

An Efficient Run Time Interface for Heterogeneous Architecture of Large Scale Supercomputing System

Prabu D., Andrew Aaron James, Vanamala V., Vineeth Simon, Sanjeeb Kumar Deka, Sridharan R., Prahlada Rao B.B., and Mohanram N.

Abstract—In this paper we propose a novel Run Time Interface (RTI) technique to provide an efficient environment for MPI jobs on the heterogeneous architecture of PARAM Padma. It suggests an innovative, unified framework for the job management interface system in parallel and distributed computing. This approach employs proxy scheme. The implementation shows that the proposed RTI is highly scalable and stable. Moreover RTI provides the storage access for the MPI jobs in various operating system platforms and improve the data access performance through high performance C-DAC Parallel File System (C-PFS). The performance of the RTI is evaluated by using the standard HPC benchmark suites and the simulation results show that the proposed RTI gives good performance on large scale supercomputing system.

Keywords—RTI, C-MPI, C-PFS, Scheduler Interface.

I. INTRODUCTION

LARGE scale parallel machines, such as PARAM Padma[3] are expected to play a prominent role in taking on the demands of long-running scientific applications. C-DAC's PARAM [1] series of super-computers are large clusters of high performance workstations interconnected through low-latency, high bandwidth communication networks. C-DAC's Tera Scale Supercomputing Facility (CTSF)[2] houses the PARAM Padma - the powerful supercomputer in India. The supercomputer nodes are connected through a primary high performance System. Area Network, PARAMNet-II [4], designed and developed by C-DAC and a Gigabit Ethernet as a secondary network The Storage System of PARAM Padma has been designed to provide a primary storage of 5 Terabytes scalable to 22 Terabytes. The network centric storage architecture, based on Storage Area Network (SAN) technologies, ensures high performance, scalable and reliable storage. File Servers are 4-way SMPs based on UltraSparc-IVprocessors operating at 900MHz with aggregate primary memory of 96GB. The operating system on compute nodes is version AIX5.1L and File Servers are having Solaris as their operating system. Load Leveler [5] is used for resource management. Load Leveler only supports parallel jobs (MPI)[6] spawned through POE[7]. In our research, we have designed highly scalable RTI to support the long-running scientific applications for heterogeneous cluster architecture system.

Manuscript received October 15, 2006

Authors are with Systems Software Development Group, Center for Development of Advanced Computing, Knowledge Park , 1 Old Madras Road, Byappanahalli, Bangalore-560038, India (e-mail: {prabud, ron, vanamala, vineeth sanjeebd, rsridharan, prahladab, mohan}@cdacb.ernet.in).

Rest of this paper is organized as follows. Section 2 a brief overview of the related work. Sections 3 discuss C-DAC's Parallel File System (C-PFS). In section 4 we provide reasons for the need of C-DAC's RTI. Section 5 shows Architecture and working of RTI. Sections 6 discuss RTI Testing Environments and experimental evaluations Finally Section 7 discuss the conclusion and future work.

II. RELATED WORK

C-DAC's Resource Management Software (RMS)[8] manages, monitors and analyzes the workload on the nodes in the cluster and unites the nodes in the cluster for efficient execution and management of programs. RMS supports sequential and parallel (MPI) applications. RMS improves the performance by scheduling the jobs on nodes depending upon their load. It queues the jobs and schedules them based on the availability of resources. Key features of RMS are:

- Remote job submission
- Supports the heterogeneous environment
- Job queuing

Load scheduling software (LSF)[9] is suite of job and workload management products from Platform Computing Corporation with the following Key Features

- Supports Interactive and Batch System.
- Job scheduler and analyzing tool for workload of cluster
- Supports heterogeneous clusters
- Check-pointing and job migration

Also a set of functional API of Lsf is available that allows user or administrator to tap into or extend the functionality. Almost all the functionality of LSF is available through GUI or through commands. C-DAC's Cluster Runtime Environment (C-CRE) comes as a part of C-DAC HPC ClusterTools[10]. C-CRE manages the submission and execution of both serial and parallel jobs on the cluster nodes, which are grouped into logical sets called partitions. Key Features of CRE are: A single job-monitoring and control point

- Load-balancing for shared partitions
- Information about node connectivity

Load Leveler is a batch job scheduling application from IBM. It provides the facility for building, submitting and processing batch jobs within a network of machines. Key Features of load leveler are:

- Supports both serial and parallel jobs.
- Supports PVM, MPI and OpenMP
- Supports Check pointing and Restart

The Portable Batch System [11], a flexible batch queuing and workload management system originally developed for NASA, operates on networked, multi-platform UNIX environments, including heterogeneous clusters of workstations, supercomputers, and massively parallel systems. Key Features of PBS are

- **Portability:** complies with the POSIX 1003.2d
- **Configurability:** easy to configure to match the requirements of individual sites.
- **Usability:** also provides a graphical user interface (GUI).

III. C-PFS: C-DAC PARALLEL FILE SYSTEMS

C-DAC High Performance Computing and Communication (HPCC) software effectively addresses the performance and usability challenges of clusters through a high performance flexible software environment. C-DAC Parallel File System (C-PFS), part of the HPCC Software available on PARAM Padma, is a client-server and user-level parallel file system, addresses the high I/O throughput requirements of scientific and engineering applications. Portions of C-PFS[12] functions have been derived from the SunClusterTools made available through the Sun Community Source Licensing (SCSL)[13] program. The proposed RTI has been integrated with C-PFS in order to provide storage support for the MPI Jobs.

IV. MOTIVATION OF RTI ON PARAM PADMA

PARAM Padma Cluster of workstations are designed to provide massive computational power to users at low cost. These are low cost and readily available alternatives to specialized High Performance Computing platforms. Random submission of jobs on clusters can cause some workstations to be heavily loaded while other workstations are idle or lightly loaded and thus contradicts the very purpose of clusters of workstations for parallel processing. The major challenge for system administrators is to allocate the processing capacity available in the locally distributed system to facilitate its maximum usage. In CTSF Job Management System (JMS) we had two options - C-DAC RMS or IBM LoadLeveler. As LoadLeveler supports checkpointing and re-start features, we used LoadLeveler as JMS. Hence we came up with the idea of RTI - the Resource Scheduler Interface Software proposed by C-DAC, which will enable Load Leveler to launch and execute C-MPI [14] (CDAC Message Passing Interface) jobs and also support C-PFS. Since PARAM Padma already has LoadLeveler installed on it, the scientific and engineering community can run parallel application using C-MPI. Also RTI is integrated with C-CRE (CDAC Cluster Runtime environment) on Solaris Storage Cluster, which will benefit them to utilise optimised implementation of MPI-IO interface provided with the C-PFS and thereby achieve enhanced performance. RTI integrated with LoadLeveler and C-MPI will function as a fully integrated platform for running MPI applications. The scope of the plug-in software is limited to the interface between the LoadLeveler and C-RTE on PARAM Padma and the interface between LoadLeveler and C-CRE on Solaris Storage Cluster.

V. ARCHITECTURE AND WORKING OF RTI

The proposed architecture of RTI to integrate with IBM load leveler[5] and Parallel Operating Environment(POE). The POE Manager is a Background process which is started when we submit the interactive POE job. POE is responsible for performing many tasks in the Parallel Environment as follows [7]:

- Gets the nodes for the submitted parallel job through correspondence with the Job Manager.
- Starts a process manager daemon background process in all the nodes of the configurations.

Directs stdout, stdin and stderr on the "home node" to all other nodes in the configuration. When LL detects a condition that it should kill the parallel job, a SIGTERM signal will be sent to POE jobs. LoadLeveler will ultimately terminate the task. CRE on the other hand is tightly coupled with C-MPI. C-MPI uses RTE(Run Time Environment) as the interface to contact CRE as shown in the Fig. below. Our objective is to implement a plug-in software, CDAC's RTI, enables LoadLeveler to contact RTE and in turn C-MPI to launch and execute the parallel jobs using C MPI as shown in Fig. 1.

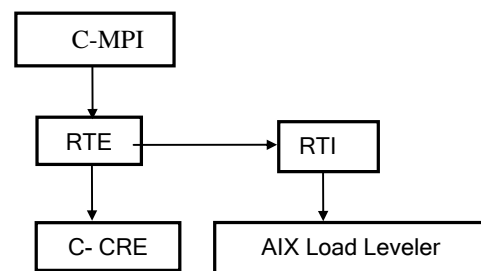


Fig. 1 CMS provides environment for CMPI jobs on AIX Environment

RTI software acts as an interface between the IBM Load Leveler and C-MPI. In this way it is providing support to the supercomputer users to use C-MPI along with the LoadLeveler. RTI acts as an interface between PARAM Padma cluster running on AIX operating system(AIX 5.1L) and storage clusters running on Solaris operating system for supporting MPI-I/O calls. RTI provides support to the PARAM Padma users with C-DAC C-PFS. RTI interacts with the IBM LoadLeveler on AIX cluster and with RTE on the storage cluster. Interaction between the plug in software and LoadLeveler is through LoadLeveler's APIs, RTI and RTE is through Remote Procedure Calls(RPCs). RTE queries information from the RTI, it in turn queries the same from LoadLeveler through LoadLeveler APIs, or from the C-PFS through the RPCs. LoadLeveler or C-PFS processes the request and sends relevant information back to RTE through RTI. The data flow between RTI, RTE, LL, C-PFS and users are explained in next section.

A. Working of RTI

The working of RTI is explained in steps 1 to 19 with reference to Fig. 2. These steps are aligned with the dataflow lines shown in Fig. 2 across different modules of RTI with these flow lines are also labelled 1 to 19 to have one to one correspondence between the explained steps and data flow.

1. User launching MPI applications through loadleveler.
2. Loadleveler spawns MPI jobs on requested number of nodes.
3. MPI jobs request RTE to get some information like port number of all the participating processes, job table etc.
4. RTE in turn contacts RTI for the information.

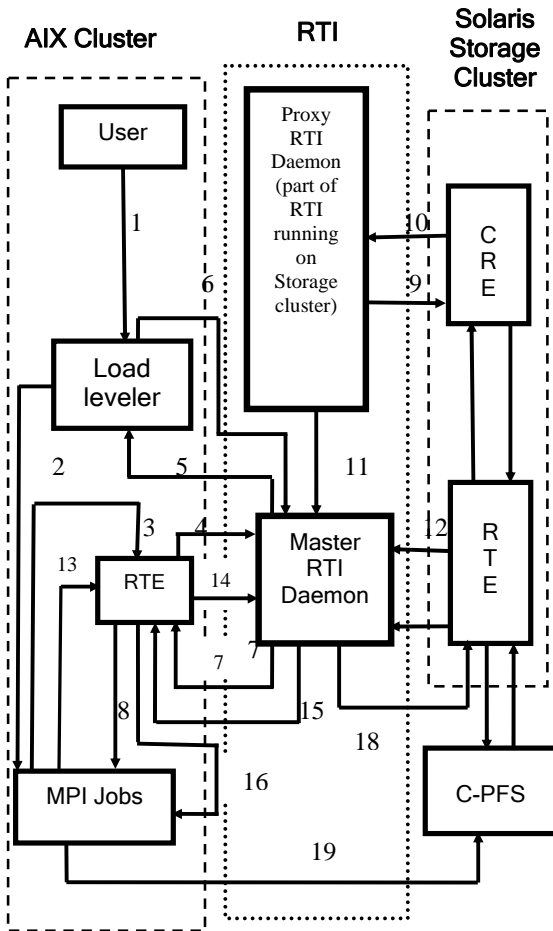


Fig. 2 Interaction of RTI with Load leveler and CDAC -CRE

5. RTI requests loadleveler to give the information requested by RTE.
6. Loadleveler gives the requested information to RTI
7. RTI in turn passes the information to RTE.
8. RTE on receiving the requested information pass on it to C-MPI.

9. Proxy RTI daemon requests CRE for the cluster table information of the storage cluster.

10. CRE gives the cluster table information to the proxy daemon.

11. On receipt of the cluster table, proxy daemon passes the same to master RTI daemon on PARAM Padma.

12. C-PFS informs Master RTI daemon about its port number through RTE.

13. MPI jobs request RTE for the port number of PFS.

14. RTE in turn requests the same from Master RTI daemon.

15. On receipt of the port number of C-PFS, Master daemon informs the same to RTE.

16. RTE receives the port number of C-PFS from master PFS daemon and passes the same to MPI jobs.

17. C-PFS contacts Master RTI daemon for the cluster and job table information through RTE.

18. Master RTI daemon gives the cluster and job table to C-PFS through RTE.

19. Upon receipt of the port number of the C-PFS, MPI jobs interact with C-PFS.

VI. EXPERIMENTAL SETUP

The effectiveness of the proposed high performance RTI interface is tested with the C-DAC's Tera-Scale Supercomputing Facility (CTSDF) is located at C-DAC Bangalore, India. As shown in Fig. 3.



Fig. 3 Picture of C-DAC's Tera-Scale Supercomputing Facility (CTSDF)

A. Test Bed Environment for RTI

PARAM Padma as shown in Fig. 3 is C-DAC's High performance scalable computing cluster, currently operating with a peak computing power of One Teraflop. The Computing and storage configuration is shown of PARAM padma are given in Table I. Effectiveness of the proposed RTI has been tested with PARAM Padma AIX cluster of 4 ways

SMP nodes connected by PARAMNet network as given in Table I.

TABLE I
DESCRIPTION OF PARAM PADMA

Specification	Compute nodes	File servers
Configuration	62 nos. of 4 ways SMP and one node. of 32 way SMP	6 nos of 4 ways SMP
No. of processors	248(Power 4@1GHz)	24(UltraSparc-IV@900MHz)
Aggregate memory	0.5 Terabytes	96 Gigabytes
Internal storage	4.5 Terabytes	0.4 Terabytes
Operating system	AIX/LINUX	Solaris
Peak computing power	992 GF (~1 TF)	--
File system	--	QFS

B. Performance Evaluation and Discussion

We conducted three sets of experiments to evaluate the performance of proposed RTI implemented on PARAM Padma test bed. These experiments are HPL Benchmarking; Two-dimensional Naviers stokes problem and PALLAS benchmarking on the PARAM Padma Tera-scale systems.

HPL, High Performance Linpack [15] is a popular benchmark suite to evaluate the performance of Super Computers and Clusters and involves solving a system of dense linear system in double precision (64-bits) arithmetic linear equations. The PARAM Padma cluster efficiency tested by using HPL benchmark. The performance found to be approximately 532 GFlops to the peak performance of 992 GFlops on PARAM Padma as shown in Fig. 4.

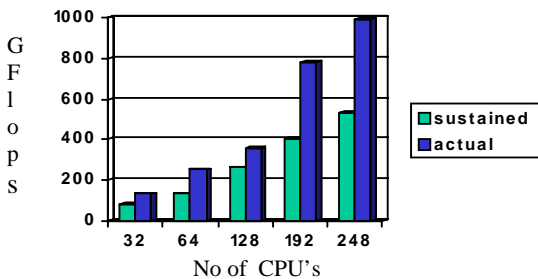


Fig. 4 Shows the HPL Performance on PARAM Padma using RTI

PARAM Padma sustained performance is 53.6% of peak performance and the matrix size used is 224000. PARAMNet interconnect has been found to perform significantly better in all HPL tests and RTI scales very well up to 62 nodes (248 processors) and beyond. The parallel software is centered upon the implementation of very high precision 3-D seismic migration and modeling algorithms, and our experimental test

indicates that the RTI software is scalable to a very large number of processors Fig. 5 gives the performance details of

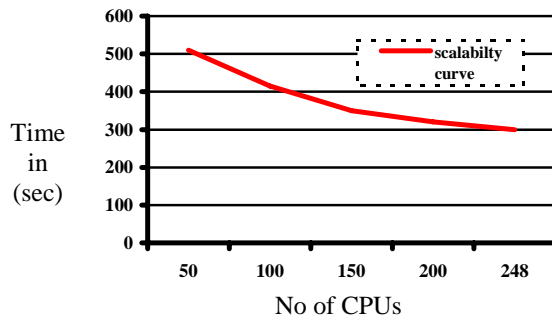


Fig. 5 Performance of two-dimensional Navier Stokes problem on PARAM Padma

2-D Navier-Stokes problem on PARAM Padma up to 248 processors, showing very high scalability. RTI interface has minimal interference with MPI job communications in PARAMPadama the communication overhead is measured very using complex PALLAS Benchmark PMB[16] is a set of MPI benchmarks namely PingPong, PingPing, Sendrecv, Exchange, Allreduce, Reduce, Reduce_scatter, Allgather, Allgather, Alltoall, Bcast and Barrier. The benchmarks Pingpong, PingPing, Sendrecv, Exchange, Allreduce, Reduce, Reduce_scatter, Bcast and Barrier have been executed on 32 nodes (128 processes) successfully. Other benchmarks showed irregular behavior using CMPI. The latency obtained using Pingpong benchmark is 26.18 μs using CMPI and 24.38 μs using Public Domain MPI (MPICH) across two AIX nodes tf01 and tf02.the summary results are shown in Table II .The benchmarks automatically execute for 2, 4, 8, 16, 32, 64, 128 and so on.

TABLE II
PALLAS BENCHMARK SUMMARY RESULTS FOR LATENCY AND BANDWIDTH

Communication Overheads	CMPI using RTI as job scheduler interface	Public Domain MPI MPICH
Latency	26.18μs	24.38 μs
Bandwidth	114.69 MBps	106.13 MBps

VII. CONCLUSIONS AND FUTURE WORK

The RTI is designed to provide efficient environment to schedule parallel jobs with high scalability. It operates in networks, including large scale clusters of workstations, supercomputers, and massively parallel systems. The RTI also provides the data storage support on Solaris platform for AIX Jobs. The RTI is designed with a powerful set of features enabling users to conduct the most complex of applications quickly and efficiently and find out how the job scheduler can help manage our workload. RTI can be extended as GSI Grid Scheduler Interface Suite [17] for the heterogeneous computing platforms that enable Processes to run single or

multiple applications operating across one or more heterogeneous servers.

REFERENCES

- [1] PARAM10000 Supercomputer ,Centre for Development of Advanced Computing(C-DAC) <http://www.cdac.in/HTML/param.asp>
- [2] CDAC Terascale supercomputing facility CTSF, www.cdac.in/html/ctsf/resource.asp
- [3] PARAM Padma Supercomputing Cluster, C-DAC, <http://www.cdac.in/html/parampma.asp>
- [4] PARAMNet, CDAC, www.cdac.in/HTML/pdf/PARAMNet.pdf
- [5] IBM's Load Leveler. http://csit1cwe.fsu.edu/extra_link/LoadL/llv2mst10.html.
- [6] William Gropps, Ewing Lusk, Nathan Doss and Anthony Skjellum. "A High-Performance, Portable Implementation of MPI Message Passing Interface Standard". Available at <http://www.mcs.anl.gov/mpi/>
- [7] IBM's Parallel Operating System (poe). <http://www-03.ibm.com/systems/p/software/pe.html>
- [8] Resource Management System, Centre for Development of Advanced Computing (C-DAC), India. at www.cdac.in/html/ssdgbler/rms.asp
- [9] Chansup Byun, Christopher Duncan and Stephanie Burk: "A Comparison of Job Managements Systems in Supporting HPC Cluster Tools" SUPERG, Vancouver, Fall 2000.
- [10] CDAC Cluster Runtime Environment. C-HPC Cluster Tools. Available at <http://www.cdac.in/html/hpcc.asp>
- [11] Portable Batch System. Available at <http://www.openpbs.com>
- [12] C-DAC Parallel File Systems. Available at www.cdac.in/html/ssdgbler/cpfs.asp
- [13] Sun Community Source Licensing. <http://www.sun.com/software/communitysource>
- [14] CDAC Message Passing Interface, <http://www.cdac.in/html/ssdgbler/cmpi.asp>
- [15] HPL - A Portable Implementation of the High-Performance Linpack Available at www.netlib.org/benchmark/hpl/
- [16] PALLAS Benchmark Available at <http://www.pallas.com/e/products/index.htm>
- [17] C-DAC, Garuda India, The National Grid Computing Initiative http://www.garudaindia.in/tech_research.asp