CS554 Project Ideas

Fabriq: Bench - Benchmarking mainstream Distributed Message Queues

Overview

Distributed Message Queues (DMQ) have proven to be essential building blocks in distributed computing towards the support for fine granular workloads. Distributed message queues such as Amazon's SQS or Apache's Kafka have been used in handling massive data volumes, content delivery, and many more. They have also been used in large scale job scheduling on public clouds. However, even these frameworks have some limitations that make them incapable of handling large scale data with high efficiency and low latency. Fabriq is a distributed message queue that runs on top of a Distributed Hash Table. The design goal of Fabriq is to achieve scalability and near perfect load balancing, while handling the operations with minimal overhead and low latency. Moreover, Fabriq is persistent, reliable and consistent. The results show that Fabriq was able to achieve high throughput in both small and large messages.

The goal of this project is to analyze the architecture and to benchmark some of the mainstream DMQs (Kafka, CQS, Hedwig, Azure Service Bus), and compare those to Fabriq. You will evaluate these Message Queues on Amazon EC2 at scales up to 128 VM instances.

Relevant Systems and Reading Material

- Fabriq paper: (Available on Blackboard)
- Kafka: http://research.microsoft.com/en-us/um/people/srikanth/netdb11/netdb11papers/netdb11-final12.pdf , http://kafka.apache.org/documentation.html
- CQS: http://www.slideshare.net/openstack/open-source-versions-of-amazons-sns-and-sqspptx , https://github.com/Comcast/cmb
- Hedwig: https://cwiki.apache.org/confluence/display/BOOKKEEPER/HedWig
- Azure Service Bus: http://azure.microsoft.com/en-us/services/service-bus/

Methodology

You will analyze the architecture differences of the above mentioned systems and evaluate their performance.

Preferred/Required Skills

- Required: Linux, Shell scripting
- Preferred: Any scripting language that you are comfortable with to allow you conduct the experiments.

Evaluation and Metrics

Operation latency and its distribution, throughput, scalability, CDF graph; expected scale is at least 128 VM instances on Amazon EC2.

Project Mentor

Iman Sadooghi, isadoogh@iit.edu