CS554 – CloudKon :DTS

Distributed Task Scheduling with Amazon SQS

### people involved.

Kamalnath N.G (CWID : A20260987)

Rajagopal Parthasarathi(CWID:A202674919)

### Abstract

Task scheduling and execution over large scale, distributed systems plays an important role on achieving good performance and high system utilization. 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.

### Problem statement

Predictions are that by the end of this decade, we will have exascale system with millions of nodes and billions of threads ofexecution. Unfortunately, today’s jobschedulers 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.

### RELATED WORK

The University of Wisconsin developed one of the earliest job schedulers, Condor [1], to harness

the unused CPU cycles on workstations for long-running batch jobs. Slurm [2] is a resource manager designed for Linux clusters of all sizes.Portable Batch System (PBS) [3] was originally developed at NASA Ames to address the needs of HPC, which is a highly configurable product that manages batch and

inter-active jobs, and adds the ability to signal, rerun and alter jobs. In 2007, a light-weight task execution framework, called Falkon [4] was developed. Falkon also has a centralized architecture, and although it scaled and performed magnitude orders better than the state of the art, its centralized architecture will not even scale to petascale systems. Sparrow is another scheduling system that focuses on scheduling very short jobs that complete within hundreds of milliseconds. It has a decentralized architecture that makes it highly scalable.

### Proposed Solution

The goal of this project is to implement a job scheduling/management system that satisfies four

major objectives:

1. **Scale**: Offer increasing throughput with larger scales through distributed services

2. **Load Balance**: Offer good load balancing at large scale with heterogeneous workloads

3. **Light-weight**: The system should add minimal overhead even at fine granular

workloads

4. **Loosely Coupled**: Critical towards making the system compact and robust.

CloudKon is a project which already does what we are expecting form the project. This project is to re implement CloudKon and understand and learn the complexities involved in creating a distributed task scheduling framework in a cloud based environment

We intend to modify the DynamoDB component of CloudKon with alternative lightweight DHT /java based caches. We are planning to evaluate ZHT[6] and Hazlecast[7] for the same and choose one of them for this purpose.

### Evaluation

Throughput, Latency, Efficiency, Utilization of the modified CloudKon will be measured against the original results obtained for CloudKon presented here[5].

### timelines

### Deliverables

Project report.

Code Base with documentation.

### CoNclution

Through this project we intend to understand how to create a scalable, light weight, dynamic scheduling framework which can work in petascale range.

through the process of creating this project intend to understand the following:

Amazon SQS, Amazon EC2 : the various intricacies involved with these Cloud component.

ZHT : How to set up an ZHT in cloud environment. How to interface to ZHT through java.

Hazlecast : how to benefit form using a java based cache in a cloud based environment.

### REFERENCES

[1] D. Thain, T. Tannenbaum, M. Livny, “Distributed Computing in Practice: The Condor Experience” Concurrency and Computation: Practice and Experience 17 (2-4), pp. 323-356, 2005.

[2] M. A. Jette et. al, Slurm: Simple linux utility for resource management. In In Lecture Notes in Computer Sicence: Proceedings of Job Scheduling Strategies for Prarallel Procesing (JSSPP) 2003 (2002), Springer-Verlag, pp. 44-60

[3] B. Bode et. al. “The Portable Batch Scheduler and the Maui Scheduler on Linux Clusters,” Usenix, 4th Annual Linux Showcase & Conference, 2000.

[4] I. Raicu, et. al. “Falkon: A Fast and Light-weight tasK executiON Framework,” IEEE/ACM SC 2007

[5] CloudKon http://www.cs.iit.edu/~iraicu/teaching/CS554-F13/proj/CloudKon-DTS.pdf

[6] [ZHT: A Light-weight Reliable Persistent Dynamic Scalable Zero-hop Distributed Hash Table, IPDPS 2013](http://datasys.cs.iit.edu/projects/ZHT/ZHT-CRC-PID2666213-Final.pdf)

[7]Hazlecast : http://www.hazelcast.com/