Power of Samples & Memory for Randomized Load Balancing

Project Report
EE384Y Packet Switch Architecture II
Spring 2002

Rajan Goyal
Jianying Luo
{rgoyal, jyluo} @ stanford.edu

Stanford University
TABLE OF CONTENTS

1 INTRODUCTION ................................................................................................................................. 3
   1.1 GOAL ................................................................................................................................................. 3
   1.2 REPORT BALANCE ......................................................................................................................... 3
   1.3 ACKNOWLEDGEMENTS ............................................................................................................... 3

2 BACKGROUND AND RELATED WORK .......................................................................................... 3
   2.1 LOAD BALANCING PROBLEM ..................................................................................................... 3
   2.2 RECENT WORK ............................................................................................................................ 3

3 SIMULATION SETUP ......................................................................................................................... 4

4 STUDY RESULT AND ANALYSIS ...................................................................................................... 4
   4.1 SYMMETRIC SERVICE RATE SERVERS ....................................................................................... 4
   4.2 ASYMMETRIC SERVICE RATE SERVERS .................................................................................. 6
      4.2.1 Normalized Asymmetric rate ................................................................................................. 6
      4.2.2 Skewed Asymmetric rate ...................................................................................................... 6
      4.2.3 Extremely Skewed Asymmetric rate .................................................................................... 8

5 CONCLUSION ..................................................................................................................................... 10

REFERENCES ......................................................................................................................................... 10
1 Introduction

1.1 Goal
The goal of the project is to study following questions:
1. What does the sample variance of queue size over all queues in a system look like for SQ(d) [refer to 2.1 for notation def.] in case of symmetric rate servers and for SQ(d, m) [refer to 2.1 for notation def.] in case of symmetric rate servers and in case of asymmetric rate servers?
2. For a group of policies SQ(d, m) where (d+m)=k, k is constant, m \geq 1, what’s the right order of these policies in terms of the effectiveness of load balancing, especially in case of asymmetric rate servers?

1.2 Report Balance
This report consists of five parts. The first part tells the goals of the project. In the second part of the report, we briefly describe the load balancing problem and its state of the art in the academic literature. Our simulation setup is explained in the third part. Then we show the simulation study result and our analysis. Finally we make the conclusion from what we learn in the simulation study.

1.3 Acknowledgements
We are grateful to Prof. Balaji Prabhakar for suggesting this topic and insight. We would also like to thank Devavrat Shah for helpful discussions.

2 Background and Related Work

2.1 Load Balancing Problem
A canonical abstraction of load balancing problem is “supermarket model”. In this model, jobs arrive at the system of N independent server queues according to a rate (N * \lambda) Poisson process. Each server serves the jobs with exponentially distributed service time. The service rate of a server queue i is \mu_i, \forall i (0 \leq i \leq N-1), such that \sum \mu_i = N. If \forall i, \mu_i = 1, the server bank is said with symmetric service rate. Otherwise, it’s with asymmetric service rate. Upon arrival an arriving job is immediately assigned to a server queue according to a particular policy.

There are four types of well-known assignment policies.
1. “Join the shortest queue” policy (in abbreviation: SQ) is well known to be optimal [4]. However when N is large, it becomes hard to implement because of its O(N) complexity.
2. “Join a random queue” policy (in abbreviation: Rand) is easy to implement and essentially each server queue becomes a M/M/1 queue.
3. “Join the shortest of d random queues” (in abbreviation: SQ(d)).
4. “Join the shortest of (d+m) queues” (in abbreviation: SQ(d, m)), where there are d fresh random samples and m memory units storing the shortest queues out of the (d+m) choices for the last job arrival.

2.2 Recent Work
It’s well known that if the server rates are symmetric, SQ(d) performs exponentially better than
Rand with respect to \( \text{Prob}(Q \geq i) \) [5], \( \text{SQ}(d, 1) \) is comparable to \( \text{SQ}(2d, 0) \) [3]; and if the server rates are asymmetric, \( \text{SQ}(d) \) is not stable even if \( d=O(n) \) and \( \text{SQ}(d, 1) \) is stable [2].

3 Simulation Setup
We build the simulation environment similar to [1], using the following three configurations.

1. symmetric service rate servers: There are 1500 server queues. Every server queue has service rate 1.0. Jobs arrive according to Poisson process with a rate of 1500 * 0.99.
2. skewed asymmetric service rate servers: There are 1500 server queues. Among them 25 servers have a high rate with a combined rate of 1000. The rest 1475 low rate servers have a combined rate of 500. Job arrival process is Poisson with a rate of 1500 * 0.99.
3. extremely skewed asymmetric service rate servers: There are 600 server queues. 10 servers have a high rate with a combined rate 599. The rest 590 low rate servers have a combined rate of 1. The job arrival process is Poisson with a rate of 600 * 0.99.

In other words, if \( N_H \) is the number of queues in the high rate class, and \( S_H \) is the total service rate of high rate class servers, then:

\[
\Sigma_i \mu_i = S_H, \quad \forall \quad (0 \leq i \leq N_H) \\
\Sigma_j \mu_j = S_L, \quad \forall \quad (N_H+1 \leq j \leq N) \\
S_H + S_L = N
\]

Only difference between these skewed cases, is the ratio of \( S_H : S_L \).

Note: In order to verify our simulation environment, especially to find out conditions, like right value of \( N \) (number of queues), \( T \) (total number of arrivals) and \( W \) (number of initial arrivals to ignore for warming up of system) for a given \( \lambda \), when the system becomes stable, we calibrated our simulator using the theoretical known results of queue size distribution. As a consequence, it verified the theoretical results as well.

4 Study Result and Analysis
For our study, we concentrated on two metrics: one is the queue size distribution, \( P(Q \geq i) \) and the other is the queue size sample variance, i.e. the variance of queue size of \( N \) queues relative to each other. The first metric tells the delay of the job and the second metric tells the delay jitter and how well load is balanced among server queues.

4.1 Symmetric Service Rate Servers
We conduct a simulation study for all policies \( \text{SQ}(d, m) \) where \( (d+m)=8 \), and \( d \geq 1 \). Probability that a queue size is at least \( i \) \((i \geq 0)\) i.e. \( P(Q \geq i) \) and the probability of sample variance of queue size over all queues are plotted.
The observations and rationale about SQ(d, m) policies with respect to symmetric service rate servers are as follows.

1. The sample variance over all queue sizes for all SQ(d, m) policies are small, because the servers have the same service rates and the effectiveness of load balancing by random sample choices and/or the use of memory.
2. SQ(8,0) has comparably good queue size in contrast with SQ(7, 1), but worse than SQ(7, 1) by a noticeable margin in terms of sample variance. This demonstrates that a policy with at least as one unit of memory can learn about the system through historical assignment transactions and thus effectively prevent any queue to grow its size more than the sizes of the other queues. This results in the reduction of sample variance over all queues.

3. The fact that the sample variances of SQ(7, 1), SQ(6, 2), SQ(5, 3), SQ(4, 4), SQ(3, 5), SQ(2, 6), SQ(1, 7) are in the increasing order demonstrates that given the constraint that \(d+m=8\), in terms of sample variance the gain due to one extra memory capacity is less than the loss due to one decrement of random sample when \(m=1\). Generally, the increasing order of sample variance for all policies SQ\(d, m\) where \(d+m=k\), \(k\) is a constant, \(m=1\), is: SQ\(k-1, 1\), SQ\(k-2, 2\), SQ\(k-3, 3\), \(\ldots\), SQ\(1, k-1\).

4. We also observe that the average queue sizes of SQ(7, 1), SQ(6, 2), SQ(5, 3), SQ(4, 4), SQ(3, 5), SQ(2, 6), SQ(1, 7) are in the increasing order. This tells that as long as there is one unit of memory capacity in the assignment policy, in terms of a single queue size the gain due to one extra random sample is more than the loss due to one decrement of the memory capacity. Generally, the increasing order of a single queue size for all policies SQ\(d, m\) where \(d+m=k\), \(k\) is a constant, \(m=1\), is SQ\(k-1, 1\), SQ\(k-2, 2\), SQ\(k-3, 3\), \(\ldots\), SQ\(1, k-1\).

4.2 Asymmetric Service Rate Servers

Based on the study results for symmetric service rate servers, memory does not help in terms of queue size distribution. In order to explore further the power of memory, we ran simulations on asymmetric service rate of queues. Three cases were considered: normalized asymmetric rate (defined below), skewed asymmetric service rate (refer to Section 3) and extremely skewed asymmetric rate (refer to Section 3).

4.2.1 Normalized Asymmetric rate

This was an attempt to make servers’ service rate random and different from each other still maintaining \(\sum \mu_i = N\). Simulation results were no different from the symmetric service rate in terms of relative standing of \(d+m=k\) set of policies. In order to be concise, we are not putting those results here.

4.2.2 Skewed Asymmetric rate

Please refer to Figure-3, Figure-4 and Figure-5 for following discussion. Figure-3 is for queue in the set of high rate servers while Figure-4 is for queue in the set of low rate servers.

Observations are as follows:
1. It is known that Rand is unstable [3] and is obvious from the graphs, also because of the nature of Rand policy high rate server queues which are small in number are loaded less making system stable.
2. Power of memory is getting visible now in terms of variance of the queue size among queues, even though the relative order for \(P(Q=i)\) is same as in case of symmetric service rate (with \(m=1\)).
Figure 3. Prob(Q ≥ X) for a single high-rate queue (skewed asymmetric service rates and \( \lambda = 0.99 \)).

Figure 4. Prob(Q ≥ X) for a single low-rate queue (skewed asymmetric service rates and \( \lambda = 0.99 \)).
4.2.3 Extremely Skewed Asymmetric rate

Please refer to Figure-6, Figure-7 and Figure-8 for the following discussion. We further skewed the asymmetry to see the effect of samples vs. memory. Now, the power of memory starts to show compared with the earlier results.

Observations:
1. With an extremely skewed system, for the low rate queues, more memory helps in terms of $P(Q_i)$ while in case of the high rate queues, memory does not help much beyond 2. (Note, results in our study match the claims made in [3], i.e. memory helps in improving tails of the queues rather than avg. queue size of the system).
2. Memory improves the sample variance of the queue sizes.

Results:
(Assume that a system is stable i.e. $m \geq 1$)
1. If $d+m$ is constant, $m \geq 1$ is required to make stable but beyond $m > 2$, there is negative affect on queue size distribution due to decrease in $d$. Also, for given value of $d$ & $m$, if we have choice to increase only of them, if $(m \geq 1)$, then it will be better to increase $d$.
2. From sample variance perspective, increase in memory help upto certain extent and when number of samples gets smaller than number of memory, there is negative effect of increase in memory on expense of extra samples.

Intuition without proof yet:
Given that:
1. Extra sample is always good in terms of queue size distribution.
2. Extra memory keeps sample variance of queue sizes low.
Since the system is stable, memory keeps load in most of queues equally distributed (this is
justification for low variance in case of memory). With an extra sample, system explores more queues in each time slot, as a result, arriving jobs are not equally dispatched to the servers (that’s why variance is not low) but most of the queues are loaded.

Figure-6 Prob(Q \geq X) for a single high-rate queue (extremely skewed asymmetric service rates and \( \lambda=0.99 \))

Figure-7 Prob(Q \geq X) for a single low-rate queue (extremely skewed asymmetric service rates and \( \lambda=0.99 \))
Figure-8 probability of sample variance of queue size over all queues (extremely skewed asymmetric service rates and $\lambda=0.99$)

5 Conclusion

In this course project we studied the load balancing effectiveness comparison among SQ(d, m) policies, where $(d+m)=k$, $k$ is a constant, for both symmetric rate server case and asymmetric rate server case.

We’ve found that in general extra sample helps in keeping queue size distribution smaller and extra memory helps in reducing variance among queues. Effect of memory is more visible for asymmetric rate servers.

References


