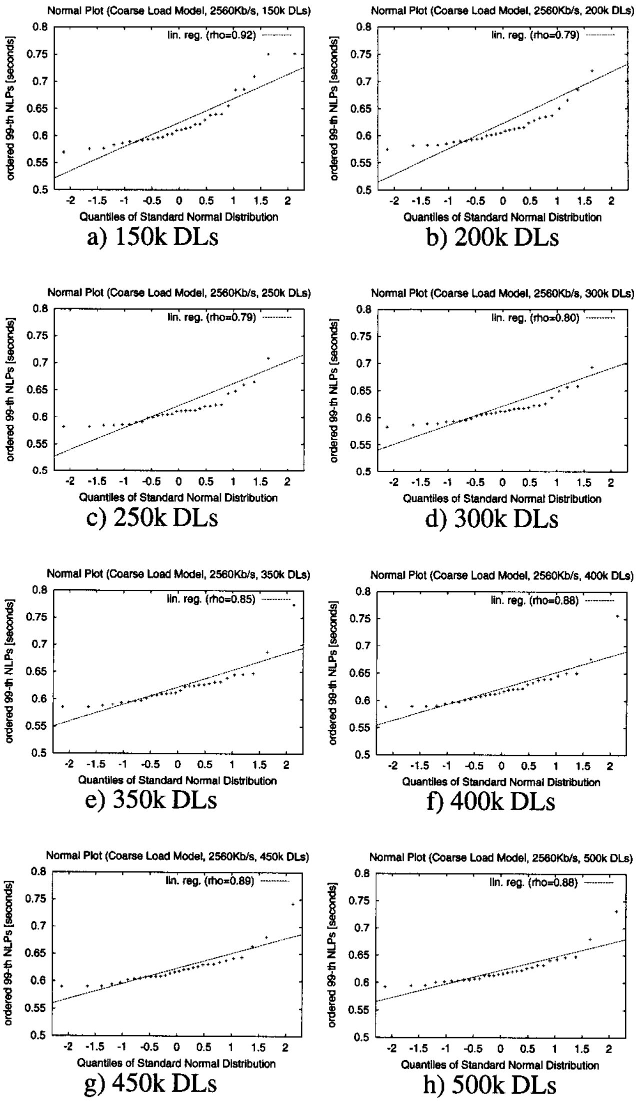
Accuracy of the 99-th percentile	Sample Size
1%	$1 \cdot 10^8$
2%	$3 \cdot 10^7$
3%	$1 \cdot 10^7$
5%	$4 \cdot 10^6$
10%	$8 \cdot 10^5$

Table 7.5: *Approx. Sample Size Required to Estimate the 99-th NLP (Coarse Model, High Util.)*

7.2.3 Medium Utilization

At medium utilization, normal plots do not indicate convergence for any of the latency quantiles investigated within samples sizes we have analyzed. For a sequence of normal plots for the 99-th latency percentile up to 500,000 downloads refer to Figure 7.10.

[1] For more sophisticated estimation procedures that lead to more accurate estimation of H see [Beran(1994)]



a) 150k DLs

b) 200k DLs

c) 250k DLs

d) 300k DLs

e) 350k DLs

f) 400k DLs

g) 450k DLs

h) 500k DLs

Figure 7.10: *Normal Plots for the 99-th NLP (Coarse Workload Model, 2560Kb/s)*

For plots for the 98-th and 95-th latency percentile refer to Appendix D Figure D.11.

There is two possible explanations for this non-convergence within sample sizes that we have simulated: First, the latency distribution has irregularities, i.e. discontinuities or discontinuities of the first derivation around the latency quantiles of interest. As a consequence quantiles cannot converge. Monitoring the histogram of latencies around the 99-th latency quantile (Appendix D Figure D.10) we do not find any indication for such irregularities. Second, the quantiles do converge but have a initial phase larger than the sample size we have done in our simulations. Presumably this is the case since for a latency quantile to converge the convolution of object sizes in simulation input and effects from TCP's congestion control and retransmission, which are rare but statistically relevant at medium utilization, have to converge.

7.3 Summary

In this chapter, we have elaborated on the validation study of the proposed method. We have applied the method described in Section 4.2 to evaluate network latency quantiles such as the 99-th, 98-th, and 95-th percentile of web downloads in a simulation of a web service. We have observed that these latency quantiles can converge within sample sizes that are feasible for simulations. We have shown that under low utilization, when effects from system adaptivity to congestion are negligible, latency quantiles converge at approximately at same sample sizes as corresponding quantiles from the heavy-tailed distribution in simulation input. We have also shown that under high utilization, when effects from system adaptivity are significant, it is still possible to detect convergence of latency quantiles which is at a rate slower than $n^{-1/2}$. However, the initial phase in this convergence, which depends on details of the system adaptivity, can become considerably long.

Chapter 8

Applications

In the previous chapters, we have introduced a method for quantile-based performance evaluation of systems with synthetically generated workload. We have assumed that the workload is generated by sampling long- or heavy-tailed distributions. We have shown results that indicate that the method can be employed to evaluate the network performance of web services for which the size distribution of downloaded objects is long-tailed. Our results indicate that this enables performance evaluation at sample sizes which are practically feasible.

In this chapter, we show the versatility of the method. We give evidence that the method can be applied to almost any problem that requires performance evaluation with long- or heavy-tailed input to which the system model of Figure 1.1 can be applied. The problem does not need to have any relation to the performance evaluation of networks in web services. The only stringent requirements for application of the method is (i) that quantiles of system output have a useful interpretation in the corresponding application scenario and (ii) that the system is such that it does not prevent quantiles of output to converge at sample sizes that are of practically feasible.

We therefore start with reviewing characteristics where measurements indicate long- or heavy-tailed distributions. We then list performance evaluation problems of systems in which these characteristics are part of the input. For each of these performance evaluation problems we review input and output of these systems to show that the problem can be abstracted with the system model given in Figure 1.1 and give arguments that quantiles in the system output are useful in related application scenarios.

In the field of computer and communication systems long- or heavy-tailed distributions appear to fit measurements for the following characteristics.

1. sizes of objects of IMAP or POP3 e-mail services [Charzinski(2002)]
2. sizes of objects of FTP transfers [Paxson and Floyd(1994)]
3. sizes of objects stored in Unix file systems [Irlam(1994)]
4. CPU requirements for Unix processes
   [Leland and Ott(1986)]
   [Harchol-Balter and Downey(1997)]
5. Direct access storage device I/O time requirements [Peterson and Adams(1996)]

The tail indices estimated from these measurements are all between 1.0 and 1.3. Such indices close to 1 are clearly in favor of our method since this implies impractically large sample sizes to evaluate the performance with frequently-used performance statistics that depend on moments of the heavy-tailed distribution.

In fields other than computer and communication systems long or heavy-tailed distributions appear to fit measurements for the following characteristics.

1. the cost for hurricane, fire, and earthquake damage [Doyle(1999)]
2. the distribution of income in the US

Hence, performance evaluation problems of systems in which these characteristics are part of the input include the following:

8.1 Network Performance Evaluation

Network performance evaluation for services such as e-mail or file transfer where file sizes are long- or heavy-tailed with tail indices close to 1 are problems where the proposed method can be applied. Similar to the web services, e-mail or file transfer services all employ request/reply transactions. Part of the system input is the size of the requested file. System output is the latency of requests. Therefore, performance evaluation of these services can be abstracted with the system model given in Figure 1.1. Similar to the web services, latency quantiles such as the 99-th percentile are useful to statistically characterize system performance.

Hence, the proposed method can be applied to evaluate problems such as

- Evaluation of network capacity provisioning
- Evaluation of advancements in protocol development
- Evaluation of optimizations of protocols e.g. for wireless scenarios

8.2 Server Performance Evaluation

Performance Evaluation for web, file, and e-mail servers where file sizes are long- or heavy-tailed with tail indices close to 1 are problems where the proposed method can be applied. Web, e-mail or file transfer services all employ request/reply transactions. Part of the system input is the size of the requested file. System output is the latency of requests. Therefore, performance evaluation of these services can be abstracted with the system model given in Figure 1.1. Latency quantiles such as the 99-th percentile are clearly useful to statistically characterize server performance.

Hence, the proposed method can be applied to evaluate problems in the context of server performance evaluation. These problems include:

- Evaluation of server configuration and server capacity provisioning
- Evaluation of CPU scheduling algorithms on a single server
- Evaluation of load balancing algorithms which schedule requests between multiple servers

One problem that deserves special attention in server configuration is to optimize parameters that control the maximum number of requests which can be processed in parallel. This parameter is is known to have great impact on server performance [Liu et al.(2003)]. So far performance evaluations to optimize such parameters for web servers have been performed with workloads generated from distributions that inherently limit variability (see Section 2.4.1 and [Liu et al.(2003)]). The proposed method can remedy this drawback and lead to more realistic results.

8.3 Computer System Performance Evaluation

The proposed method can also be applied in the evaluation of computer systems since CPU and I/O time requirements of are long- or heavy-tailed.

The input in these evaluations are the CPU requirements and the arrival rates of the processes. The CPU requirement is typically long- or heavy-tailed, the arrival rate is exponential. The output is the wall time or response time, the slowdown, i.e. the ratio of CPU requirement and wall time as well as the queue length. Therefore, performance evaluation of these services can thus be abstracted with the system model given in Figure 1.1. Since quantiles of response time, slowdown, or queue-length have a natural interpretation, the proposed method can also be applied for the evaluation of computer systems.

Hence, the proposed method can be applied in the evaluation of computer systems. Problems include:

- Evaluation of migration policies in a network of workstations
- Evaluation of task assignment policies for a distributed server

8.4 Outside Computer and Communication Systems

We restrict to denoting that the proposed method may be of use in fields outside of computer and communication systems namely in economics. We do not elaborate on any of the examples since the author of this thesis is not familiar with these fields.

8.5 Summary

We have shown that the quantile-based method which this thesis proposes has more applications than the evaluation of the network performance of web services. Further applications in the evaluation of network performance include evaluations of network capacity provisioning, evaluations of advancements and optimizations in protocol development for services such as e-mail or file transfer. Applications in the evaluation of server performance include evaluations of capacity provisioning, evaluations of server configuration, and evaluations of algorithms for request scheduling and load balancing for web, file, and e-mail services. More applications of the method are in the field of computer systems. These include the evaluation of migration policies in a network of workstations as well as the evaluation of task assignment policies for distributed servers.

Chapter 9

Conclusions and Further Work

In this thesis, we have developed a new method for performance evaluation of systems with synthetic workloads that are generated by sampling from a heavy-tailed distribution. This method enables the evaluation of the performance of systems under heavy-tailed input at sample sizes that are practically feasible. The method exploits the fact that quantiles of system output, such as the 99-th percentile, converge long before frequently-used statistics such as the average. We have shown that for many applications of the method, such quantiles are a useful statistic to characterize system performance. We have further shown results that indicate that this method can be employed to evaluate network performance in simulations of web services. Moreover, we have shown the generality of the method by giving applications of the method other than capacity provisioning for networks of web services.

9.1 Review of Contributions

In Section 1.4 we have stated the research contribution of this thesis. Now it is time to revisit these contributions. Below, each of these contributions is assessed.

1. **Quantiles are suitable statistics for performance evaluation**
   We give evidence that quantiles are suitable statistics for performance evaluation of systems with synthetic workloads that were generated by sampling heavy-tailed distributions.

   To show the suitability of quantiles for performance evaluation we have shown (i) that quantiles have useful interpretation in many application scenarios that are related to statistical quality of service guarantees for the system and (ii) that quantiles of output can converge at sample sizes that are feasible for performance evaluations.

   We have reviewed the interpretation of quantiles in a number of application scenarios. For the example for the application of a web service we have argued that quantiles, such as the 99-th download latency quantile, naturally reflect user-perceived system performance since 99% of the downloads have a latency smaller than the 99-th percentile.

   To show that quantiles can converge at sample sizes that are feasible for performance evaluations we have reviewed probability theory. We have given theory from which we can infer that quantiles in system in- and output converge to a normal distribution if the underlying distributions are sufficiently regular. We have further applied this theory to show that quantiles in system input converge at sample sizes that are practically feasible for performance evaluations. For the example of the application of performance evaluation of a web service we have calculated that the 99-th percentile of a distribution of web objects converges to a 5% relative accuracy at a sample size of $1.2 \cdot 10^5$. Moreover, we have given indications from theory that quantiles in system output can generally converge at sample sizes that are practically feasible for performance evaluation. For the example of the application of performance evaluation of a web service we have presented results which indicate that quantiles such as the 99-th download latency percentile can converge at sample sizes below $5 \cdot 10^5$ which is practically feasible to achieve.

2. **A priori bounds for evaluation duration**
   We provide lower bounds that estimate the initial phase in the convergence of quantiles in system output.

   We have shown how to apply statistics to evaluate this convergence of quantiles from a heavy-tailed distribution in system input. It seems reasonable to assume that this convergence estimates the initial phase of the convergence of the corresponding quantile in system output. For example for the application of performance evaluation of a web service this can further be supported with results from the validation study.

3. **Estimation of quantiles in system output**
   We give a method to test whether quantiles of simulation output have converged. In case of convergence the method additionally provides accurate estimates for the quantiles.

   Our review of probability theory shows that quantiles in system output generally converge to a normal distribution when the underlying distribution is sufficiently regular, which is reasonable to assume in most application scenarios. We have therefore proposed to employ standard procedures to assess this convergence to a normal distribution. The method that we have proposed combines normal probability plots to visually pretest convergence, a hypothesis test to make the pretest reliable, and monitors the rate of convergence. Moreover, in case of convergence, the standard procedures in our method can also be employed to estimate the quantile.

4. **Practicability of the method**
   We show that the test method can be employed to evaluate the network performance of web services in terms of latency quantiles.

   We have applied our test method to evaluate the performance of web service in a client server scenario. The results indicate that the method can be employed to estimate network latency quantiles at sample sizes that are of practical use. Hence the method can be employed for such performance evaluations. We have additionally verified that the average network latency did not converge in our performance evaluations and thus cannot be evaluated.

5. **Versatility of the method**
   We show that our method is not limited to the performance evaluation of network performance of web services. Instead, the method has a variety of further applications which need not be related to the evaluation of network performance of web services.

   We have shown that our method can be employed to evaluate a number of problems in the field of computer- and communication systems. These include network capacity provisioning and protocol evaluation for e-mail and file transfer services. They also include server capacity provisioning for web, e-mail, and file servers as well as the evaluation of server configuration and algorithms for request scheduling and load balancing. Further applications of the method include the evaluation of computer systems. Examples are the evaluation of migration policies in a network of workstations as well as task assignment policies for distributed servers.

9.2 Further Work

In this section, we present some open issues and unsolved problems related to the proposed method.

Impact of Topology For the application of performance evaluation of a web service, our evaluation has assumed a dumbbell network topology which represents some bottleneck link in a network. This may hold as a base line for evaluations. However, it remains to be seen how variations in topology, speed of access and backbone links affect the convergence of network latency percentiles. It also remains to be seen how variations of utilization on these different topologies affect convergence.

Impact of Document Popularity For the application of performance evaluation of a web service we have not modeled document popularity when generating workload. Modeling this document popularity means to introduce correlations in workload generation which in turn affects convergence properties. However, it remains to be seen how this affects the convergence of network latency percentiles in the evaluation.

Testing Applications A number of applications of the proposed method have been defined and analyzed in theory. Now it remains to be seen whether quantiles that are of interest to the application scenarios converge at sample size that are feasible for performance evaluations.

Normality Tests We have tested convergence to normality by employing linear regression to quantify the linearity of the normal plot. However, more than two dozen normality test procedures could be used instead of quantifying the linearity in the normal plot. Therefore it remains to be seen how results are affected when different normality tests are employed to assess the convergence of quantiles in system output.

Acknowledgements

I would like to thank

- Bernhard Plattner, Polly Huang, and Martin Maechler for introducing me into the world of academic research and their invaluable support and patience.
- my colleagues Ralph Keller, Vincent Lenders, Matthias Gries, Oliver Lamparter, Samarjit Chakraborty for valuable discussions and suggestions to improve my work as well as for proof-reading parts of this thesis.
- my parents, my son Joel, and his mother Franziska Manz for their moral support while working on this thesis.

Biography

Ulrich Fiedler was born in Singen/Hohentwiel, Germany on January, 12th, 1965 where he also attended high school (Abitur [mathematical/scientific degree] in May 1984). After the compulsory military service, he began a study of mathematics and physics at the Eidgenössische Technische Hochschule in Zürich, Switzerland (ETHZ) in November 1985. In November 1990, he graduated with a diploma degree (Dipl. Physiker ETH). He then worked for Siemens Electrocom GmbH, Konstanz, Germany as a software-engineer and project-leader in the field of interpretation of optical character recognition results for postal automation. In 1995, he attended the post-graduate course in computer engineering at ETHZ focusing on databases and computer networks while continuing to work at Siemens Electrocom GmbH.

In Janary 1998 he joined the computer engineering and networks laboratory at ETHZ. He worked on projects in the fields of quality of service for Internet-based workflows, diffserv for Intra-nets which serve voice and data, and methods to assess system performance for data traffic with heavy-tailed characteristics. The latter project resulted in this thesis.

Currently, he works on a technology transfer project that is based on parts of this thesis.

Bibliography

[Arlitt and Williamson(1996)] M. F. Arlitt and C. L. Williamson. Web Server Workload Characterization: The Search for Invariants. In *Proceedings of SIGMETRICS'96*, pages 126 – 137, Philadelphia, Pennsylvania, USA, May 1996. ACM.

[Bajaj et al.(1998)] S. Bajaj, L. Breslau, and S. Shenker. Is Service Priority Useful in Networks? In *Proceedings of the ACM Sigmetrics '98*, Madison, Wisconsin USA, June 1998.

[Barford and Crovella(1998)] P. Barford and M. Crovella. Generating Representative Web Workloads for Network and Server Performance Evaluation. In *Proceedings of Performance '98/ACM SIGMETRICS '98*, pages 151–160, Madison, Wisconsin, USA, June 1998.

[Ben Fredj et al.(2001)] S. Ben Fredj, T. Bonald, A. Proutiere, G. Regnie, J. Roberts (France Telecom) Statistical Bandwidth Sharing: A Study of Congestion at Flow Level. In *Proceedings of SIGCOMM'01*, pages 111–122, San Diego, California, USA, Aug. 2001. ACM.

[Beran(1994)] J. Beran. *Statistics for Long-Memory Processes.* Chapman Hall/CRC, Boca Raton, FL, USA, 1st edition, 1994.

[Berners-Lee et al.(1996)] T. Berners-Lee, R. Fielding, and H. Frystyk. Hypertext Transfer Protocol — HTTP/1.0. RFC 1945, Internet Request For Comments, May 1996.

[Charzinski(2002)] J. Charzinski. Internet Traffic - Characteristics, Performance and Models. Nov. 2002.

[Christiansen et al.(2000)] M. Christiansen, K. Jaffay, D. Ott, and F. D. Smith. Tuning RED for Web Traffic. In *Proceedings of SIGCOMM'2000*, pages 139–150, 2000.

[Crovella and Bestavros(1996)] M. Crovella and A. Bestavros. Self-similarity in World Wide Web Traffic: Evidence and Possible Causes. In *Proceedings of SIGMETRICS'96*, pages 160 – 169, Philadelphia, Pennsylvania, USA, May 1996. ACM.

[Crovella et al.(1999)] M. Crovella, R. Frangioso, and M. Harchol-Balter. Connection Scheduling in Web Servers. In *USENIX Symposium on Internet Technologies and Systems (USITS '99),*, Oct. 1999.

[Crovella and Lipsky(2000)] M. Crovella and L. Lipsky. Simulations with Heavy-tailed Workloads. In K.Park and W. Willinger, editors, *Self-Similar Network Traffic and Performance Evaluation*, chapter 3, pages 89–100. Wiley-Interscience, NY, 2000.

[Doyle(1999)] J. Doyle. Highly optimized Tolerance: A Mechanism for Power Laws in designed Systems. From Talk Slides presented at MIT. Mar. 1999.

[Erramilli et al.(1996)] A. Erramilli, O. Narayan, and W. Willinger. Experimental Queueing Analysis with Long-Range Dependent Packet Traffic. *IEEE/ACM Transactions on Networking*, 4(2):209–223, 1996.

[Feldmann et al.(1999)] A. Feldmann, A. Gilbert, P. Huang, and W. Willinger. Dynamics of IP Traffic: A Study of the Role of Variability and the Impact of Control. In *Proceedings of SIGCOMM'99*, Cambridge, Massachusetts, USA, Sept. 1999. ACM.

[Fiedler(2001)] U. Fiedler, P. Huang, B. Plattner. Towards Provisioning Diffserv Intra-Nets. In *Proceedings of IWQoS'01*, pages 27–43, Karlsruhe, Germany, June 2001. Springer.

[Fielding et al.(1999)] R. Fielding, J. Gettys, J. C. Mogul, H. Frystyk, and T. Berners-Lee. Hypertext Transfer Protocol — HTTP/1.1. RFC 2616, Internet Request For Comments, June 1999.

[Filliben(1975)] J. Filliben. The Probability Plot Correlation Coefficient Test for Normality. *Technometrics*, pages 111–117, 1975.

[Goldie and Kluppelberg(1997)] C. Goldie and C. Kluppelberg. Subexponential Distributions. In R. F. R. Adler and M. Taqqu, editors, *A Practical Guide to Heavy Tails: Statistical Techniques for Analysing Heavy Tails*, pages 435–460. Birkhauser, Basel (CH), 1997.

[Hampel et al.(1986)] F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. Wiley, NY, 1986.

[Harchol-Balter et al.(1999)] M. Harchol-Balter, M. E. Crovella, and C. D. Murta. On choosing a Task Assignment Policy for a distributed Server System. *Journal of Parallel and Distributed Computing*, 59(2):204–228, 1999.

[Harchol-Balter and Downey(1997)] M. Harchol-Balter and A. B. Downey. Exploiting Process Lifetime Distributions for dynamic Load Balancing. *ACM Transactions on Computer Systems*, 15(3):253–285, 1997.

[Huber(1981)] P. Huber. *Robust Statistics.* Wiley, NY, 1981.

[Irlam(1994)] G. Irlam. Unix File Size Survey. http://www.base.com/gordoni/ufs93.html, Sept. 1994.

[Jacobson(1988)] V. Jacobson. Congestion Avoidance and Control. In *ACM SIGCOMM '88*, pages 314–329, Stanford, CA, Aug. 1988.

[Johnson et al.(1994)] N. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univariate Distributions*, volume 1 of *Wiley Series in Probability and Mathematical Statistics.* Wiley, NY, 2nd edition, 1994.

[Krishnamurthy and Rexford(2001)] B. Krishnamurthy and J. Rexford. *Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement.* Addison-Wesley, 1st edition, 2001.

[Leland et al.(1994)] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the Self-Similar Nature of Ethernet Traffic (Extended Version). *IEEE/ACM Transactions on Networking*, 2(1):1–15, Dec. 1994.

[Leland and Ott(1986)] W. E. Leland and T. J. Ott. Load-balancing Heuristics and Process Behavior. In *Proceedings of SIGCOMM'86*, pages 54–69, May 1986.

[Likhanov et al.(1995)] N. Likhanov, B. Tsybakov, and N. Georganas. Analysis of an ATM Buffer with Self-Similar Fractal Input Traffic. In *Proceedings of IEEE INFOCOM '95*, pages 985–992, Boston, MA, USA, Apr. 1995.

[Liu et al.(2003)] X. Liu, L. Sha, Y. Diao, S. Froehlich, J. Hellerstein, and S. Parekh. Online Response Time Optimization of Apache Web Server. In *Proceedings of IWQoS'03*, Berkeley, CA, USA, June 2003. Springer.

[Mah(1997)] B. Mah. An Empirical Model of HTTP Network Traffic. In *Proceedings of the IEEE Infocom*, pages 592–600, Kobe, Japan, Apr. 1997. http://www.ca.sandia.gov/ bmah/Papers/Http-Infocom.ps.

[Mandelbrot and Wallis(1968)] Mandelbrot and Wallis. Noah, Joseph and operational Hydrology. *Water Resources Res.*, 4(5):909–918, 1968.

[Nahum(2002)] E. Nahum. Personal e-mail Exchange. Dec. 2002.

[Nahum et al.(2001)] E. Nahum, M. Rosu, S. Seshan, and J. Almeida. The Effects of Wide-Area Conditions on WWW Server Performance. In *Proceedings of SIGMETRICS'01*, pages 257–267, Cambridge, Massachusetts, USA, June 2001. ACM.

[Nolan(2002)] J. Nolan. Stable Distributions - Models for Heavy-tailed Data. http://academic2.american.edu/ jpnolan/stable/chap1.pdf, Aug. 2002.

[Odlyzko(2000)] A. Odlyzko. The Internet and other Networks: Utilization Rates and Their Implications. *Information Economics and Policy*, 12:341–365, 2000. http://www.research.att.com/ amo/doc/networks.html.

[Park et al.(1996)] K. Park, G. Kim, and M. Crovella. On the Relationship between File Sizes, Transport Protocols, and Self-similar Network Traffic. In *Proceedings of the IEEE International Conference on Network Protocols*, pages 171–180, Oct. 1996. http://cs-pub.bu.edu/faculty/crovella/papers.html.

[Park et al.(1997)] K. Park, G. Kim, and M. Crovella. On the Effect of Traffic Self-similarity on Network Performance. In *Proceedings of the SPIE International Conference on Performance and Control of Net-*

work Systems, pages 296–310, Nov. 1997. http://cs-pub.bu.edu/faculty/crovella/papers.html.

[Park and Willinger(2000)] K. Park and W. Willinger. Self-similar Network Traffic: An Overview. In *Self-Similar Network Traffic and Performance Evaluation*, chapter 1. Wiley-Interscience, NY, 2000.

[Paxson and Floyd(1994)] V. Paxson and S. Floyd. Wide-area traffic: The Failure of Poisson Modeling. In *Proceedings of SIGCOMM'94*, pages 257–268. ACM, Aug. 1994.

[Paxson and Floyd(1997)] V. Paxson and S. Floyd. Why We Don't Know How To Simulate the Internet. In *Winter Simulation Conference*, pages 1037–1044, 1997.

[Peterson and Adams(1996)] D. Peterson and D. Adams. Fractal Patterns in DASD I/O traffic. In *Proceedings of CMG*, Dec. 1996.

[Rao(1973)] C. R. Rao. *Linear Statistical Inference and Its Applications*. Wiley, New York, 2nd edition, 1973.

[Raunak et al.(2000)] M. Raunak, P. Shenoy, P. Goyal, and K. Ramamritham. Implications of Proxy Caching for Provisioning Networks and Servers. In *Measurement and Modeling of Computer Systems*, pages 66–77, 2000.

[Rice(1995)] J. Rice. *Mathematical Statistics and Data Analysis, 2nd edition*. Duxbury Press, 1995.

[Willinger et al.(1995)] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson. Self-similarity through high Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level. In *Proceedings of SIGCOMM'95*, pages 100–113, Cambridge, Massachusetts, USA, Aug. 1995. ACM.

[Willinger et al.(1997)] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson. Self-similarity through high Variability:

Statistical Analysis of Ethernet LAN Traffic at the Source Level. *IEEE/ACM Transactions on Networking*, 5(1):71–86, 1997.

[Mathematica(1999)] S. Wolfram. *The Mathematica Book.* "Cambridge University Press", 4th edition, 1999.

[ns-2(2000)] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu. Advances in Network Simulations. *IEEE Computer*, May 2000.

[nam(2003)] URL. Nam Network Animator. http://www.isi.edu/nsnam/nam/index.html, 2003.

[ns-2 Change Log (2003)] URL. Ns Change Log. http://www.isi.edu/nsnam/ns/CHANGES.html, 2003.

[ns-2 Research)] URL. Research using ns. http://www.isi.edu/nsnam/ns/ns-research.html, 2003.

[statguide(1997)] URL. PROPHET StatGuide: Examples of Normal Probability Plots. http://www.basic.nwu.edu/statguidefiles/probplots.html, 1997.

Appendix A

E-mail Exchange with Jan Beran

In Section 3.3.1 we have claimed that the author of [Beran(1994)] has argued in an e-mail exchange that results in Section 8.3 of the book also apply for quantiles. The e-mail exchange (in German) is documented here.

```
From: Jan.Beran@uni-konstanz.de
To: Martin Maechler <maechler@stat.math.ethz.ch>
Subject: Re: Asym.Verteilung empirischer
             Quantile unter "Long Range Dependence"
Date: Wed, 05 Mar 2003 12:10:48 +0100 (MET)

Die Aussage des Satzes ist auch fuer den Median gueltig.
Ein Beweis ergibt sich eleganter aus bekannten
funktionalen Grenzwertsaetzen fuer die empirische
Verteilung von "long-memory processes", wie sie v.a.
von Giraitis bewiesen wurden. Fractional G.n. ist dabei
nur ein Spezialfall (es ist eben nicht
einfacher den Satz fuer diesen Spezialfall zu beweisen).

 <....>
```

```
Zitiere Martin Maechler <maechler@stat.math.ethz.ch>:

> >>>>> "Jan" == Jan Beran <Jan.Beran@uni-konstanz.de>
> >>>>>     on Wed, 05 Mar 2003 09:35:32 +0100 (MET) writes:

  <....>

>     Jan> Kurze Antwort auf Deine Frage:
>     Jan> Soviel ich sehe gibt es keinen Druckfehler.
>     Jan> Das c_\gamma kommt aus Theorem 2.2.
> Aha! ok, leuchtet jetzt ein ...
>
>     Jan> Die psi-Funktion ist streng genommen nicht
>     Jan> genuegend regulaer, um die Bedingungen in
>     Jan> Beran (1991) zu erfuellen (auch der Median
>     Jan> erfuellt die Bedingungen eigentlich nicht).
>
> "eigentlich" : Heisst dies, dass die Aussage des
> Theorems trotzdem gilt (jetzt mal fuer den Median),
> auch wenn der Beweis angepasst werden muesste?
>
>     Jan> Man kann aber natuerlich diese
>     Jan> psi-Funktion beliebig genau durch eine
>     Jan> geneugend regulaere Funktion approximieren.
>     Jan> (Die genauen mathematischen Details
>     Jan>  aufzuschreiben erfordert
>     Jan>  natuerlich ein bisschen Arbeit...)
>
> klar.
> Fuer fractional Gaussian noise (und ARIMA?) ist die
> Asymptotik des Medians sicher doch bekannt? --
> und jene fuer alpha-Quantile (festes alpha)
> waere dann analog?

  <....>

>
>     Jan> Zitiere Martin Maechler <maechler@stat.math.ethz.ch>:

  <....>

>     >> Man ja Quantile (mit festem alpha) als
>     >> M-Schaetzer definieren, allerdings mit etwas
>       >> unueblicher psi-Funktion.
>       >> Dann wuerde ja alles aus Beran(1991) folgen --
>       >> wobei ich nicht dieses sondern Dein
>       >> Chapman&Hall Buch vor mir habe.
>       >> Ich beziehe mich auf S.151 ff, speziell
>       >> Theorem 8.2, auf S.153.
>       >> Dort hat die Formel fuer
>       >> \sigma_{\mu} noch ein c_{\gamma} drin,
>       >> was aber, glaub ich, nicht
>       >> (dort wenigstens) definiert ist --
>       >> stattdessen definierst Du c_1 und c_2 ---
>       >> Tippfehler?
>       >>
>       >> Neben der c_\gamma Frage, bleibt jene, ob die
>       >> psi-Funktion des alpha-Quantils
>       >> genuegend regulaer ist (nicht differenzierbar
>       >> bei 0), um die Voraussetzungen von
>       >> Theorem 8.2 zu erfuellen.
```

Appendix B

Examining Normal Plots

Normal plots can be employed to visually assess the fit of data to a normal distribution. If the data in the set follows a normal distribution, the result of the plot is close to a straight line (see Section 4.2.1 for details). Deviations from this straight usually may reveal deviations from normality. This appendix lists the most frequently encountered deviations.

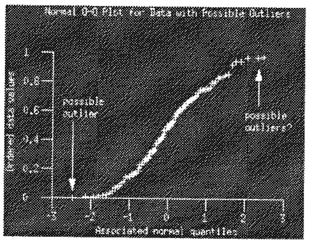
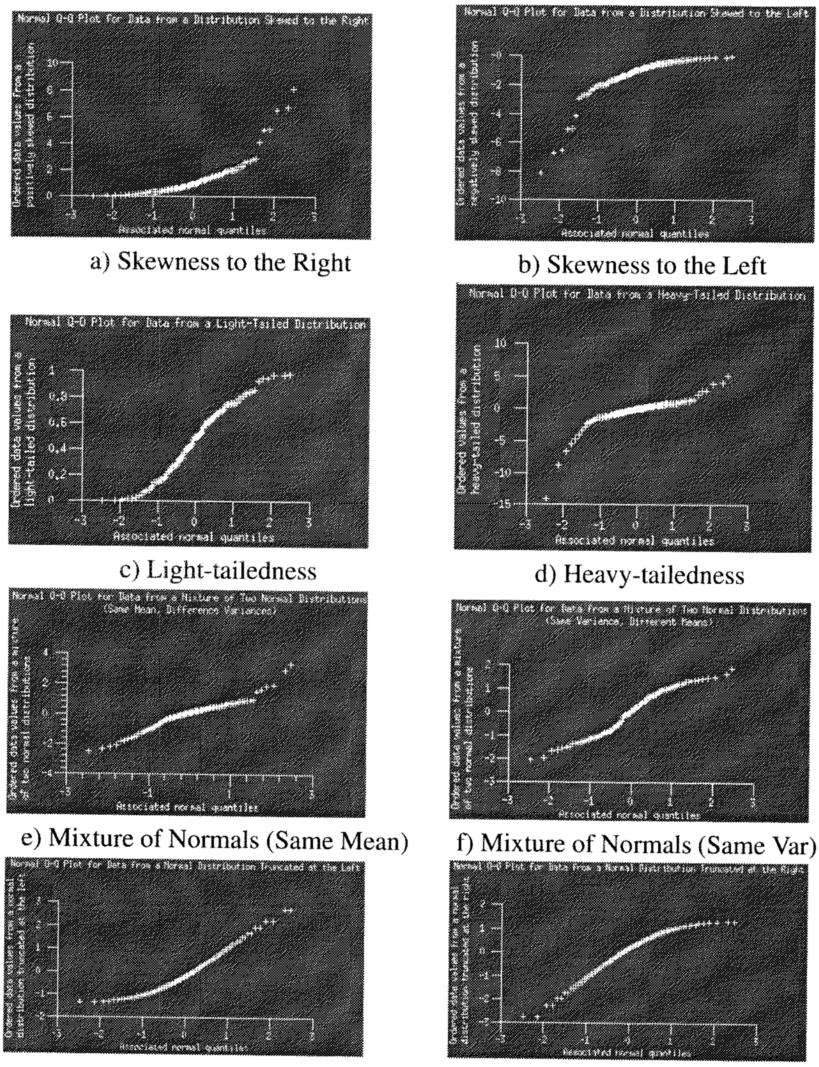


Figure B.1: *Example: Normal Plot for Data with Outliers [statguide(1997)]*



a) Skewness to the Right

b) Skewness to the Left

c) Light-tailedness

d) Heavy-tailedness

e) Mixture of Normals (Same Mean)

f) Mixture of Normals (Same Var)

g) Normal Truncated at Left

h) Normal Truncated at Right

Figure B.2: *Normal Plots: Possible Deviations from Normality [statguide(1997)]*

- *Suspected outlier(s)*
  For data sampled from a normal distribution, the X-Y values in the normality plot will lie along a hypothetical straight line passing through the main body of the X-Y values. If this is generally true, with a few points lying off that hypothetical line, those points are likely outliers, as with the smallest data value and, perhaps, the largest two data values in the hypothetical example shown in Figure B.1.

- *Skewness to the right*
  If both ends of the normality plot bend above a hypothetical straight line passing through the main body of the X-Y values of the probability plot, then the population distribution from which the data were sampled may be skewed to the right. Figure B.2 (a) shows a hypothetical example of a normal probability plot for data sampled from a distribution that is skewed to the right.

- *Skewness to the left*
  If both ends of the normality plot bend below a hypothetical straight line passing through the main body of the X-Y values of the probability plot, then the population distribution from which the data were sampled may be skewed to the left. Figure B.2 (b) shows a hypothetical example of a normal probability plot for data sampled from a distribution that is skewed to the left.

- *Light-tailedness*
  If the right (upper) end of the normality plot bends below a hypothetical straight line passing through the main body of the X-Y values of the probability plot, while the left (lower) end bends above that line (an S curve), then the population distribution from which the data were sampled may be light-tailed. Figure B.2 (c) shows a hypothetical example of a normal probability plot for data sampled from a distribution that is light-tailed.

- *Heavy-tailedness*
  If the right (upper) end of the normality plot bends above a hypothetical straight line passing through the main body of the X-Y values of the probability plot, while the left (lower) end bends below it, then the population distribution from which the data were sampled may be heavy-tailed. Figure B.2 (d) shows a hypothetical example of a normal probability plot for data sampled from a distribution that is heavy-tailed.

- *Mixtures of normal distributions*
  Data may be sampled from a mixture of normal distributions. Depending on the means and variances of the component normal distributions, and on the relative proportions of the data that come from each distribution, a mixture of normal distributions may produce a variety of normal probability plots. Figure B.2 (e) shows a hypothetical example of a normal probability plot for data sampled from a mixture of two normal distributions with the same average but different variances. Such a mixture of normal distributions may be hard to distinguish from a symmetric, heavy-tailed distribution. Figure B.2 (f) shows an example of a normal probability plot for data sampled from a mixture of two normals with the same variance but different averages. Such a mixture of normal distributions may be hard to distinguish from a light-tailed distribution.

- *Truncated normal distributions*
  The normal probability plot for data sampled from a truncated normal distribution will resemble one for data from a skewed distribution. Figure B.2 (g) shows a hypothetical example of a normal probability for data sampled from a normal distribution truncated at the left. This may be hard to distinguish from a normal probability plot for a distribution skewed to the right. Figure B.2 (h) shows a hypothet-

ical example of a normal probability plot for data sampled from a normal distribution truncated at the right. This may be hard to distinguish from a normal probability plot for a distribution skewed to the left.

Appendix C

Normality Tests

We briefly review the most frequently-used normality tests.

- *Kolmogorov-Smirnov test*
  The Kolmogorov-Smirnov is based on the evaluation of the greatest discrepancy between the observed and expected cumulative distribution $|S_n(x) - F(x)|$. This test can be applied to test whether data follow any specified distribution, not just the normal distribution. The Kolmogorov-Smirnov test becomes a conservative test and thus loses power if the parameters σ and μ of the normal distribution are not specified beforehand, but must be estimated from the sample data. The Kolmogorov-Smirnov test will not indicate the type of nonnormality, say whether the distribution appears to be skewed or heavy-tailed. Examination of the normal plot for the data is necessary to provide clues as to why the data failed the Kolmogorov-Smirnov test.

- *Shapiro-Wilk and D'Agostino-Pearson tests*
  The Shapiro-Wilk test calculates a W statistic that tests whether a random sample, $x_1, x_2, \ldots, x_n$ comes from

(specifically) a normal distribution. Small values of W are evidence of departure from normality. The W statistic is calculated as follows:

$$W = \frac{(\sum_{i=1}^{n} a_i x_{(i)})^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{C.1}$$

where the $x_{(i)}$ are the ordered sample values ($x_{(1)}$ is the smallest) and the a_i are constants generated from the averages, variances and covariances of the order statistics of a sample of size n from a normal distribution. This test and the D'Agostino-Pearson test are specifically designed to detect departures from normality, without requiring that the parameters σ and μ of the hypothesized normal distribution be specified in advance. These tests tend to be more powerful than the Kolmogorov-Smirnov test, but, as general tests, they will not indicate the type of non-normality, say whether the distribution appears to be skewed as opposed to heavy-tailed (or both). Examination of the normal plot for the data is necessary to provide clues as to why the data failed the Shapiro-Wilk or D'Agostino-Pearson test.

- *D'Agostino's test for skewness*
  D'Agostino's test for skewness tests for non-normality due to a lack of symmetry. Data sampled from a symmetric distribution may not fail the skewness test, even if the distribution is substantially light-tailed (such as a uniform distribution) or heavy-tailed (such as a Cauchy distribution, or a mixture of normal distributions with the same average but different variances). Thus, failure to reject the null hypothesis does not necessarily mean that the data come from a normal distribution. If data fail the skewness test, the conclusion is that the underlying distribution is significantly skewed, but that does not preclude the possibility that it is also substantially heavy-tailed or light-tailed with respect to the normal distribution (as might be the case with data from

a mixture of normal distributions with the same average but different variances). Examination of the normal plot may help in detecting whether the underlying distribution might also have non-normal tails.

- *Anscombe-Glynn test for kurtosis*
  The Anscombe-Glynn test for kurtosis tests for non-normality due to tail heaviness relative to the normal distribution. Data sampled from a distribution with tail heaviness comparable to that for the normal distribution may not fail the kurtosis test, even if the distribution is substantially skewed (such as a truncated normal distribution, or a mixture of normal distributions with the different averages but the same variance). Thus, failure to reject the null hypothesis does not necessarily mean that the data come from a normal distribution. If data fail the kurtosis test, the conclusion is that the underlying distribution has non-normal kurtosis, but that does not preclude the possibility that is also substantially skewed with respect to the normal distribution. Examination of the normal probability plot may help in deciding whether the underlying distribution might also be skewed.

A alternative method to applying one or more of these tests is based on employing linear regression to evaluate the deviation from linearity in a normal plot.

Appendix D

Details of Simulation Evaluation

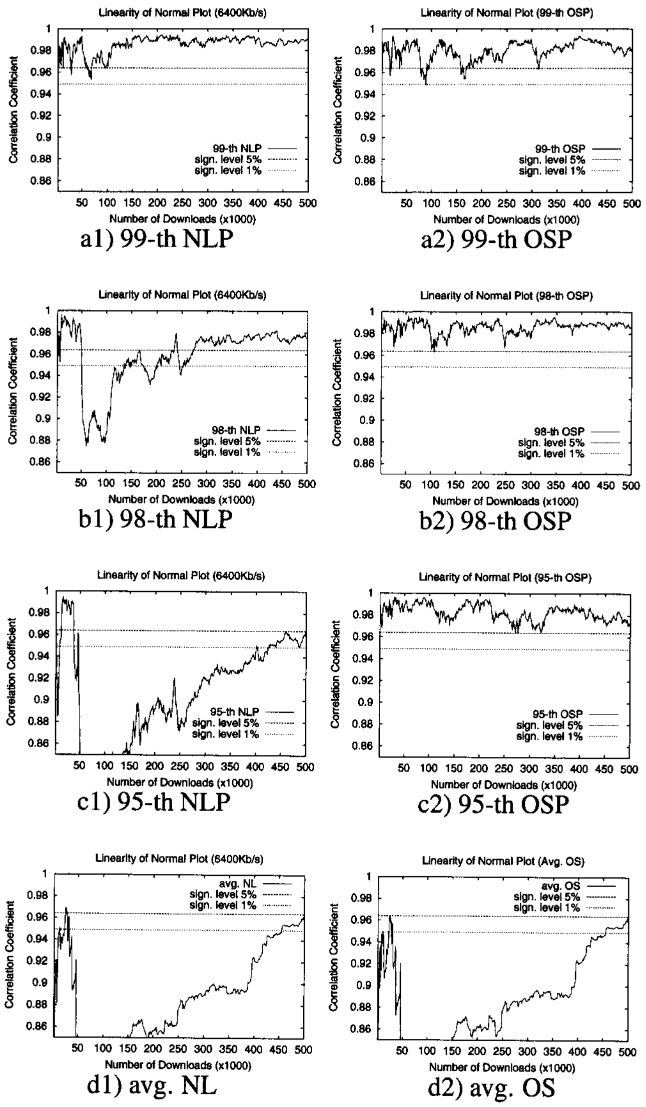
This appendix lists all details of the results discussed in Section 7.2.

Percentile	Latency
99-th	0.275 ± 0.004 sec
98-th	0.205 ± 0.002 sec
95-th	0.155 ± 0.001 sec

Table D.1: *Estimated NLPs (Coarse Model, Low Utilization, 500k DLs)*

Percentile	Latency
99-th	7.35 ± 0.92 sec
98-th	5.08 ± 1.37 sec
95-th	2.42 ± 0.41 sec

Table D.2: *Estimated NLPs (Coarse Model, High Utilization, 500k DLs)*



a1) 99-th NLP

a2) 99-th OSP

b1) 98-th NLP

b2) 98-th OSP

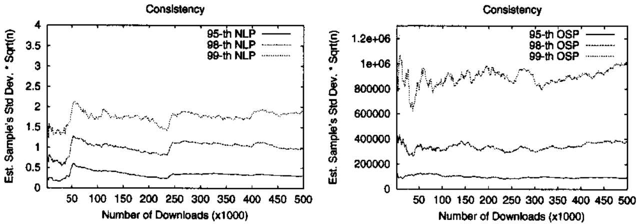
c1) 95-th NLP

c2) 95-th OSP

d1) avg. NL

d2) avg. OS

Figure D.1: *Convergence of NLPs, OSPs and avg. NL, avg OS (Coarse Model, Low Util.)*



a1) NLPs

a2) OSPs

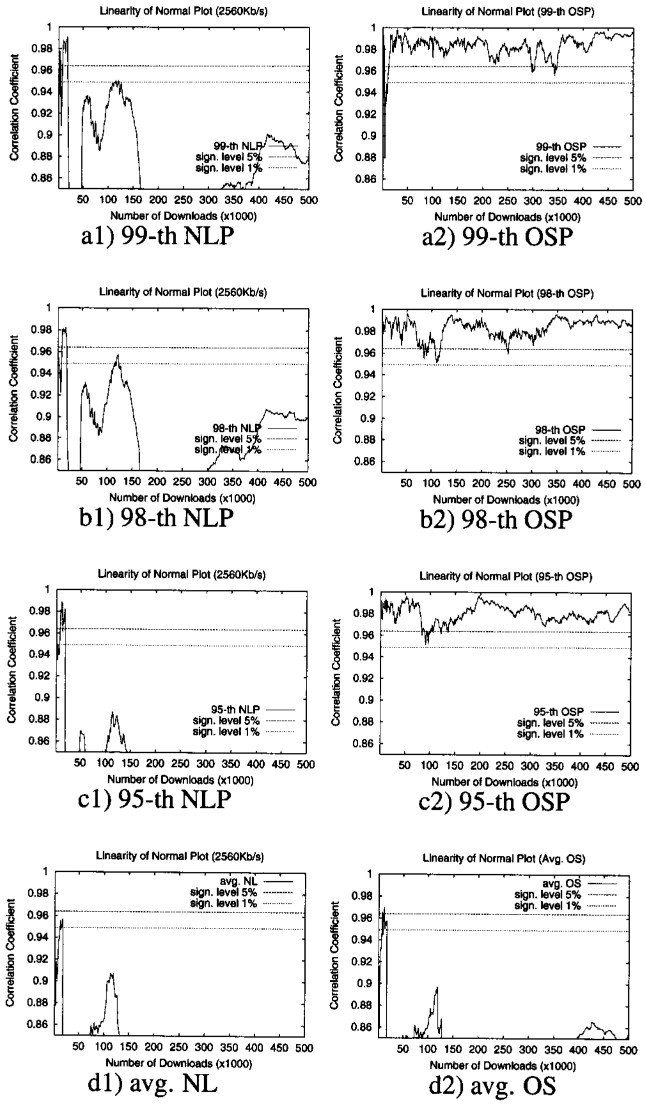
Figure D.2: *Consistency of Convergence (Coarse Model, Low Utilization)*

Percentile	Latency
99-th	0.77 ± 0.03sec
98-th	0.49 ± 0.01sec
95-th	0.30 ± 0.003sec

Table D.3: *Estimated NLPs (Accurate Model, Low Utilization, 120k DLs)*

Utilization	#Downloads around 99-th NLP
Low	62, 735
Medium	24, 126
High	4, 879

Table D.4: *Population around 99-th NLP (Coarse Model, 30x150k DLs)*



a1) 99-th NLP

a2) 99-th OSP

b1) 98-th NLP

b2) 98-th OSP

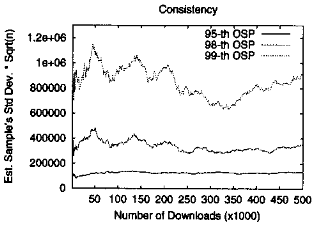
c1) 95-th NLP

c2) 95-th OSP

d1) avg. NL

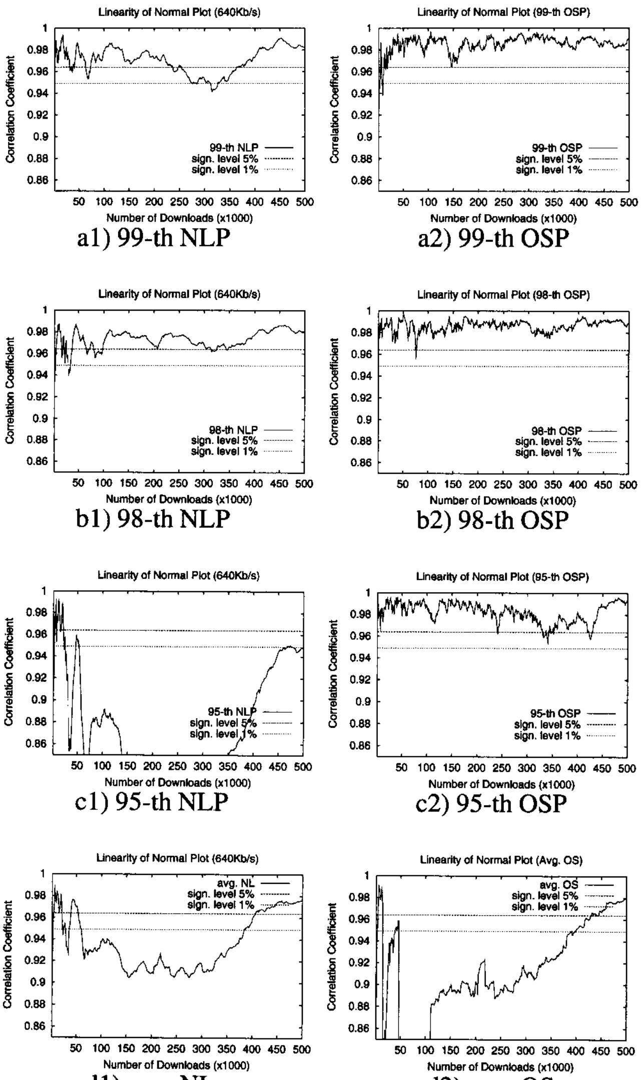
d2) avg. OS

Figure D.3: *Convergence of NLPs, OSPs and avg. NL, avg OS (Coarse Model, Medium Utilization)*



a2) OSPs

Figure D.4: *Consistency of Convergence (Coarse Model, Medium Utilization)*



a1) 99-th NLP

a2) 99-th OSP

b1) 98-th NLP

b2) 98-th OSP

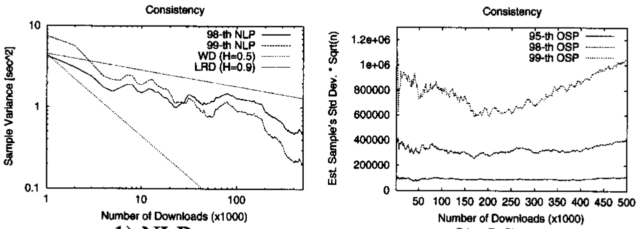
c1) 95-th NLP

c2) 95-th OSP

d1) avg. NL

d2) avg. OS

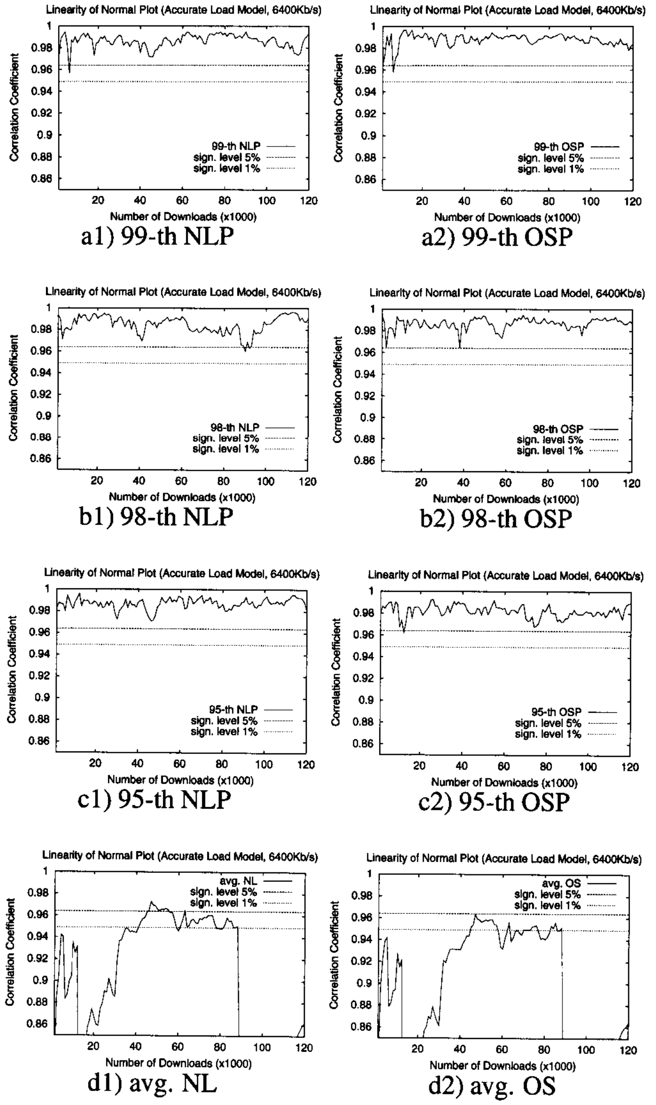
Figure D.5: *Convergence of NLPs, OSPs and avg. NL, avg OS (Coarse Model, High Utilization)*



a1) NLPs

a2) OSPs

Figure D.6: *Consistency of Convergence (Coarse Model, High Utilization)*



a1) 99-th NLP

a2) 99-th OSP

b1) 98-th NLP

b2) 98-th OSP

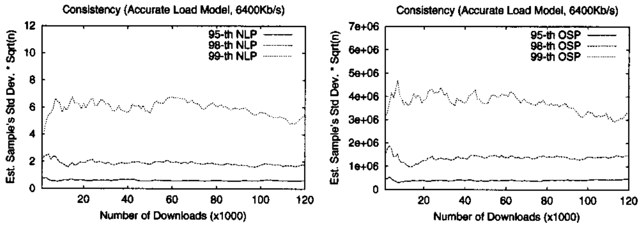
c1) 95-th NLP

c2) 95-th OSP

d1) avg. NL

d2) avg. OS

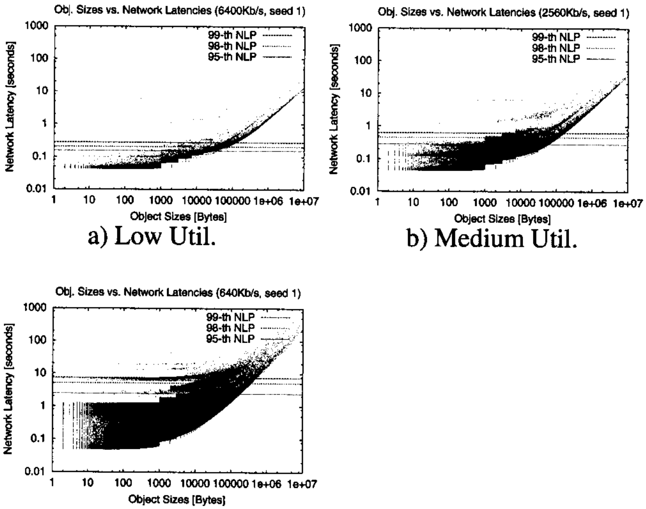
Figure D.7: *Convergence of NLPs, OSPs and avg. NL, avg OS (Accurate Model, Low Util.)*



a1) NLPs

a2) OSPs

Figure D.8: *Consistency of NLPs, OSPs (Accurate Model, Low Util.)*

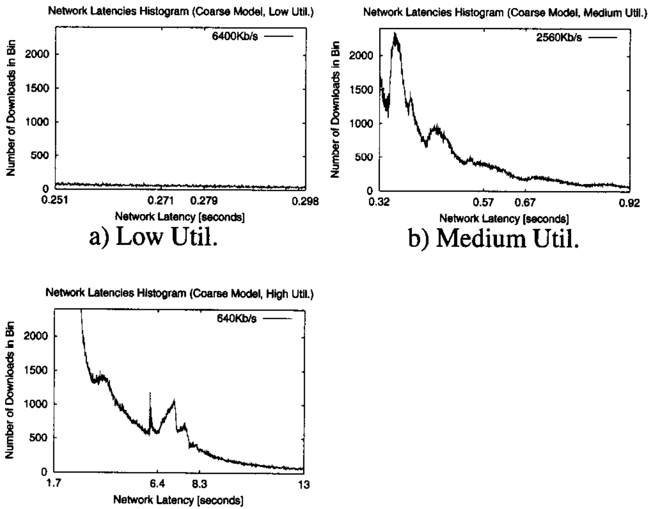


a) Low Util.

b) Medium Util.

c) High Util.

Figure D.9: *Object Size / Network Latency Relation (Coarse Model)*

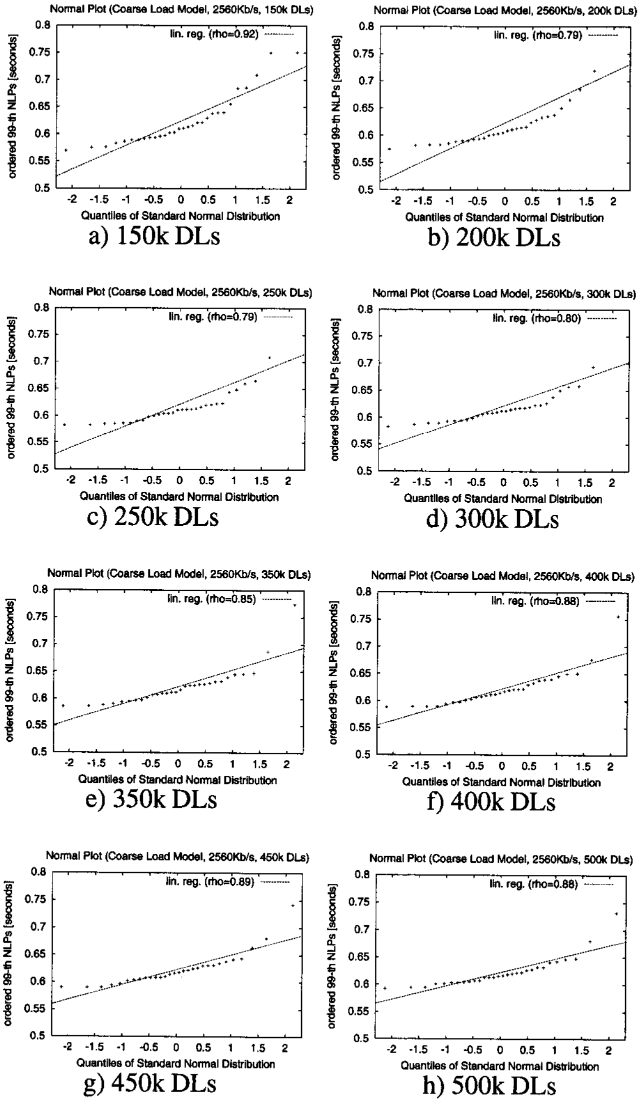


a) Low Util.

b) Medium Util.

c) High Util.

Figure D.10: *Histogram of Network Latencies around the 99-th NLP (Coarse Model)*



a) 150k DLs

b) 200k DLs

c) 250k DLs

d) 300k DLs

e) 350k DLs

f) 400k DLs

g) 450k DLs

h) 500k DLs

Figure D.11: *Normal Plots for the 99-th NLP (Coarse Workload Model, 2560Kb/s)*