# Advanced Systems Lab Report

Autumn Semester 2017

Name: YOUR_NAME
Legi: YOUR_LEGI

**Grading**

| **Section** | **Points** |
|---|---|
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| 5 | |
| 6 | |
| 7 | |
| Total | |

## Notes on writing the report **(remove this page for submission)**

The report for the lab needs not be extensive but it must be concise, complete, and correct. Conciseness is important in terms of content and explanations, focusing on what has been done and explanations of the results. A long report is not necessarily a better report, especially if there are aspects of the design or the experiments that remain unexplained. Completeness implies that the report should give a comprehensive idea of what has been done by mentioning all key aspects of the design, experiments, and analysis. Aspects of the system, be it of its design or of its behavior, that remain unexplained detract from the credibility and grade of the report. Correctness is expected in terms of the explanations being logical and correlated with the numbers in the experiments and the design.

Remember that this is a report about the system you have designed and built, about the experiments you have performed, and about how you interpret the results of the experiments and map them to your design and implementation. Please do not contact us seeking confirmation and assurances about, e.g., whether the report is sufficient, your interpretation of the data, validation of concrete aspects of your design, or whether you have done enough experiments. Making those decisions is your job and part of what the course will evaluate.

The report will be graded together with the code and data submitted. **You will need at least 400 points to pass.** You might be called for a meeting in person to clarify aspects of the report or the system. By submitting the report, the code, and the data, you confirm that you have done the work on your own, the code has been developed by yourself, the data submitted comes from experiments your have done, you have written the report on your own, and you have not copied neither code nor text nor data from other sources.

Each section will have to contain a description of the experimental setup, graphs (where applicable) and tables with most important numbers. More importantly, each experiment and modeling result should be accompanied by an explanation that puts the result in the context of your system. Without proper explanations you risk losing most of the points in the experimental sections. Furthermore, it is expected that the interactive law holds for all experiments. In case throughput and response time do not match, it is imperative that you explain why, otherwise you risk losing most of the points for the experiment in question.

In each section you can find a table summarizing the system configuration that should to be used for that particular section. These configurations should be considered as a guideline to help you find interesting configurations where we expect your system to change its behaviour (e.g., going from an under-saturated to a saturated system). If you do not observe such a change inside that range of parameters, you might have to run additional experiments outside the given range. In that case, ask your assistant for advise.

This document provides you with the end-results that have to be included in the report. Please keep in mind that you are encouraged to provide additional graphs and experiments that help you in understanding your system. These additional experiments should be chosen such that they support the claims you make in your explanations and help you illustrate your point.

## Formatting guidelines

We expect you to use this template for the report, but you are free to use another text processor. Keep in mind the following:

- We expect you to submit **a single PDF that has the same section structure as this template, and answers all points we outline here**. If you use this file, you should remove this page with notes, and the short description provided by us at the beginning of sections.

- Keep the same cover page as on this document and **fill out your name and legi number**. Leave the grading table empty.
- Write your name and Legi number on every page.
- The main text should be in **single-column format with 11pt font on A4 paper**. In case you don't start with one of the files provided by us, **for margins use 2.54 cm (1 inch) on all sides**.
- The length of report should be at minimum 25, but should not exceed 50 pages (including figures, tables, etc.).

## How to Generate Load and Populate the Data

To generate load on the system, you will use memtier. When using memtier, you can specify two parameters: the number of threads (CT) and the number of virtual clients per thread (VC). The total number of virtual clients per memtier is CPM=CT*VC. When referring to "number of clients" in the system, it is the number of memtier instances times the number of virtual clients per instance, i.e., NumClients=NumMemtier*CPM.

Increasing NumClients for experiments is always done by first fixing the number of virtual machines (VMs) and memtier instances and then changing the VC setting for all memtiers. In most cases it is enough to increase VC in large increments, unless you want to explore the behavior of your system around the inflexion point or other points which are of interest. In this case, you can use more fine-grained steps.

For all experiments, use 2 CT per VM and between 1 and 32 VC per thread. We recommend to choose at least 6 points in this range. You will have to use up to 3 VMs for clients. In case you use a single middleware, set up each memtier instance with 2 CTs. If you use two middlewares, start two memtier instances per VM (each connected to one middleware) with 1 CT each.

For all experiments conducted in this report, the value size has to be 1024 Bs, and the number of keys 10000 (`--key-maximum=10000`). Before running experiments, populate the key-value store using a write-only workload long enough to write all keys at least once. All sets should have a long expiry time, to ensure that keys are not evicted during the experiments (e.g., `--expiry-range=9999-10000`). The loading of the memcached servers is not part of the measured performance, and the middleware has to be restarted after the initial population, before running experiments.

## 1 System Overview (75 pts)

Describe the implementation of your system and highlight design decisions relevant for the experiments. Explain how messages are parsed and how statistics are gathered in a multi-threaded setting. Provide figures containing all the threads and queues in your system (including the network and the memcached servers). Include illustrations that show how requests of different types are handled (e.g., components involved in processing the request and method calls). Please include all details necessary to understand artifacts and effects in your experiments that arise from your implementation choices.

## 2 Baseline without Middleware (75 pts)

In this experiments you study the performance characteristics of the memtier clients and memcached servers.

### 2.1 One Server

Both, for a read-only and write-only workload plot the throughput and the response time as a function of NumClients. All clients are connected to a single memcached instance.

Use 3 load generating VMs, with one memtier (CT=2) each, and vary the number of virtual clients (VC) per memtier thread between 1 and 32. Show how the behavior of the server changes as we add more clients.

| | |
|---|---|
| Number of servers | 1 |
| Number of client machines | 3 |
| Instances of memtier per machine | 1 |
| Threads per memtier instance | 2 |
| Virtual clients per thread | [1..32] |
| Workload | Write-only and Read-only |
| Multi-Get behavior | N/A |
| Multi-Get size | N/A |
| Number of middlewares | N/A |
| Worker threads per middleware | N/A |
| Repetitions | 3 or more |

#### 2.1.1 Explanation

Describe in which phase the memcached servers are under-saturated, saturated, or over-saturated. Describe how throughput and response time correlate. Explain what further conclusions can be drawn from the experiment.

### 2.2 Two Servers

For a read-only and write-only workload plot throughput and response time as a function of NumClients. The clients are connected to two memcached instances.

Use 1 load generating VM, with one memtier (CT=1) connected to each memcached instance (two memcache instances in total), and vary the number of virtual clients (VC) per memtier thread between 1 and 32. Show how the behavior of the server changes and explain what conclusions we can draw from this experiment.

|  |  |
|---|---|
| Number of servers | 2 |
| Number of client machines | 1 |
| Instances of memtier per machine | 2 |
| Threads per memtier instance | 1 |
| Virtual clients per thread | [1..32] |
| Workload | Write-only and Read-only |
| Multi-Get behavior | N/A |
| Multi-Get size | N/A |
| Number of middlewares | N/A |
| Worker threads per middleware | N/A |
| Repetitions | 3 or more (at least 1 minute each) |

### 2.2.1 Explanation

Describe how this experiment compares to the previous section. Which results are the same and which ones differ? Explain what further conclusions can be drawn from the experiment.

## 2.3 Summary

Based on the experiments above, fill out the following table:

Maximum throughput of different VMs.

|  | Read-only workload | Write-only workload | Configuration gives max. throughput |
|---|---|---|---|
| One memcached server |  |  |  |
| One load generating VM |  |  |  |

Write at least two paragraphs about how both results relate. Describe what is the bottleneck of this setup is. If the maximum throughput for both experiments is the same, explain why. If it is not the case, explain why not. Write down key take-away messages about the behaviour of the memtier clients and the memcached servers.

# 3 Baseline with Middleware (90 pts)

In this set of experiments, you will have to use 1 load generator VM and 1 memcached server, measuring how the throughput of the system changes when increasing the number of clients. Scaling virtual clients inside memtier has to be done as explained in the previous sections. Plot both throughput and response time as measured on the middleware.

## 3.1 One Middleware

Connect one load generator machine (one instance of memtier with CT=2) to a single middleware and use 1 memcached server. Run a read-only and a write-only workload with increasing number of clients (between 2 and 64) and measure response time *both at the client and at the middleware*, and plot the throughput and response time measured in the middleware.

Repeat this experiment for different number of worker threads inside the middleware: 8, 16, 32, 64.

| | |
|---|---|
| Number of servers | 1 |
| Number of client machines | 1 |
| Instances of memtier per machine | 1 |
| Threads per memtier instance | 2 |
| Virtual clients per thread | [1..32] |
| Workload | Write-only and Read-only |
| Multi-Get behavior | N/A |
| Multi-Get size | N/A |
| Number of middlewares | 1 |
| Worker threads per middleware | [8..64] |
| Repetitions | 3 or more (at least 1 minute each) |

### 3.1.1 Explanation

Provide a detailed analysis of the results (e.g., bottleneck analysis, component utilizations, average queue lengths, system saturation). Add any additional figures and experiments that help you illustrate your point and support your claims.

## 3.2 Two Middlewares

Connect one load generator machine (two instances of memtier with CT=1) to two middlewares and use 1 memcached server. Run a read-only and a write-only workload with increasing number of clients (between 2 and 64) and measure response time *both at the client and at the middleware*, and plot the throughput and response time as measured in the middleware.

Repeat this experiment for different number of worker threads inside the middleware: 8, 16, 32, 64.

If in your experiment the middleware is not the bottleneck, repeat the experiment that reaches the highest throughput but using two load generator VMs (each with 2x memtier CT=1) instead of one. Otherwise, explain how you know that the middlewares are the limiting factor in terms of throughput.

| | |
|---|---|
| Number of servers | 1 |
| Number of client machines | 1 |
| Instances of memtier per machine | 2 |
| Threads per memtier instance | 1 |
| Virtual clients per thread | [1..32] |
| Workload | Write-only and Read-only |
| Multi-Get behavior | N/A |
| Multi-Get size | N/A |
| Number of middlewares | 2 |
| Worker threads per middleware | [8..64] |
| Repetitions | 3 or more (at least 1 minute each) |

### 3.2.1 Explanation

Provide a detailed analysis of the results (e.g., bottleneck analysis, component utilizations, average queue lengths, system saturation). Add any additional figures and experiments that help you illustrate your point and support your claims.

## 3.3 Summary

Based on the experiments above, fill out the following table. For both of them use the numbers from a single experiment to fill out all lines. Miss rate represents the percentage of GET requests that return no data. Time in the queue refers to the time spent in the queue between the net-thread and the worker threads.

Maximum throughput for one middleware.

| | Throughput | Response time | Average time in queue | Miss rate |
|---|---|---|---|---|
| Reads: Measured on middleware | | | | |
| Reads: Measured on clients | | | n/a | |
| Writes: Measured on middleware | | | | n/a |
| Writes: Measured on clients | | | n/a | n/a |

Maximum throughput for two middlewares.

| | Throughput | Response time | Average time in queue | Miss rate |
|---|---|---|---|---|
| Reads: Measured on middleware | | | | |
| Reads: Measured on clients | | | n/a | |
| Writes: Measured on middleware | | | | n/a |
| Writes: Measured on clients | | | n/a | n/a |

Based on the data provided in these tables, write at least two paragraphs summarizing your findings about the performance of the middleware in the baseline experiments.

## 4 Throughput for Writes (90 pts)

### 4.1 Full System

Connect three load generating VMs to two middlewares and three memchached servers. Run a write-only experiment. You need to plot throughput and response time measured on the middleware as a function of number of clients. The measurements have to be performed for 8, 16, 32 and 64 worker threads inside each middleware.

| | |
|---|---|
| Number of servers | 3 |
| Number of client machines | 3 |
| Instances of memtier per machine | 2 |
| Threads per memtier instance | 1 |
| Virtual clients per thread | [1..32] |
| Workload | Write-only |
| Multi-Get behavior | N/A |
| Multi-Get size | N/A |
| Number of middlewares | 2 |
| Worker threads per middleware | [8..64] |
| Repetitions | 3 or more (at least 1 minute each) |

#### 4.1.1 Explanation

Provide a detailed analysis of the results (e.g., bottleneck analysis, component utilizations, average queue lengths, system saturation). Add any additional figures and experiments that help you illustrate your point and support your claims.

### 4.2 Summary

Based on the experiments above, fill out the following table with the data corresponding to the maximum throughput point for all four worker-thread scenarios.

Maximum throughput for the full system

| | WT=8 | WT=16 | WT=32 | WT=64 |
|---|---|---|---|---|
| Throughput (Middleware) | | | | |
| Throughput (Derived from MW response time) | | | | |
| Throughput (Client) | | | | |
| Average time in queue | | | | |
| Average length of queue | | | | |
| Average time waiting for memcached | | | | |

Based on the data provided in these tables, draw conclusions on the state of your system for a variable number of worker threads.

## 5 Gets and Multi-gets (90 pts)

For this set of experiments you will use three load generating machines, two middlewares and three memcached servers. Each memtier instance should have 2 virtual clients in total and the number of middleware worker threads is 64, or the one that provides the highest throughput in your system (whichever number of threads is smaller).

For multi-GET workloads, memtier will generate a mixture of SETs, GETs, and multi-GETs. Memtier only allows to specify the maximum number of keys in a multi-GET request. Therefore, be aware that requests can also contain fewer keys than the provided value. It is recommended to record the average size of the multi-GETs. You will have to measure response time on the client as a function of multi-get size, with and without sharding on the middlewares.

### 5.1 Sharded Case

Run multi-gets with 1, 3, 6 and 9 keys (memtier configuration) with sharding enabled (multi-gets are broken up into smaller multi-gets and spread across servers). Plot average response time as measured on the client, as well as the 25th, 50th, 75th, 90th and 99th percentiles.

| | |
|---|---|
| Number of servers | 3 |
| Number of client machines | 3 |
| Instances of memtier per machine | 2 |
| Threads per memtier instance | 1 |
| Virtual clients per thread | 2 |
| Workload | memtier-default |
| Multi-Get behavior | Sharded |
| Multi-Get size | [1..9] |
| Number of middlewares | 2 |
| Worker threads per middleware | max. throughput config. |
| Repetitions | 3 or more (at least 1 minute each) |

#### 5.1.1 Explanation

Provide a detailed analysis of the results (e.g., bottleneck analysis, component utilizations, average queue lengths, system saturation). Add any additional figures and experiments that help you illustrate your point and support your claims.

### 5.2 Non-sharded Case

Run multi-gets with 1, 3, 6 and 9 keys (memtier configuration) with sharding disabled. Plot average response time as measured on the client, as well as the 25th, 50th, 75th, 90th and 99th percentiles.

|  |  |
|---|---|
| Number of servers | 3 |
| Number of client machines | 3 |
| Instances of memtier per machine | 2 |
| Threads per memtier instance | 1 |
| Virtual clients per thread | 2 |
| Workload | memtier-default |
| Multi-Get behavior | Non-Sharded |
| Multi-Get size | [1..9] |
| Number of middlewares | 2 |
| Worker threads per middleware | max. throughput config. |
| Repetitions | 3 or more (at least 1 minute each) |

### 5.2.1 Explanation

Provide a detailed analysis of the results (e.g., bottleneck analysis, component utilizations, average queue lengths, system saturation). Add any additional figures and experiments that help you illustrate your point and support your claims.

## 5.3 Histogram

For the case with 6 keys inside the multi-get, display four histograms representing the sharded and non-sharded response time distribution, both as measured on the client, and inside the middleware. Choose the bucket size in the same way for all four, and such that there are at least 10 buckets on each of the graphs.

## 5.4 Summary

Provide a detailed comparison of the sharded and non-shareded modes. For which multi-GET size is sharding the preferred option? Provide a detailed analysis of your system. Add any additional figures and experiments that help you illustrate your point and support your claims.

# 6 2K Analysis (90 pts)

For 3 client machines (with 64 total virtual clients per client VM) measure the throughput and response time of your system in a 2k experiment with repetitions. All GET operations have a single key. Investigate the following parameters:

- Memcached servers: 2 and 3
- Middlewares: 1 and 2
- Worker threads per MW: 8 and 32

Repeat the experiment for (a) a write-only, (b) a read-only, and (c) a 50-50-read-write workload. For each of the three workloads, what is the impact of these parameters on throughput, respectively response time?

|  |  |
|---|---|
| Number of servers | 2 and 3 |
| Number of client machines | 3 |
| Instances of memtier per machine | 2 |
| Threads per memtier instance | 1 |
| Virtual clients per thread | 32 |
| Workload | Write-only, Read-only, and 50-50-read-write |
| Multi-Get behavior | N/A |
| Multi-Get size | N/A |
| Number of middlewares | 1 and 2 |
| Worker threads per middleware | 8 and 32 |
| Repetitions | 3 or more (at least 1 minute each) |

## 7 Queuing Model (90 pts)

Note that for queuing models it is enough to use the experimental results from the previous sections. It is, however, possible that the numbers you need are not only the ones in the figures we asked for, but also the internal measurements that you have obtained through instrumentation of your middleware.

### 7.1 M/M/1

Build queuing model based on Section 4 (write-only throughput) for each worker-thread configuration of the middleware. Use one M/M/1 queue to model your entire system. Motivate your choice of input parameters to the model. Explain for which experiments the predictions of the model match and for which they do not.

### 7.2 M/M/m

Build an M/M/m model based on Section 4, where each middleware worker thread is represented as one service. Motivate your choice of input parameters to the model. Explain for which experiments the predictions of the model match and for which they do not.

### 7.3 Network of Queues

Based on Section 2, build a network of queues which simulates your system. Motivate the design of your network of queues and relate it wherever possible to a component of your system. Motivate your choice of input parameters for the different queues inside the network. Perform a detailed analysis of the utilization of each component and clearly state what the bottleneck of your system is. Explain for which experiments the predictions of the model match and for which they do not.