Performance modelling and simulation of skewed demand in complex systems

Interim Report

Stephen Shephard SN: 160337206

School of Computing Science, Newcastle University, Newcastle upon Tyne, NE1 7RU
`s.shephard2@newcastle.ac.uk`

1 Introduction

There are many high-profile examples of whole IT systems brought down by skewed customer demand for part of their services. Customers were prevented from using any part of the London 2012 Olympic ticketing website on launch day to avoid demand overloading the system [6]. HBO Go was brought down by demand for the finale of "True Detective" [3]. Apple's iTunes Store suffered outage on the launch day of the iPhone 7 (new iPhone registration is carried out via an iTunes function) [7].

It is possible to design and build more resilient systems through effective use of Cloud technologies where higher than normal demand for one function or type of resource would not block access to the others. Skewed demand may be isolated so that it only affects parts of a system, or shared equally between different components. (The system may also adapt to demand by elastic scaling of resources, but this will be beyond the scope of the project).

In a complex system we need to examine how these components interact. What impact would a particular choice of middleware have on the demand at the database, for example? To examine these questions we will choose an approach to build models of selected Cloud technology components that may be combined into different system architectures.

2 Aims and Objectives

The aim is to demonstrate that PEPA (Performance Evaluation Process Algebra) [4] may be used to construct models that are good predictors of the behaviour of real systems under skewed demand scenarios, and provide insights into the effectiveness of Cloud technologies for handling skewed demand.

2.1 Objectives

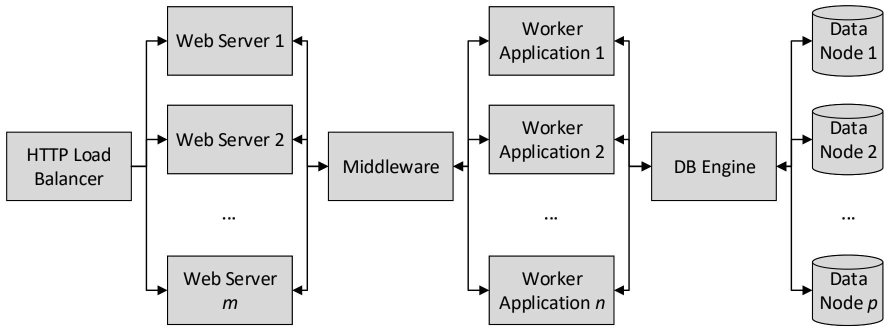
1. Show that distributed databases and middleware queues may be modelled in PEPA
2. Show that these PEPA component models may be used to test skewed demand
3. Show that the component models may be composed into system models and their behaviour under test is different to the components alone
4. Build test system architectures for the PEPA models and show that the test system behaviour matches that predicted by the models
5. Identify areas of further work

3 Method

We will consider an example system based on the Olympic ticketing use case above, where tickets will be available for a number of sports. Such an application may be generalised to any system for allocating and releasing other resources with variable demand.

In all system architectures users will access the system from a web-based front end. Tickets will be stored in a database partitioned across several data stores. In between the web servers and database will be worker applications to service user requests, connected to the web servers by some middleware (see Figure 1).

Fig. 1. Ticketing application distributed architecture



The components modelled will be distributed database nodes using horizontal partitioning [1], distributed database nodes with Cassandra-style replication [5] and a shared middleware queue [2]. These will be tested then composed into three system architectures - *simple microservices* (separate worker applications and databases for Athletics and Cycling tickets), *distributed database via a shared queue* and *distributed database with replication via a shared queue*. Finally these system architectures will be built on Microsoft Azure and compared with the models.

4 Progress to date

I've completed the following tasks to date. See Table 1 for done dates and percentage completion against each of the tasks in the timed work plan.

– Reviewed "Investigating Cloud Technologies to Maximise Availability of Oversubscribed Resources" for Research Skills module
– Agreed initial scope of "Performance modelling and simulation of skewed demand in complex systems" and submitted Ethics form
– Carried out further background learning and research into technology choices
– Set up GitHub structure for Model, Code and Report development and produced outline and timed work plans
– Produced and tested PEPA component models
– Produced PEPA system models

5 Timed work plan

There is a detailed Microsoft Project Plan in the project git repository at https://github.com/sshephard2/msc-project. Table 1 shows a high-level summary.

5.1 Risks

1. Development may take longer than expected due to lack of experience in Java Spring or AngularJS. In mitigation, I have already developed a simple Angular to Cassandra Java Spring application during the Learning phase.
2. The models may not after all be good matches to the behaviour of real systems. There is contingency time in the plan, but this may also be an interesting result for discussion.

Table 1. Planned tasks (subtasks in italics)

Task	Start	End	Complete
Agree initial scope	Tue 25/04/17	Tue 25/04/17	100%
Learning	Fri 05/05/17	Fri 02/06/17	100%
Literature Review	Fri 05/05/17	Tue 30/05/17	100%
LaTeX	Fri 05/05/17	Fri 02/06/17	100%
Java Spring	Thu 18/05/17	Fri 26/05/17	100%
Databases	Wed 17/05/17	Fri 26/05/17	100%
Organisation	Wed 03/05/17	Fri 05/05/17	100%
Git structure	Wed 03/05/17	Wed 03/05/17	100%
Outline plan	Fri 05/05/17	Fri 05/05/17	100%
Ethics Form	Mon 08/05/17	Mon 08/05/17	100%
Interim Report	Mon 05/06/17	Wed 14/06/17	25%
Component Models	Wed 10/05/17	Tue 06/06/17	100%
System Models	Tue 09/05/17	Tue 20/06/17	89%
Build Applications	Wed 21/06/17	Tue 25/07/17	0%
Simple Microservices	Wed 21/06/17	Thu 29/06/17	0%
Shared Queue + Distributed DB	Fri 30/06/17	Wed 12/07/17	0%
Shared Queue + Distributed DB with replication	Thu 13/07/17	Tue 25/07/17	0%
Presentation	Wed 26/07/17	Fri 11/08/17	0%
Dissertation	Thu 15/06/17	Fri 25/08/17	0%
Introduction and Background	Thu 15/06/17	Mon 19/06/17	0%
Models	Wed 21/06/17	Tue 27/06/17	0%
Application Report	Wed 26/07/17	Tue 01/08/17	0%
Conclusions	Wed 02/08/17	Fri 04/08/17	0%
Ready for review	Fri 04/08/17	Fri 04/08/17	0%
Review updates	Mon 14/08/17	Fri 18/08/17	0%
Dissertation Due	Fri 25/08/17	Fri 25/08/17	0%

References

1. Agrawal, S., Narasayya, V., Yang, B.: Integrating vertical and horizontal partitioning into automated physical database design. In: Proceedings of the 2004 ACM SIGMOD international conference on Management of data. pp. 359–370. ACM (2004)
2. Curry, E.: Message-oriented middleware. Middleware for communications pp. 1–28 (2004)
3. Dan Deeth, Sandvine: Hbo goes down (2014), http://www.internetphenomena.com/2014/03/hbo-goes-down/, [Online; accessed 15-March-2017]
4. Hillston, J.: A compositional approach to performance modelling. Computers & Mathematics with Applications 32(6), 136 (1996)
5. Lakshman, A., Malik, P.: Cassandra: a decentralized structured storage system. ACM SIGOPS Operating Systems Review 44(2), 35–40 (2010)
6. Nick Pearce, Telegraph: London olympics 2012: ticket site temporarily crashes as it struggles to cope with second-round demand (2011), http://www.telegraph.co.uk/sport/olympics/8595834/London-Olympics-2012-ticket-site-temporarily-crashes-as-it-struggles-to-cope-with-second-round-demand.html, [Online; accessed 2-March-2017]
7. The Next Web: itunes is down for many users around the world (2016), https://thenextweb.com/apple/2016/09/16/itunes-store-is-down-for-some-users, [Online; accessed 15-March-2017]