

IndoGRID : a gateway for GRID based public computation in Indonesia

Z. Akbar, I. Firmansyah, A.A. Nataatmaja and L.T. Handoko*

Abstract

This short paper is intended as a final report for the above research project funded partially by the Indonesia Toray Science Foundation (ITSF) Research Grant 2006. The completion of enabling the LIPI Public Cluster to be accessible for public is reported. The whole architecture, the microcontroller-based control and monitoring system and the advanced web interface integrating all aspects are briefly explained. Finally, future prospect and plan of further development are commented.

1 Introduction

LIPI Public Cluster (LPC) is a cluster-based computing facility maintained by Lembaga Ilmu Pengetahuan Indonesia - LIPI (the Indonesian Institute of Sciences) [1]. Although it is still a small scale cluster in the sense of number of nodes already installed, it has unique characteristics among existing clusters around the globe due to its openness. Here "open" means everyone can access and use it anonymously for free to execute any types of parallel programmings [2].

The development of LPC was initially motivated by real needs for high performance and advanced computing, especially in the research field of basic natural sciences. Even in Indonesia, the needs are growing along with the advances of scientific researches. In the last decades, clustering low specs (and low cost) machines becomes popular to realize an advanced computing system comparable to, or in most cases better than the conventional mainframe-based system with significant cost reduction [3].

In general a cluster is designed to perform a single (huge) computational task at certain period. This makes the cluster system is usually exclusive and not at the level of appropriate cost for most potential users, neither young beginners nor small research groups, especially in developing countries like Indonesia. It is clear that the cluster is in that sense still costly, although there are certain needs to perform such advanced computings. No need to say about educating young generations to be the future users familiar with parallel programmings. This background motivates us to further develop an open and free cluster environment for public [4].

According to its nature, LPC is in contrast with any conventional clusters, designed to accommodate multiple users with their own parallel programmings executed independently at the same period. Therefore an issue on resource allocation is crucial, not only in the sense of allocating hardwares to the appropriate users but also to prevent any

*Group for Theoretical and Computational Physics, Research Center for Physics, Indonesian Institute of Sciences (<http://teori.fisika.lipi.go.id>).

interferences among them. In LPC we have deployed a new algorithm to overcome this problem, namely the dependent [5] and independent multi-block approaches [6].

Concerning its main objective as a training field to learn parallel programmings, the public cluster should be accessible and user-friendly for all users with various level of knowledges on parallel programming. It also should have enough flexibility regarding various ways of accessing the system in any platforms as well. This can be achieved by deploying a web-based interface integrating all aspects. Presently we have resolved some issues, such as security from anonymous users to prevent any kinds of interference among different tasks running simultaneously on multi blocks [5, 6], algorithm for resource allocation management [7] and the real-time monitoring and control over web for both administrators and end-users [8].

In this paper we first present briefly the concept of LPC including the work flow from the initial registration to the execution of computational works. Thereafter we discuss the whole architecture and its supporting systems. Finally we conclude with some comments and discussion.

2 The concept and work flow

In order to overcome the issues mentioned in the preceeding section, it is clear that we should develop and utilize an integrated system in LPC. However, concerning real demands in Indonesia presently, we assume that our potential users would be beginners in parallel programmings with educational or self-learning purposes. Although the cluster is of course going to be used by some experts and larger research groups to perform more serious advanced computings, we do not expect any anonymous users with heavy computational works. The reason is mainly the limited resources, i.e. number of nodes and its specifications. Actually we rather prefer as many as people use our cluster as a training field to learn parallel programmings. In average we plan to provide only 2 ~ 4 nodes in a relatively short period, namely less than 3 days, for each anonymous user.

These characteristics completely differ with another existing clusters which are usually used by certain people or groups bringing similar types of computational works appropriate for the cluster after particular procedures like submitting proposals and any kinds of letter of agreements. In our case, incoming users are completely anonymous and then no way in any means to know the type of computational works being executed. This fact consequently brings another problems as maintaining job executions owned by different users at same period simultaneously. Under these assumptions and conditions, the public cluster should fulfill very challenging requirements that might be irrelevant in any conventional clusters, that is :

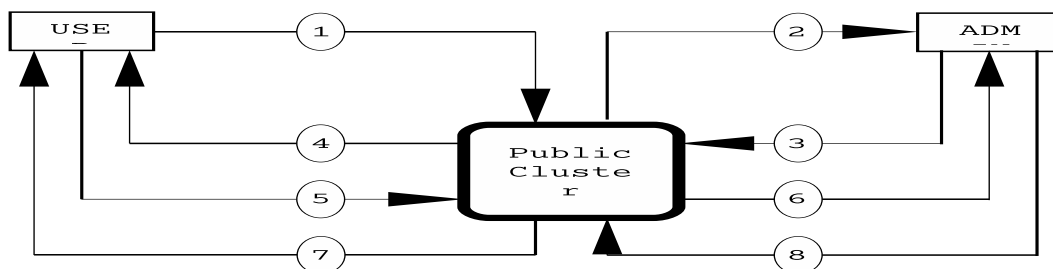


Figure 1: The interaction flow of administrator and users at the LPC.

- Security :
This is the main issue we should encounter from the first. The users should have enough privileges and freedom to optimize their opportunities to learn parallel programming, while their access must be limited at the maximum level for the sake of security.
- Flexibility :
It is impossible to provide the same level of flexibility for anonymous users as well-defined users with direct accesses through `ssh`, but we should allow as much as possible the users to execute their codes on LPC. Also there should be a freedom on assigning the number of nodes for each user, since each user could require various number of nodes depending on the actual computational capacities needed.
- Stability :
Simultaneous executions by different users with various programmes in different blocks of cluster without any interferences among them requires new innovations on cluster management. It includes some techniques to incorporate wide range of nodes with different specifications, that is actually ranging from the old Intel 486 to the latest Athlon based nodes.
- Efficiency :
Since the cluster is dynamically divided into several blocks with various number of nodes inside according to the users requests, each node should be able to be completely turned on or off partially without any interruptions to another working nodes.

All of these require modifications on some existing tools, and also new developments on alternative softwares and hardwares as well [4, 5, 6, 7, 8].

Regarding to the above-mentioned concept, we have defined the work flow in LPC as depicted in Fig. 1, that is,

1. A new user should complete an initial registration by providing personal data, the content of job will be performed and the number of nodes requested for the job.
2. The application is reviewed and verified by the administrator. After approval, the administrator will assign the nodes provided and its usage period.
3. After reconfirmations by user, proving their agreements with the provided nodes and usage period, the administrator will switch the nodes on that also activate all daemons automatically.
4. The users should adjust their parallel programmes to fit the provided nodes.
5. The user uploads all necessary programmes and libraries if any. At this stage the programmes can be executed immediately.
6. The administrator and automated system will monitor the usage of all running users.
7. Finished jobs and the results can be downloaded by owners.
8. Once the usage period is over, the nodes are turned off automatically.

Again, these procedures can be done fully and remotely through web. We should mention that the types and number of nodes allocated for a newly assigned block is done automatically utilizing a decision making tool based on the extended genetic algorithm embedded in the web-interface [7]. Now we are ready to discuss the architecture of web-interface in LPC.

3 The architecture

The integrated supporting system consists of microcontroller-based control and monitoring, software-based middlewares and web-based interface. All of them should take over all aspects in LPC, from communication among the nodes till running related tools and softwares to perform computational works. The web interface is the only way connecting the internal system with the rest of the world.

In term of communication, there are two kinds of communication in LPC : a) between users and gateway server through HTTP protocol, and b) between gateway server and nodes consisting blocks of cluster through `ssh` or `scp`. This is to limit direct access to a block of particular cluster for security reason. On the other hand, this still allows the system to interact with another elements of cluster as IO server and master nodes of each block. As depicted in Fig. 2, we put a gateway server as a common entrance for users which serves all related web-based applications. The gateway server further sends and receives commands related to file or parallel programming operations using secure remote login `ssh` / `scp`.

In order to reduce the overall cost, we have developed a new architecture in LPC, the so called the multi-block approach [5, 6], to enable multiple blocks of small cluster running simultaneously using single master as illustrated in the left figure of Fig. 3. The approach is quite powerful for processor intensive works that are the typical works at LPC.

On the other hand, the web application in LPC should accommodate the needs of any parallel programmings which vary and change along the time. Then we must make it as flexible as possible to guarantee its compatibilities in the future. Also, any future modifications in the web application must not alter another software components. This can be achieved by adopting three-tier software architecture [9]. This method is appropriate for distributed client / server applications which require high performances in terms of flexibility, maintainability, reusability and scalability, while at the same time it simplifies the complexities of users' distributed processes. This architecture is shown in the right figure of Fig. 3. As can be seen in the figure, each layer is completely separated which could then simplifies its developments and implementations. Presentation layer behaves as a front interface and interacts directly by receiving inputs from users. The HTML files

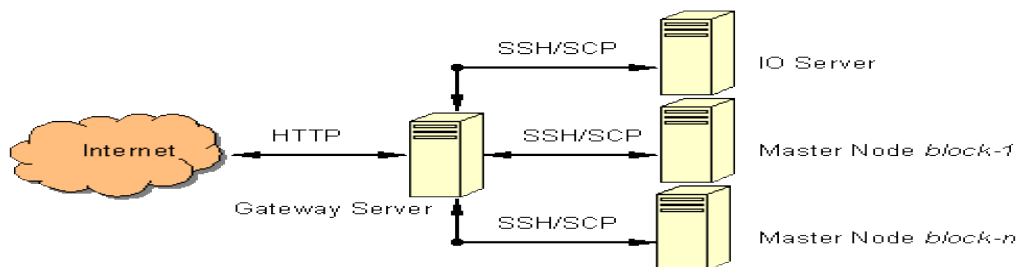


Figure 2: The communication architecture in LPC.

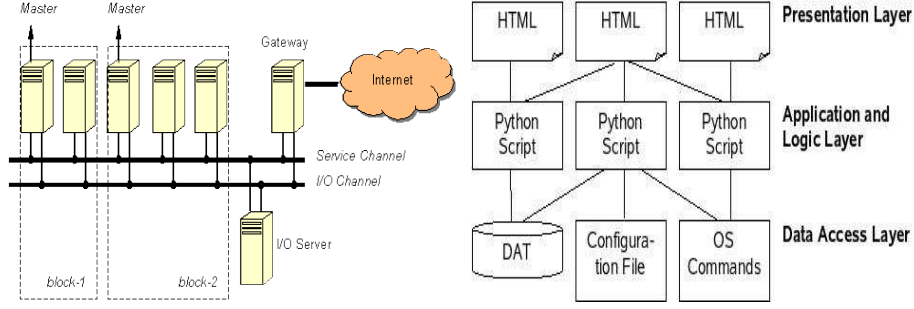


Figure 3: The architecture of multi-block approach (left) and three-tier web-based interface (right) in LPC.

in this layer could contain either static or dynamic (through application and logic layer) contents. For instance in the case of LPC they are generated dynamically by Python scripts. The application and logic layer handles all programming logics to serve user needs as uploading the jobs, executing codes and so forth. Further, the data access layer provides the access to data files from the database system, configuration files or generated by any commands in operating system. Moreover, the main task for web-interface in LPC is providing the users an appropriate environment for parallel programming. At time being, there are several parallel programming environments like MPI, PVM and so on [10]. Also some modified versions of them as MPICH (version 1 and 2) and LAM-MPI. The web-interface should allow users to choose the preferred one, or to change from one to another. In order to enable dynamic change of environment according to user preferences, we have deployed a mechanism based on the Modules package [11]. This package enables environment switching by changing its path and variables through a single command.

Further issue is how to keep the whole performance of cluster during operation. This can be done by embedding "real-time" control and monitoring system and displaying the results through web. We have developed a dedicated device for this purpose using microcontroller-based hardware as shown in Fig. 4 [8]. The control system manages the hardware aspect of each node, for instance turning them on or off. The microcontrollers are interfaced by Python and some external programmes in C++. The monitoring system takes the data of external physical observables such casing temperatures and humidities. The physical parameters for each node can be retrieved easily through BIOS. Both control and monitoring systems are integrated to realize an automated hardware control, i.e. shutting down a node if the temperature exceeds the predefined threshold, etc. Beside these hardware-based management systems, we have also implemented the software-based management system to control all nodes more efficiently. We have deployed a tool set

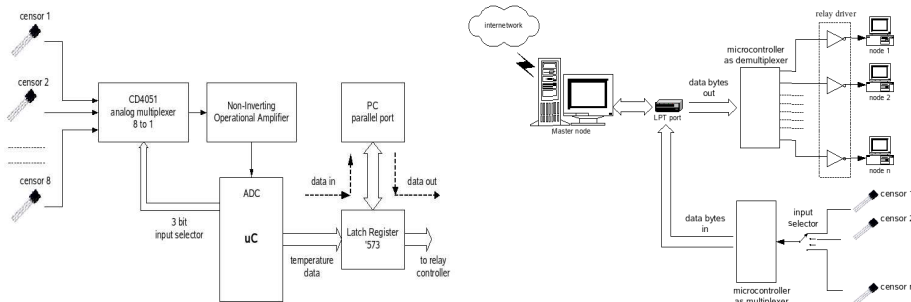


Figure 4: The block diagrams for monitoring (left) and control (right) system.

called Cluster Command and Control (C3) [12] and Ganglia [13]. C3 bundles useful commands to enable automatic processes throughout multiple nodes, while Ganglia is used to monitor the current condition of cluster through web-interface. We have integrated these tools in our web-interface to create a more comprehensive web-based interface matches specific needs in LPC.

For resource management, we have implemented our own resource allocation algorithm based on the extended genetic algorithm embedded in the web-interface [7]. This is different with the conventional cluster where the resource allocation should be performed during the computational period to achieve the best performance. In the public cluster like LPC, the conventional resource allocation is irrelevant, because it is done only at the first time of node assignment after the approval procedure. Once a block consisting of some resources has been allocated, it remains unchanged during the usage period. Of course, resource allocation within a block is still relevant, but it uses the existing tools chosen by users. Moreover, doing an appropriate load-balancing within a block used by user is an important aspect of learning parallel programming, so this task should be done by the users themselves.

4 Summary and discussion

We have introduced the LPC which has been running fully and is available publicly for free. Motivated by its uniqueness, the system contains a lot of innovations and integrates both aspects of hardware and software. The architecture is quite unique globally and requires careful developments.

The LPC so far enjoys numerous visits and many active users as can be seen through the web [1], ranging from high school to college students. This fact has definitely proven the usability of the system by (mostly) beginners in parallel programmings. The overall performance, especially our multi-block system, is so far quite good as expected. Several performance analysis are also available on the web.

We would like to comment some points regarding the current status of LPC :

- In some specific cases, the work-flow in LPC can be automated to reduce administration works. For instance, it could be realised for a “public” cluster limited to a community with predefined users and types of computational works. However, in our case the present work flow is the simplest and most “automated” one regarding the security and its level of openness.
- Related to this, the main issue in daily operation is to prevent in advanced malicious users who try to submit huge never-ending computational tasks. Actually, this is the main reason keeping current work-flow which is quite powerfull to deal with this potential problem.
- The current system is in principle portable to other existing clusters. Especially the concept could be an interesting alternative for some old clusters that might still be usefull for public education purposes.

Now, a regular paper [16] has been submitted to the related journal, and a patent draft [15] registered to the Indonesian Patent Office is still in progress.

As future work, we are going to investigate more comprehensively the overall performances, especially in the case of simultaneous heavy computational jobs of multiple

users. However, the current performances have been reported partially in our previous works [5, 6, 7]. Also following recent progress in high performance computing, we are in the step of connecting some blocks in LPC into grid computing under global grid collaboration. Actually this is the main target in this project which we have not accomplished so far due to another reason. As written in the proposal, we plan to participate to the Global Grid collaboration under AsiaLink Programme with European counterpart, however we have not yet fulfilled a requirement of 3 doctoral students involved in the project. At time being, we have only 2 doctoral students in this project.

Acknowledgement

This work is financially supported by the Riset Kompetitif LIPI in fiscal year 2007 under Contract no. 11.04/SK/KPPI/II/2007 and the Indonesia Toray Science Foundation Research Grant 2007.

References

- [1] LIPI Public Cluster, <http://www.cluster.lipi.go.id>.
- [2] L.T. Handoko : "Public Cluster : mesin paralel terbuka berbasis web", Indonesian Copyright, No. B 268487 (2006).
- [3] A. Goscinski, M. Hobbs and J. Silcock : "A Cluster Operating System Supporting Parallel Computing", Cluster Computing, Vol. 4, pp. 145156 (2001).
- [4] Z. Akbar, Slamet, B.I. Ajinagoro, G.J. Ohara, I. Firmansyah, B. Hermanto and L.T. Handoko : "Open and Free Cluster for Public", Proceeding of the International Conference on Rural Information and Communication Technology 2007, 2007.
- [5] Z. Akbar, Slamet, B.I. Ajinagoro, G.J. Ohara, I. Firmansyah, B. Hermanto and L.T. Handoko : "Public Cluster : parallel machine with multi-block approach", Proceeding of the International Conference on Electrical Engineering and Informatics, 2007.
- [6] Z. Akbar and L.T. Handoko : "Multi and Independent Block Approach in Public Cluster", Proceeding of the 3rd Information and Communication Technology Seminar, 2007.
- [7] Z. Akbar and L.T. Handoko : "Resource Allocation in Public Cluster with Extended Optimization Algorithm", Proceeding of the International Conference on Instrumentation, Communication, and Information Technology, 2007.
- [8] I. Firmansyah, B. Hermanto, Slamet, Hadiyanto and L.T. Handoko : "Real-time control and monitoring system for LIPI Public Cluster", Proceeding of the International Conference on Instrumentation, Communication, and Information Technology, 2007.
- [9] J. Foreman, J. Gross, R. Rosenstein, D. Fisher, K. Brune : "C4 Software Technology Reference Guide - A Prototype", Handbook (CMU/SEI-97-HB-001), Software Engineering Institute, Carnegie Mellon University, 1997.

- [10] For examples, see :
 Message Passing Interface Forum : "MPI2: A Message Passing Interface standard".
 International Journal of High Performance Computing Applications, 12(12):1299,
 1998.
- [11] John L. Furlani : "Modules: Providing a Flexible User Environment", Proceedings
 of the Fifth Large Installation Systems Administration Conference (LISA V), pp. 141-
 152, San Diego, CA, September 30 - October 3, 1991. <http://modules.sourceforge.net>.
- [12] Ray Flanery, Al Geist, Brian Luehke and Stephen L. Scott : "Cluster
 Command & Control (C3) Tool Suite", 3rd Austrian-Hungarian Workshop on
 Distributed and Parallel Systems (DAPSYS 2000) in conjunction with Eu-
 roPVM/MPI 2000, Balatonfured, Lake Balaton, Hungary, September 10-13, 2000.
<http://www.csm.ornl.gov/torc/C3/>.
- [13] Ganglia Monitoring System, <http://ganglia.sourceforge.net>.
- [14] Z. Akbar and L.T. Handoko : "Web-based Interface in Public Cluster", Proceed-
 ing of the 9th International Conference on Information Integration and Web-based
 Applications and Services, 2007.
- [15] Z. Akbar, Slamet, B.I. Ajinagoro, G.J. Ohara, I. Firmansyah, B. Hermanto and
 L.T. Handoko : "Sistem komputasi paralel multi blok dengan master node tunggal",
 submitted as Indonesian Patent, 2008.
- [16] Z. Akbar, Slamet, B.I. Ajinagoro, G.J. Ohara, I. Firmansyah, B. Hermanto and L.T.
 Handoko : "LIPI Public Cluster : an architecture to enable public access to cluster",
 submitted to the Journal of Parallel Processing, 2007.