

Thesis Specification

A Centralised data aggregation layer across different consuming platforms

February 7, 2015

1 Background

The project in the context of networking, caching and data aggregation. The goal of the project is to investigate different solutions for implementing a caching layer used in a backend middleware, design and develop prototypes for each solution, run set of tests and select the most optimal solution according to the parameters described in Project Description section. The selected solution is then further developed and integrated into an existing middleware provided by the company, and an analysis is then made in a real life scenario.

2 Project Description

Applications frequently require communication with external services. Every external service specifies their own API. The problem can be defined as following: There are a lot applications written for different operating systems and they require frequent communication with many external services. What architecture is the most appropriate for communication between applications and external services?

One of the solutions, is a simple direct communication between application and services. The application will send requests to every service and wait for responses. This solution has several drawbacks: one should maintain many communication layers written on different languages. When the external API changes, one should rewrite every communication layer. This increase the cost of development and time consumption.

The different solution will be to introduce the middle layer between applications and external services. The applications will communicate with middle layer using predefined protocol, middle layer will gather data from external services and send it to the applications. This approach has several benefits: every application will be written with standardised communication API, defined by the middle layer; if external API changes only middle layer should be rewritten; the response time can be decreased by caching the data in the middle layer.

The project tries to break the connection between applications and external services by providing the middle layer. The applications will communicate with the middleware using predefined format that is the same for every platform. The middleware will manage the requests, translate them to the format appropriate for the external services and forward them to the appropriate servers. It will also manage the responses from servers and send them back to the applications.

The main goal of the project is to select the optimal strategy/architecture for caching layer according to the parameters specified below:

- Hit/Miss/Error rates
- Speed of caching mechanism
- Hardware utilization (space used etc)
- Costs
- Scalability
- Response times to clients
- Conforming to standards
- Redundancy (solution surviving one or more malfunctioning servers)

3 Methods

The goal of the first phase of the project is to study various solutions to the intended middle-ware, each of which will have its own implementation and design, possibly bound to a specific programming language. As such, the methods used for building prototypes for each identified solution is not possibly to specify in great detail at this point in time, but the following technologies / languages have been identified as possible areas of interest in meetings with the company at which this thesis will take place:

Databases:

- Redis
- Memcached
- MongoDB
- MySQL

Web Caching Related Protocols and Standards:

- HTTP 1.0/1.1 protocols
- Internet Cache Protocol

4 Relevant Courses

- Advanced computer architecture
- Wireless Communication and Networked Embedded Systems
- Large scale programming

5 Delimitations

The prestudy phase will focus around finding possible solutions and creating a chart based on findings, allowing for initial comparisons between them. Given that the intended implementation phase of the thesis work is approximately 12 weeks, and that the selected solution will require additional work to be integrated into the existing middleware of the company is estimated at 4 weeks, that leaves roughly 8 weeks to create prototypes. If more than 3 possible solutions are identified, I will perform a first selection of 3 solutions based on the prestudy information alone. This means that under no circumstances there will be no more than 3 prototypes developed.

The additional integration using the selected solution as part of the existing middleware provided by the company should be implemented in practice, but could instead be part of a theoretical discussion where the end result is only derived from previous tests on the prototypes and learnings during the project.

6 Time Plan

Phase Name	Task	Estimated time
Prestudy phase	Finding and reading literature on topics noted above	26 January - 9 February
	Meeting with the reviewer	9 February
	Identifying possible solutions	10 February - 12 February
	Creation of charts and other material to easily compare solutions	13 February - 22 February
	Presentation solutions to company, selection of solutions (if more than 3)	23 February
	Meeting with the reviewer	25 February
Implementation phase	Developing a prototype of solution A	24 February - 10 March
	Developing a prototype of solution B	11 March - 25 March
	Meeting with the reviewer	25 March
	Developing a prototype of solution C	26 March - 16 April
	Data gathering / Benchmarking and tests / technical comparisons / discussion (parameters listed above)	16 April - 23 April
	Meeting with the reviewer	23 April
	Selection of a single prototype (solution) for further implementation in a real world scenario, running on existing middleware.	24 April
	Integration work, using prototype as basis.	24 April - 8 May
	Tests in actual middleware / benchmarking and discussion regarding how well the integrated implementation works in comparison to expectations.	11 May - 18 May
	Meeting with the reviewer	18 May
Report	Conclusion, Further Discussions	19-20 May
	Presentation preparation	21 May - 25 May