CloudKon: DTS

Distributed Task Scheduling with Amazon STACK

Contents

[Team 2](#_Toc367024035)

[Abstract 2](#_Toc367024036)

[Problem Statement 2](#_Toc367024037)

[Related Work 2](#_Toc367024038)

[Proposed Solution 2](#_Toc367024039)

[Evaluation Metrics 3](#_Toc367024040)

[References 3](#_Toc367024041)

# Team

* Kamalnath N.G (CWID : A20260987)
* Rajagopal Parthasarathi(CWID:A202674919)

# Abstract

Many Task computing aims to bridge the gap between High Performance Computing and High Throughput Computing. Tasks may be small or large, uniprocessor or multiprocessor, compute-intensive or [data-intensive](http://en.wikipedia.org/wiki/Data-intensive_computing). But generally MTC tasks are communication intensive. Task Scheduling of MTC tasks 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 stack 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 of execution. 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:

* **Scalability -**  Throughput increases with the increase in number of distributed node
* **Load Balancing -**  Efficient and effectiive implementation of system to distribute the workload across all the interacting nodes.
* **Light-weight** - System should involve minimal overhead even while executing fine granular workloads.
* **Loosely Coupled** - Critical Design element to achieve efficient implementation of distributed execution fabric

CloudKon is an effective implementation for distributed task scheduling framework in a cloud based environment. Our proposal is to implement an system similar to CloudKon over Amazon stack. The key differentiators of our solution from CloudKon are

* Dynamic provisioning engine substituting the Amazon CloudWatch.
* Evaluate ZHT[6] or Hazlecast[7] as replacement for the DynamoDB component of CloudKon
* Optimize the implementation to support different types of workloads- Bag of Tasks,Fan in,Fan out and Pipeline

# Evaluation Metrics

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

# 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/>