Master of Science in Informatics at Grenoble
Parallel,Distributed and Embedded System(PDES)

Multiparameter resource selection for next generation HPC platforms

Dineshkumar RAJAGOPAL

1st September 2015

Research project performed at BDS team in BULL-SAS

Under the supervision of:
Dr. Yiannis GEORGIOU
BULL-SAS, Echirolles

Defended before a jury composed of:
Prof. Arnaud LEGRAND
Prof. Martin HEUSSE
Prof. Noel DEPALMA
Prof. Olivier GRUBBER
Prof. Olivier RICHARD

September 2015

Abstract

RJMS[2](Resource and Job Management System) is a middleware system software in the supercomputer system software stack, is responsible for managing resources and selects the best resources to schedule the user′s job request. Its main operations are to manage queue, assign job′s priority, select next job and select best resources. Even though resource selection is one of the internal operation of the batch scheduler, Improper resource selection operation leads to improper resource management and increases the cost of HPC system maintenance, ownership and poor performance experience for users. Most of the RJMS resource selection policies follows best-fit, network topology aware and internal nodes resource consumption to increase the performance and throughput of HPC system.

This report concerns new perspective of resource selection in RJMS by energy efficient resource selection policy and LAYOUTS[1] based selection plugin implementation in SLURM[3](Simple Linux Utility for Resource Management) batch scheduler. LAYOUTS is a generic resource management framework to separate resource management from the resource selection plugin to keep the code maintainable. Managing resources information is dynamic, due to the evolution of cluster architecture and internal nodes. Separating resource management will give fine grain flexibility within the SLURM resource selection plugin to implement policy effectively. Due to the power wall problem and large number of nodes, HPC systems are moving from the performance oriented to the energy efficient. Energy efficient HPC sytem can be achieved by different methods and techniques supported at the level of hardware and software. RJMS leverages those features to manage the HPC system energy concerned. Here we proposed new best-fit energy efiicient resource selection policy to compare the energy efficiency with earlier non-energy efficient selection policy.

Experiment results of different resource selection policy and implementation follows the same behaviour at system level(system utilization and throughput) for the real system workload of ESP banchmark. Even though different selector can not distinguish at system level, Individual jobs resource selection time and waiting time for the LAYOUTS based implementation is higher than the earlier custom implementation. New LAYOUTS based implementation can adapt for different policy and criterias with the considerable overhead than current cons_res(consumable resource) resource selection plugin in SLURM. Energy efficient resource selection policy consumes less energy than common bes-fit performance oriented resource selction policy.

Contents

— 1 —
Introduction

High Performance Computer(HPC) is providing computing services to various users(Scientist, Data analyst) by using various technology(Cluster, Grid, Cloud). Who all are providing the computing services has to manage their own resources to satisfy the users computation requirement. SLURM is a RJMS, used in the most well known TOP500 HPC clusters(supercomputers) to manage the resources. RJMS is the connecting bridge between the users jobs and resources to distribute the computing resources(Nodes,etc) to the jobs of users effectively. RJMS knows the complete details about jobs of users and HPC resources, so it would be the perfect candidate to manage the cluster effectively as a manager. SLURM-RJMS overall system behaviour depends on the combination of scheduler and selector configuration. Selector(Resource selector) is an entity to select the best resources for the current job according to the selection policy. SLURM architecture follows plugin mechanism to support different cluster architecture, policy and users requirement. SLURM plugins maintain information of resources and implementation of policy together to lose maintanance and global view of the code. Resource management is dynamic and independent from the selection policy, so separating resource mangement from the plugin will resolve the above mentioned problems. LAYOUTS framework in SLURM developed to manage resources globally independent of the specific plugin, so new powerful select plugin can be implemented by using this framework. This report answers the following questions, how to implement a new select plugin using LAYOUTS and change the resource selection policy to support energy efficiency.

Software engineerings basic rule of thumb for flexibility is to manage logic and data separately like three-tier architecture in the client-server computing. Recent days RJMS follow this architecture to maintain data in the database. Even though the RDBMS data management supports fast implementation and greater flexibility, loses performance and reliability of the RJMS system. New LAYOUTS framework was developed to retain the good properties and remove the bad properties mentioned above. Layout manages resources information and relationship between resources hierarchically(tree), supporting internal aggregate functionalities(sum, avg, count,..) and access resources information by using LAYOUTS APIs.

Select plugin manages all the resources information of HPC and implementation of selection policy, so it would be the perfect candidate plugin to use LAYOUTS to separate data from the selection policy. Current select plugin has improper resource management(basic information and relationship between resources) to lose the maintenance and need to calculate information from the basic information of resources and relations to lose the global view of code. For example in SLURM, node and core relationship is well defined, but socket related information has to infer from the number of sockets in node and node -core relationship. LAY-

OUTS can maintain the resources information and relationship details in different levels. In this report we used the word layout frequently and having different meaning, to distinguish by using the following convention. The word LAYOUTS to mean framework and Layout(s) or layout(s) to mean the data management plugin. SLURM use parsable configuration file(text file) to keep the configuration and information of the system. LAYOUTS used the configuration file to keep the data information for different layouts plugin and access data of the layout by using APIs.

LAYOUTS APIs hide the internal architecture of layouts and access the current resources information easily. If the selection policy was changed then the new select plugin was implemented without changing the layout data management plugin and vice-versa also. Informal explanation of layout is to visualise the complete details of resource informations to assist plugin developers and cluster administrator.

Supercomputers number of cores and nodes are following the moore?s law steadily, so clusters energy consumption is increasing at the same rate. HPC researchers try to reduce the energy consumption by leveraging energy related features supported at the different level of hardware and software. RJMS placed between user?s job and resources, so it can leverage all the information to manage resources effectively. We proposed energy efficient bestfit resource selection policy and implemntation using LAYOUTS. The experiment results showing the new selection policy consuming less energy than the older approach.

The report was organized as follow. Chapter 2 concern the related works of others regarding RJMS implementation and resource selection policies. Chapter 3 explain the basic SLURM architecture and old resource selection algorithm and code level details for SLURM developers. Chapter 4 illustrates the new generic resource management LAYOUTS architecture, features and APIs. Chapter 5 explains the LAYOUTS based resource selection plugin implementation architecture and energy efficient resource selection algorithm and implementation. Chapter 6 analyse the experiment results and experiment procedure for the different plugin. Finally Chapter 7 conclude the report with important results and possible future work.

— 2 —
State of the art

RJMS(batch scheduler) core functionality is to schedule the resources upon jobs and the basic high level components and architecture is shown in the figure. Batch schedulers overall system property depends on the scheduler and selector combination(schedule cycle) as shown in the figure. Resource selector has to select the best resources to satisfy the jobs requirement, so it is important as the scheduler functionality. If the batch scheduler had perfectly architected and implemented, then the behaviour of the whole schedule cycle would have been well defined. Currently RJMS research is mainly focused on the different algorithms and limitations of scheduler functionalities. Scheduler and selector functionalities are naturally defined perfectly and tradeoff for making decision based on the criterias(user or server criteria).

This section was organised as follow. Section 2.1 will explain the different RJMS implementation and limitation. Section 2.2 explains the scheduling algorithms and limitation . Finally section 2.3 explains the resource selection algorithm and limitations.

2.1 RJMS implementation and limitation

Most of the RJMS follows the same high level architecture, but differ by implementation and supporting features. SLURM like RJMS was targeted for supercomputers to support scalability and performance, so the system was implemented to achieve scalability and performance. SLURM supports upto 64000 nodes and all the operations are performed very quckly. Because of the scalability and performance, SLURM was used in the most of the TOP500 supercomputers(eg.CEA NOVA, CURIE and Tihane). Any software can not achieve all the three properties of scalable, performance oriented and extensible without good architecture and implementation. SLURM used plugin mechanism to support extensibility, but extensbility within the plugin is questionable.

OAR was another RJMS developed by Inria to support scalable and flexible system by using high level programming tools(perl and MySql). Maui, Torque also follows similar implementation like OAR. Imporatant difference among the both implementation is data management. OAR uses RDBMS(mysql or postgresql) to mangae all the RJMS entities, But SLURM manages all the information using its own data structure. Implementation of resource selection operation was interesting to the following section. In OAR, they use sql query to get resources instantaneous availability to select the best resources. In SLURM, they use internal data structures raw information of resources to calculate the reuired information to select the best resources. Performance wise SLURM based custom implementation was good, Flexibility wise

Table 2.1: Different criterias can be supported in the Batch scheduler resource selection policies

Criteria type	Criterias
Server	Energy, Temparature
User	Performance

OAR based implementation was good, so it is a tradeoff to choose the best implementation method.

2.2 Resource selector

The report is concerned about resource selection, but active research works in resource selction is very less compared to scheduler. As shown in the figure, select operation is as important as schedule operation. If the selection operation is not good, then the behaviour of schedule cycle will be affected irrespective of scheduler operation.

Most of the resource selection operation in RJMS is best-fit single criteria(or objective) only. HPC was earlier focused on performance, so the RJMS management also based on performance. HPC applications performance is not only based on computation, depends on communication also. Network topology aware resource selection will reduce communication latency and improve performance, if the selected nodes are connected in a single leaf switch. Best-fit(most-fit) resorce selection is same as best-fit dynamic heap memory allocation. Best-fit policy select resource having minimum satisfiable resources than maximum satisfiable resources. This paper[**?**] mention different resource selection policies and its limitations. In the same paper[**?**] they explained the resource selection policies different criterias and properties in more detail. Theoretically Resource selection policy has to avoid the following properties overall.

- **Resource contention** happened, when a resource was shared between two jobs then the jobs competate for the shared resource. It affects the performance of the Jobs.
- **Resource scarcity** happened, when the scarce resources are exploited by the jobs. It increases the job waiting time untill the resources available.
- **Resource fragmentation** happened, when the improper resource selection policy was supported and the order of jobs in the workload. It reduces system utilization of the system.
- **Over provisioning** happened, when the improper resource selection policy was supported. It affects system utilization of the system.
- **Under provisioning** happened, when the improper resource selection policy was supported. It affects users application performance.

Resource selection opearation was not only based on the selection policies, it depends on the selction criterias also. criteria for resource selction depends on the user criterias or server criterias, this is mentioned in the table 2.1. In the chapter 5 we explains best-fit performance

criteria based resource selection policy. In the chapter 5 we point out the changes of best-fit resource selection policy to support multiple criterias.

2.3 Job scheduler

Batch scheduler perform scheduling is based on the combination of Resource selector and Job scheduler, and it is called schedule cycle and shown in the figure. Job sceduler perform the Job queue mangement, job priority assignment and select next jobs. SLURM using queue to manage jobs, but some of the RJMS not using queue and it is based on planning, this was explained in the paper []. Select next job is the simple to pick head of the queue, but the job in head is based on the jobs priority. Job priority assignment based on QOS, Job scheduler policy. In simple Job scheduler algorithm is to assign high priority for earlierly arrived jobs and keep the first arrived job front of the queue. Different Job scheduler algorithm is summarised in the paper[].

— 3 —
SLURM Architecture

SLURM is the famous resource manger in the HPC cluster world, because of its scalability and performance. SLURM was implemented by using **C** programming language and **autoconf** tools. **C** is the basic high level programming language, so SLURM developers developed basic data structure, parsing configuration files, remote method invocation(RMI) message format, wrapping and unwrapping, logging and mechanism for different plugins in src/common directory of SLURM project. The implementation, code convention and naming were from the SLURM version 15.08, it may be changeable in the future version. SLURM architecture and deamon program in the different level was shown in the figure . Slurmctld is the controller deamon running in the SLURM server to arbitrate whole resources of cluster. Slurmd is the computing deamon run in the computing nodes of cluster to control the jobs execution and monitoring nodes state. Slurmdbd is the deamon accounting information in the local or remothe mysql database server. Resource selector and job scheduler were implemented in the controller deamon, so all the new resource selection plugin were implemented in the slurmctld. However the new plugin implemented went wrong will not affect other functionalities of the system to isolate one plugin from whole system.

SLURM has its own command for users and cluster administrators to submit jobs and manage resources. Every command and it is purpose is given in the table. More details about the command can find in the SLURM manual. SLURM was used by different vendors and clusters, so they can use existing plugins of SLURM or they can develop their own plugins to leverage their own cluster features. Currently SLURM supports IBM bluegeen, LLNL Cray and Alps, and Bullx clusters. Bullx is an intel cluster and it is very commonly used cluster type. All the implemntation mentioned .Plugin mechanism adds flexibility to the system easily, but loses maintanance within the plugin. Resource management within the plugin was the reason for inflexibility. LAYOUTS framework can manage resources globally to separate resource management from the plugin is possible.

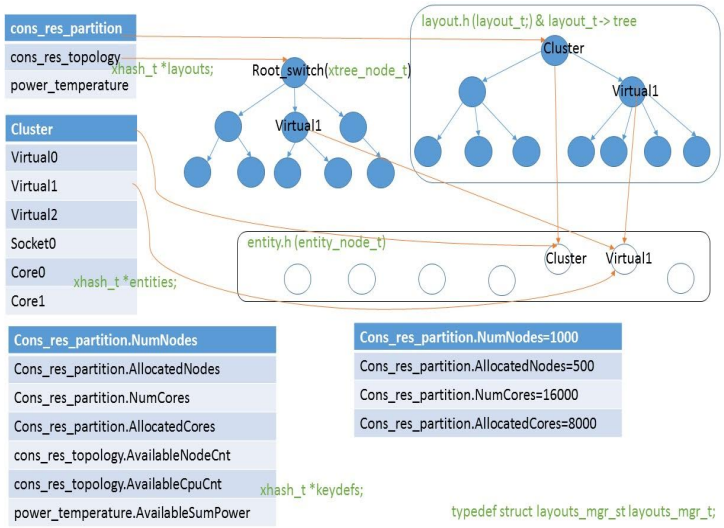

Figure 3.1: Figure example

Table 3.1: SLURM entities for resource mangement

Entities	Purpose
Nodes	This is the physical computing nodes of HPC to manage by RJMS
Partitions	This is the logical entity to group nodes based on the features of Hardware and software

— 4 —

LAYOUTS Framework

— 5 —

Resource Selection Improvement

— 6 —

Experimentation and Performance Evaluation

— 7 —
Conclusion

Bibliography

[1] Francois Chevallier. Design of a generic framework for resource management in slurm. Master's thesis, ISTIA,University of Angers, September 2013.

[2] Yiannis Georgiou. *CONTRIBUTIONS FOR RESOURCE AND JOB MANAGEMENT IN HIGH PERFORMANCE COMPUTING*. PhD thesis, University of Grenoble, HAL open archive, Novembre 2010.

[3] Morris A. Jette, Andy B. Yoo, and Mark Grondona. Slurm: Simple linux utility for resource management. In *In Lecture Notes in Computer Science: Proceedings of Job Scheduling Strategies for Parallel Processing (JSSPP) 2003*, pages 44–60. Springer-Verlag, 2002.