Available online at www.sciencedirect.com
ScienceDirect
Future Generation Computer Systems 24 (2008) 158–165
FGCS FUTURE GENERATION COMPUTER SYSTEMS
www.elsevier.com/locate/fgcs

MGF: A grid-enabled MPI library☆

F. Gregoretti[a], G. Laccetti[b], A. Murli[b], G. Oliva[a,*], U. Scafuri[a]

[a] *Institute of High Performance Computing and Networking ICAR-CNR, Naples, Italy*
[b] *University of Naples Federico II, Naples, Italy*

Received 29 September 2006; received in revised form 18 January 2007; accepted 26 March 2007
Available online 6 April 2007

Abstract

Computational grids allow access to several computing resources interconnected in a distributed heterogeneous infrastructure for parallel computing. This powerful resource aggregation increases the application runtime environment complexity. A simple programming model, capable of hiding this complexity, facilitates the use of grid technology in high-performance computing. The message passing interface can play this role and make the grid more accessible to developers with parallel programming skills.

In this paper we present MGF, a grid-enabled MPI implementation which extends the existing MPICH-G2. MGF aims are: to allow the transparent use of coupled Grid resources within the MPI library; to give programmers a detailed view of the execution system network topology; to use the most efficient channel available for point-to-point communications and finally, to improve collective operation efficiency by introducing a delegation mechanism.

Keywords: MPI; Message passing; Grid computing; MPICH-G2

1. Introduction

Grid technologies [1] allow the aggregation of multiple distributed resources for a single application execution. We refer to multi-site parallel applications [2] as those consisting of multiple groups of processes running on one or more potentially heterogeneous distributed resources. These applications can exploit the high-performance nature of the grid.

Aggregation of high-performance resources in the grid increases the application runtime environment complexity: we deal with different levels of memory hierarchy, various interconnection topologies and technologies and heterogeneous architectures.

The present and future success of computational grids depends on the ability of libraries and tools to hide this complexity from the users. MPI-based programming environments can make this technology more accessible for end-users with parallel programming skills and can be adopted for multi-site application development. MPI is easy to understand and use, architecture-independent, portable and a widely-used standard for high-performance computing. The focus of our work was to design, develop and test an MPI implementation that simplifies the extension of MPI applications to the grids.

Previous work [3] shows that topology-aware communication patterns can improve the efficiency of MPI collective operations in grid environments. Furthermore the use of communication daemons on the front-end nodes of clusters, like those provided by the PACX-MPI library [4], enables transparent inter-cluster communication, thus increasing portability of MPI codes from traditional parallel machines to metacomputers and grids. The use of a delegation mechanism to avoid needless message passing through these daemons is a further improvement for inter-cluster communications in collective operations [6].

We have developed a library called MGF based on MPICH-G2 [7], which implements the communications daemons on the PACX-MPI model with the delegation mechanism mentioned

☆ This work has been partially supported by Italian Ministry of Education, University and Research (MIUR) within the activities of the WP9 workpackage "Grid Enabled Scientific Libraries", part of the MIUR FIRB RBNE01KNFP *Grid.it* project.

* Corresponding author.
E-mail addresses: francesco.gregoretti@na.icar.cnr.it (F. Gregoretti), giuliano.laccetti@dma.unina.it (G. Laccetti), almerico.murli@dma.unina.it (A. Murli), oliva.g@na.icar.cnr.it (G. Oliva), scafuri.u@na.icar.cnr.it (U. Scafuri).

0167-739X/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2007.03.009

above. The use of MGF is transparent to the user in the sense that no modification of the source code is required. Moreover MGF expands the topology description provided by MPICH-G2 by including information about existing private networks.

In Section 2 we briefly review the state of the art in Grid-enabled MPI implementations; in Section 3 we give a detailed description of the MGF architecture; in Section 4 we describe experimental results and illustrate the benefits of the delegation mechanism by comparing our implementation with PACX-MPI; in Section 5, we discuss about our future work and in Section 6 we give our conclusions.

2. State of the art

There are several projects for the realization of MPI libraries for grids [8]: MagPIe [9], MPICH-G2, MPI_Connect [10], MetaMPICH [11], Stampi [12], PACX-MPI and MPICH/Madeleine III [13], etc. Many of these implementations allow multiple machines, potentially based on heterogeneous architectures, to be coupled for MPI program execution and to use vendor-supplied MPI libraries over high-performance networks for intra-machine messaging. The most widespread and complete are MPICH-G2, PACX-MPI and MPICH/Madeleine III.

- *MPICH-G2* is a grid-enabled implementation of the MPI v1.1 standard which uses grid services provided by the Globus Toolkit for user authentication, resources allocation, file transfer, I/O management, process control and monitoring. MPICH-G2 is based on the MPICH library, which is developed and distributed by the Mathematics and Computer Science Division at Argonne National Laboratory. MPICH-G2 implements topology-aware collective operations that minimize communications over the slowest channels.
- *PACX-MPI* is a complete MPI-1 standard implementation and supports some routines of the MPI-2 standard. PACX-MPI is developed by the Parallel and Distributed Systems working group of The High Performance Computing Centre in Stuttgart. PACX-MPI uses daemon processes executing on the front-end nodes of each parallel computer for inter-machine communications.
- *MPICH/Madeleine III* is an MPI implementation based on Madeleine III [13] a multi-device library able to transparently support most of the important protocols for intra-cluster communication (VIA, BIP/Myrinet, SISCI/SCI, etc.). Madeleine III has a built-in inter-device forwarding functionality to transfer messages across networks. This functionality has been carefully designed in order to minimize the latencies in the forwarding operation. MPICH/Madeleine III is thus capable of routing messages between heterogeneous networks even when these are based on different protocols and every node belonging to more than one network can become a gateway using any of the supported protocols. Madeleine III has been designed to be used when links between clusters are fast as internal cluster links and therefore is not suitable for a generic grid environment.

We chose to base our work on MPICH-G2 because we believe that many of its features are very useful in grid environment. For instance, MPICH-G2 provides the user with an advanced interconnecting topology description with multiple levels of depth, thus giving a detailed view of the underlying execution environment. It uses the Globus Security Infrastructure [14] for authorization and authentication and the Grid Resource Allocation and Management protocol [15] for resources allocation. Furthermore, MPICH-G2 is not cluster-specific and hence enables the use of any type of grid resource (e.g. single hosts) for MPI process execution. Finally it implements multilevel topology-aware collective communications, which have been proved [3] to perform better than the PACX-MPI two-level approach.

However, MPICH-G2 usage becomes complicated for application developers in the presence of clusters where only the front-end node is provided with a public IP address. This is due to the fact that unlike PACX-MPI, MPICH-G2 doesn't provide any routing mechanism among networks. Therefore, MPI processes started on computing nodes belonging to different private networks are unable to contact one another. This prevents the transparent porting of MPI application to grids where clusters with private networks are used.

3. MGF library

MGF (MPI Globus forwarder) is an MPI library based on MPICH-G2 that enables the transparent coupling of multiple grid resources for the execution of MPI programs. In particular, communication is made possible between clusters with private networks. From the application's point of view all the resources will appear as a single parallel computer. The principal aims of MGF are to allow parallel MPI applications to be executed on grids without modification of the source code; to give a programmer a detailed view of the underlying network topology of system during execution; to use the most efficient channel available for any point-to-point communication and finally, to implement efficient collective operations.

An example of a simple computational grid where MGF can be used is depicted in Fig. 1.

The picture shows four parallel machines located on two different sites. Site A hosts a cluster of four nodes each equipped with a network interface with a public IP address and a cluster with a private network where only a node has a public IP address. Site B hosts a private network cluster and a parallel computer with an high performance network whose nodes have all a public IP address. In a private network cluster, we distinguish the *front-end node* that is connected to both Internet and cluster network from *internal nodes* only connected to the cluster network.

3.1. Communication channels

In this context we define a *communication channel* as the network path that a message needs to follow from a source MPI process to a destination. We distinguish between two communication channels classes: