# Extreme Computing - First Practical Assignment

Stratis Viglas

November 3, 2013

## 1 Introduction

The goal of this assignment is to write a 3000-word essay (roughly, 6-8 pages of text) on the infrastructure and uses of cloud computing. More specifically, you are given a description of a computing problem. You have to analyse this decription further, identify how it relates to the issues touched upon during the lectures, and then propose a computing infrastucture to address the problem at hand.

The proposed infrastructure can be in the form of an architectural diagram, or a simple bulleted list of points. There are no requirements set in stone, choose whichever form you think makes more sense for your presentation style.

Also, this being an essay, means that there is no right or wrong answer. Of course, some answers make more sense than others (for instance, saying "I would use 10,000 monkeys to write random code for me" is an interesting experiment but makes little sense). The basic rule of thumb is that if you can argue about what you have written, in the sense of exhibiting a clear way of thinking and providing supporting objective arguments, then the result is perfectly acceptable.

## 2 Context

Massbase is a new startup company that is active in the area of data gathering and analytics. In particular, they are in the business of building online recommendation engines, but with a twist. Their goal is to scavenge tweets from Twitter and status messages from Google+ and Facebook–or any social networking medium that is hip these days–and identify relevant user interests. They call each such piece of information a blip. Each blip, apart from

the actual text, is also tagged with certain structured pieces of information, provided by the APIs of the aforementioned services: the originating user; a location; perhaps if it contains references to other users; and so on.

Massbase will operate with subscriptions. That is, users of the service will register with the company and allow them access to their blips. Massbase have the following rough set of specifications/requirements for the service they aim to provide:

Monitor the services mentioned above for new blips. For each blip, tag it with user and location and store it in their system. Given that these services are interconnected, they want to have deduplication in place (for instance, a tweet and a status update with the same text and from the same person should count as one blip and only be stored once). They do not care if they do it in real time or later on, but eventually (say, one day after the blips were first encountered) they do not want any duplicates in their store. Depending on the service, blips may come already tagged with a value (e.g., hash tags in tweets); these tags should be isolated and stored by themselves, but without losing the connection to the blip. In addition to explicit tags, there should also be implicit tags, i.e. tags one can identify by performing some post-processing of the blip. This need not necessarily be instant – they can afford to do it in bulk, though latency should not be more than a couple of hours after the blip was first encountered. They want to be able to identify networks of users. For instance, users who tend to respond to one another through tweets are most likely in the same social cycle. Same thing applies to common tags: users using common tags repeatedly, most likely share similar interests. They should be able to aggregate among multiple dimensions. For instance, they would like to group tags by location, or locations by tags. Given the volume of data, such computations are to be performed in bulk. However, the results should be stored in an efficient way that allows for fast retrieval, so they can provide their service. They would also like to be able to detect trends, but ideally they would also like to have a notion of tag disambiguation. They should be able to detect whether a blip refers to a product, or an event, or a movie, or a TV series, etc. The way they envisage doing this is by identifying certain information in the blip. For instance, the use of watched will most likely refer to a movie, or a TV series. What they effectively want to do is to send a message to each user after they detect a blip from him/her to alert him of related information. An example might be a user interested in theatre: they should be able to provide her/him with recommendations for current theatre performances in her/his area that she/he might want to see. This can be done either in real-time (for instance, using a direct blip to the user) or in bulk (for instance, an overnight email with a list of recommendations). Recommendations might be in the form

of products/events or in the form of other users. In addition to the locality aspects of the service, there should be a temporal aspect as well. Recently used tags should have more importance than tags used a few months ago, even if the frequency of earlier tags is higher. The system should be always running and it should be elastic, in the sense that it should be able to allocate resources dynamically to the different types of computation. For instance, if there is a high load of blips coming in, it is more important to capture the stream of blips as opposed to perfoming aggregation on the background. Likewise, if the system is low on load, it should be able to scale down, either by reallocating resources to background processes, or even freeing them up completely.

Being impressed with you having taken Extreme Computing and your excellent mark in the course, Massbase made you a job offer you could not resist and have decided to put you in charge of infrastructure. What they want you to do is come up with a proposal for the architecture of the underlying system they will use. Specifically, the investors have given the company an infrastructure budget to be spent over the next two years in any way the company sees fit – at which point the investors in the startup will evaluate whether Massbase will receive further funding, or they will pull the plug. You responsibility in Massbase is to make the best use of this budget – in addition to providing the overarching system design.

# 3 Questions to think about:

Should Massbase use a structured parallel database for the task, or should they go for a more virtualised approach, i.e., should they go for a cloud-based infrastructure? Given that they are operating under a budget, should they build their own private cloud, or rent it from one of the existing cloud providers? You can make assumptions about the budget (say, a total of B million pounds to be spent over two years) and how much renting a cloud costs (e.g., x pounds/GB/day, or along those lines) or whatever else you deem important in your computations. Can they actually use a one-size-fits-all solution, or do they have to compromise in the form of a hybrid one? If so, which parts can be cloud-based, and which parts require standard solutions? How far can they go in automating the provisioning of resources? Will they need dedicated personnel to do that, or is there a clear way to do this with the chosen infrastructure?

# 4 Structure

Below is a potential (and rough) structure for your essay. You do not need to follow it, but it might be helpful in terms of organising your thoughts:

Introduction Design considerations (what are the salient aspects of the problem, and what technology currently exists to support them?) Alternatives (including advantages and disadvantages of each) Proposed solution (including arguments of why you think it best fits the problem at hand) Conclusion

# 5 Marking guidelines

There is a total of 100 marks available. The marks will be awarded as follows:

10 marks for writing quality (so please proofread your essay before submission); 40 marks for presenting the use cases and commenting on their feasibility. You should identify any common patterns that you have used to provide your proposal; note that you need to focus not only on what is already possible, but also on what needs to be done and what should be dropped as a goal; 20 marks for the proposed solution and how well it aligns to the presented arguments; 20 marks for your arguments and well they are supported by existing work; 10 marks discrectionary (e.g., general style of writing, exhibiting off-the-beaten-track thinking, etc.).