CS554 Project Ideas

CloudKon:DTS - Distributed Task Scheduling with Amazon SQS

Overview

Predictions are that by the end of this decade, we will have exascale system with millions of nodes and billions of threads of execution. Unfortunately, today's job schedulers have centralized Master/Slaves architecture (e.g. Slurm, Condor, PBS, SGE), where a centralized server is in charge of the resource provisioning and job execution. This architecture has worked well in modest scales and coarse granular workloads, but it has poor scalability at the extreme scales of petascale systems with fine-granular workloads. The goal of this project is to leverage Amazon Simple Queuing Service (SQS) as a public cloud service to provide a scalable task scheduling system that supports Many Task Computing (MTC) workloads. SQS is a distributed queue service with the purpose of providing content delivery on extreme scales. In this project you will be designing a task scheduling system that uses SQS as its building block. An important point to notice is that SQS is not explicitly designed for this purpose. Therefore there are some requirements for MTC type workloads that cannot be met by SQS. For example SQS guarantees at least once message delivery, but not exactly once delivery. However in many MTC workloads it is necessary to provide exactly once task execution. The system designer has to make sure that the system satisfies these requirements using other components and tools. Furthermore, systems such as CloudWatch should be used to dynamically provision EC2 resources under increasing loads. The solution should be compared to Falkon, Sparrow, Slurm, and CloudKon. Note that this project essentially re-implements CloudKon. This is a project for 1~2 people.

Relevant Systems and Reading Material

Amazon SQS:

- http://aws.amazon.com/sqs/
- http://sqs-public-images.s3.amazonaws.com/Building_Scalabale_EC2_applications_with_SQS2.pdf
- http://awsdocs.s3.amazonaws.com/SQS/latest/sqs-gsg.pdf

Many Task Computing paper: I. Raicu, Y. Zhao, I. Foster, "Many-Task Computing for Grids and Supercomputers," 1st IEEE Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS) 2008. http://datasys.cs.iit.edu/events/MTAGS08/MTAGS08_p25.pdf

Sparrow: Scalable Scheduling for Sub-Second Parallel Jobs; http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-29.pdf

SLURM paper: https://e-reports-ext.llnl.gov/pdf/241220.pdf

CloudKon: http://www.cs.iit.edu/~iraicu/research/publications/2013_Qual-IIT_CloudKon.pdf

Methodology

This is a component based project. In order to implement this system, you will need to implement these components: Client, Worker, SQS, Dynamic Provisioner, Monitoring component, etc.

Preferred/Required Skills

Programming language choice: Java, C/C++, Python

Skills/knowledge: REST API, Amazon EC2, Amazon SQS, Amazon DynamoDB, Linux Bash Scripting, distributed queues.

Performance Metrics

Throughput, Latency, Efficiency, Utilization

Project Mentor

Iman Sadooghi: isadoogh@iit.edu, http://datasys.cs.iit.edu/~isadooghi/