Reading Assignment 3

Summary

Falkon was designed and implemented to enable the rapid and efficient execution of many independent jobs on large compute clusters. According to the previous paper author mentions the 3 techiniques used by falkon (1) multi-level scheduling techniques to enable separate treatments of resource provisioning and the dispatch of user tasks to those resources; (2) a streamlined task dispatcher able to achieve order-of-magnitude higher task dispatch rates than conventional schedulers; and (3) performs data caching and uses a data-aware scheduler to leverage the co-located computational and storage resources to minimize the use of shared storage infrastructure. The current paper is an extension of the previous paper so that loosely coupled programming on petascale can be brought to a practical use. Since the current trend is data intensive rather than hardware specific

this work enables the execution of highly parallel computations although the jobs would be serial maintaining the loosely coupled status thus no modifications being made to the original applications. Thus according to the author this approach helps us to use the full potential of petascale systems. For example IBM blue gene p which has 65,536 nodes . Thus they can be efficiently utilized for the petascale application by using the respective approach. The author has tested the following approach using 2 methods one being microbenchmarks and the second being testing on real applications from two domains viz. energy modeling and molecular dynamics. The results mentioned in the test promised a scale upto 160k with high efficiency and the execution of 1000 tasks per second. The Author mentions about MTC(Many-Task-Computing) which consists of both dependent and independent task that can be scheduled on many different computing resources. MTC denotes high performance computations comprising of multiple distinct activities cupled via file system operations. There are certain factors mentioned in this paper as to why we can use the MTC applications on petascale HPC systems, they are i) The i/o subsystems ii) The cost to manage and run is less as compared to conventional systems iii) Solves the utilization issues. iv) One of the only scale capable of providing enough compute power for some applications.

How is this work different from the related work ?

1. One of the research in which existing HTC mode and cobalt was used to make cobalt scheduling system rather than using falcon which itself was developed from. According to author the performance analysis the falcon gave at-least one magnitude better performance.
2. The other HTC systems focuses on distributed computing for execution. The HTC systems such as Hadoop, map reduce, condor targets processing over hundreds of compute nodes. But this paper targets the processing of the highly multi task applications on single but highly parallel super computers. These super computers lacked the gateway for execution of such applications previously. This paper tries to answer this issue.
3. After the evaluation of the performance of the falcon and loose programming model developed over it, the author was able to prove that the multi task applications can efficiently run on the peta scale systems.
4. The local resources management on BlueGene super computer assigns the task to psets that is 64 quad core compute nodes. Hence each task gets 256 processors. The loosely coupled applications has many single core tasks. So, the single core task has to take 256 cores, in which many cores stay idle.

5. Falkon is mostly focussed on performance and efficiency rather than robustness and recoverability. Thus it does not limit the efficiency of MTC applications as there is no overhead to make sure that the data can be recovered. Thus performance evaluation can be be done on falkon and no modifications are being done to the real applications.

Top 3 things the paper does well

1. The multilevel scheduling which uses the local resource management to acquire psets and then assign tasks at single core level granularity is very good approach.
2. Provides high efficiency and throughput
3. Static resource provisioning. The application directly requests the cobalt for number of processors and time required. Once the cobalt assigns the resource, falcon is bootstrapped and application directly communicates with falcon for core level of granularity.

3 Things paper could do better

1. The paper argues that peta scale systems have advantage over the grid clusters and tera scale systems, but the paper does not define exactly what application are targeted for this research apart from multi task applications. Also, the paper doesn't provide good comparison between tera scale and peta scale systems.
2. The research has extended the falcon developed for compute cluster, to develop the loosely coupled scripting programming model for petascale supercomputers. The author doesn't define challenges for this modification. – It defines
3. Falkon provides very high efficiency for a task having a task length of 256 seconds according to the microbenchmark performed. However, for real applications having such a task length is not feasible as the operations may take few days instead of hours.
4. Falkon has reliability issues at large scale. A machine such as Blue Gene /L has a MTBF of 10 days .Thus when running loosely coupled applications via Falkon, the failure of a single node affects the tasks being run on the failed node. Although the advantage is that it affects the tasks performed on the failed node and not the other node

*** if you were to be an author of a follow up paper to this paper, what extensions would you make to improve on this paper?**

Improving the reliability and robustness

Reducing the task length to a certain amount thus not compromising the efficiency and also the time taken for scheduling would be less