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) [8].

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).

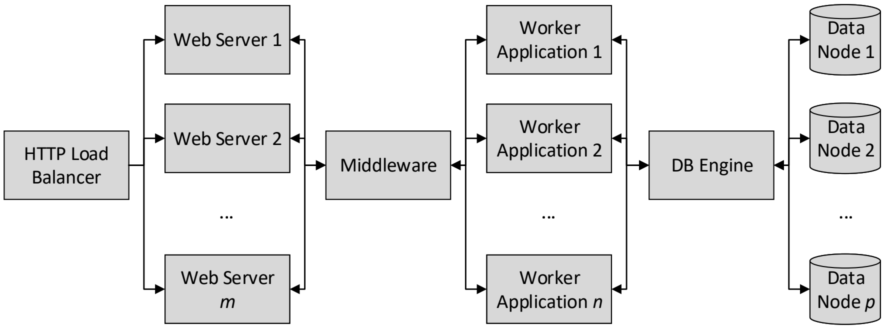
In a complex system we need to examine how these components interact. What impact would a particular choice of middleware have on the demand at the database, for example? To examine these questions we will choose an approach to build models of selected Cloud technology components that may be combined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

2 Aims and Objectives

We will consider an example system based on the Olympic ticketing use case above. Such a ticketing application may be generalised to any system for allocating and releasing other resources with variable demand. Tickets will be for a multi-sport event, and each will consist of a ticket type (the sport), date, row, and seat number.

The system may be implemented using different distributed architectures but all will have the same features. 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.

Fig. 1. Ticketing application distributed architecture



Models will be built in PEPA (Performance Evaluation Process Algebra) [4] for the following Cloud technology components:

- Databases
  - Separate databases
  - Distributed database nodes using horizontal partitioning [1]
  - Distributed database nodes with Cassandra-style replication [5]
- Shared middleware queue [2]
- Event streaming application for eventual consistency architectures [7]

The PEPA component models will be tested then composed into a set of different system architectures below, which will also be built on the Microsoft Azure platform. Both the models and real systems will be tested under different scenarios of skewed demand.

Simple microservices Two separate databases, one for Athletics tickets, one for Cycling.

Operational microservices A more 'natural' microservices architecture partitioning the system by operation (Book, Search, Return) with a separate database for each. The databases maintain eventual consistency via an event streaming application e.g. using Kafka.

Shared queue middleware Requests via a shared queue to worker applications going to a distributed database with two nodes, Athletics and Cycling.

Distributed database with replication Requests via a shared queue to worker applications going to a distributed database with three nodes, Athletics, Cycling and Diving, where each partition is replicated on another node.

Objectives. The objectives are to determine whether or not PEPA models of the components examined, once composed into architectural models:

- are good predictors of the behaviour of real systems under skewed demand scenarios
- provide insights into the effectiveness of Cloud technologies for handling skewed demand.

3 Progress

Done	Task
15/03	Completed "Investigating Cloud Technologies to Maximise Availability of Oversubscribed Resources" for Research Skills module
25/04	Agreed initial scope of "Performance modelling and simulation of skewed demand in complex systems"
05/05	Documented first outline plan in GitHub
08/05	Submitted Ethics form
31/05	Produced PEPA models for all except Event Streaming component and Operational Microservices archiecture

4 Plan

Start	Due	Task
05/06	16/06	Produce interim report
..	..	..
03/08	09/08	Produce presentation
10/08	11/08	Attend presentation
12/08	25/08	Update and submit final version of dissertation

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. Posta, C.: The hardest part about microservices: Your data (2016), https://developers.redhat.com/blog/2016/08/02/the-hardest-part-about-microservices-your-data/, [Online; accessed 5-March-2017]
8. 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]