PROJECT PROPOSAL

CS550, Fall 2013, Anusree Kailash

PROPOSAL TOPIC: IMPLEMENT AND ANALYZE A DISTRIBUTED RESOURCE MANAGEMENT AND JOB ALLOCATION ALGORITHM USING A DHT

Introduction:

In modern high performance computing, the use of large scale distributed systems ensure that job allocation and resource management with an eye on efficient load-balancing is key criteria in determining the scaling performance of the system.

There are many implementations of DHT in place today which allow the shared queue of jobs between nodes. My goal is to study and utilize a popular DHT and write a simple task scheduling application which takes into account scalability and load balancing to perform the jobs in the queue. Fault tolerance is also to be incorporated to prevent jobs from being lost due to node crash.

Some of the DHT being considered for this project is Cassandra, Chord and Hazelcast. Currently there are some task scheduling algorithms written which are centralized(Eg: Falkon,Sparrow) which are found to be not scaling well[VII]. These will be studied to understand the underlying factors to be considered while implementing a task scheduler. The decentralized algorithms also often rely on certain centralized functions to reduce the complexity of the system. In such cases, a comparison can be made and new approaches may be arrived at to mitigate the complexity.

In this project, I plan to

1. study various distributed task scheduling systems and distributed hash table implementations in existence today
2. based on the previous study (1) implement a distributed system job scheduling application using a well-known distributed hash table
3. implement a load balancing system without need of a central load balancing server.
4. implement a fault tolerance module to take care of node failure scenarios without loss of data/job.
5. analyze the performance of the implemented system for the scaling of number of nodes for the same work load
6. analyze the performance of the system for the scaling of work load for fixed number of nodes
7. conclude on key observations and improvement ideas for such a decentralized job scheduling system

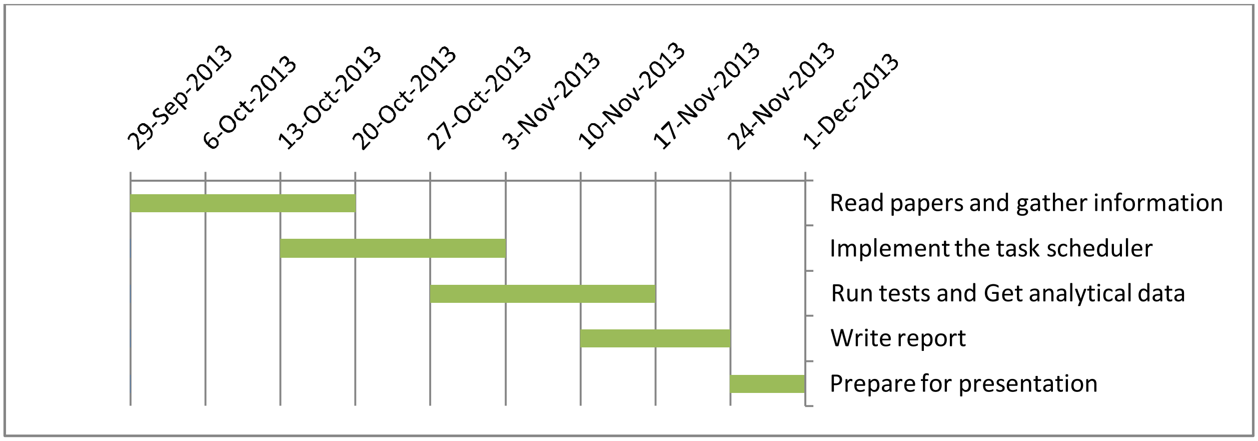
The goal of the project is to understand the challenges in job scheduling, scaling and fault-tolerance for a distributed system and come up with solution ideas and conclusion based on the analytical data. The overall time frame available and schedule will be a fairly significant factor in the level of detail ultimately achieved. Actual implementation of the load balancing or fault tolerance modules will be subject to the time available and is considered an optional/desirable goal.

Team:

This will be an individual project. The study, implementation and analysis as well as the report will be made by me alone.

Plan:

The following Gantt Chart shows the approximate plan for the above project proposal:



Note: The above figure is only a rough draft. As the project progresses from conception to realization, external factors such as examinations and other assignments might influence the schedule to a certain extent. However, constant reviews will be done to ensure that the project proceeds as planned and the goals set out in this document are achieved to a satisfactory extent.

References:

The following list of references represents an intended reading list. Whether or not a reference is ultimately relevant and in what capacity/extent, to this project will be ascertained during the information gathering process, and as such, all listed below may or may not be referenced. Also additional sources may be chanced upon during the research process.

[I] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. MIT Laboratory for Computer Science.

[II] H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris, I. Stoica. "Looking up data in P2P systems", Communications of the ACM, 46(2):43–48, 2003

[III] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, W. Vogels. “Dynamo: Amazon’s Highly Available Key-Value Store.” SIGOPS Operating Systems Review, 2007

[IV] Tonglin Li, Xiaobing Zhou, Kevin Brandstatter, Dongfang Zhao, Ke Wang, Anupam Rajendran, Zhao Zhang, Ioan Raicu. ZHT: A Light-weight Reliable Persistent Dynamic Scalable Zero-hop Distributed Hash Table in IEEE International Parallel & Distributed Processing Symposium, IEEE IPDPS ’13, 2013

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

[VI] J.M. Wozniak, B. Jacobs, R. Latham, S. Lang, S.W. Son, and R. Ross. “C-MPI: A DHT implementation for grid and HPC environments”, Preprint ANL/MCS-P1746-0410, 2010

[VII] Iman Sadooghi, Ioan Raicu. "CloudKon: a Cloud enabled Distributed tasK executiON framework"

[VIII] Fabio V. Hecht, Thomas Bocek, Burkhard Stiller. "B-Tracker: Improving Load Balancing and Efficiency in Distributed P2P Trackers", IEEE P2P 2011 proceedings

[IX] Che-Wei Chang and Hung-Chang Hsiao. "Stochastic Load Rebalancing in Distributed Hash Tables",2011 IEEE 17th International Conference on Parallel and Distributed Systems, 2012 IEEE 18th International Conference on Parallel and Distributed Systems

[X] http://www.hazelcast.com/whatishazelcast.jsp

[XI] Sun Grid Engine Tutorial: http://www.cbi.utsa.edu/book/export/html/29