

IEEE DISTRIBUTED SYSTEMS ONLINE 1541-4922 © 2005 Published by the IEEE Computer Society
Vol. 10, No. 10; October 2005

Cluster Computing and Grid 2005 Works in Progress

This is the second in a two-part series () of works-in-progress articles taken from a special session, which was part of the Cluster Computing and Grid 2005 conference (<http://www.cs.cf.ac.uk/ccgrid2005>), held in Cardiff, UK. The session was organized by Mark Baker (University of Portsmouth, UK) and Daniel S. Katz (Jet Propulsion Laboratory, US). For more information, you can contact the session organizers or the authors of the articles.

A Pluggable Architecture for High-Performance Java Messaging

Mark Baker, *University of Portsmouth*
Aamir Shafi, *University of Portsmouth*
Bryan Carpenter, *University of Southampton*

Efforts to build Java messaging systems based on the Message Passing Interface (MPI) standard have typically followed either the JNI (Java Native Interface) or the pure Java approach. Experience suggests there's no "one size fits all" approach because applications implemented on top of Java messaging systems can have different requirements. For some, the main concern might be portability, while for others it might be high bandwidth and low latency. Moreover, portability and high performance are often contradictory requirements. You can achieve high

performance by using specialized communication hardware but only at the cost of compromising the portability Java offers. Keeping both in mind, the key issue isn't to debate the JNI versus pure Java approaches, but to provide a flexible mechanism for applications to swap between communication protocols.

To address this issue, we have implemented MPJ Express based on the *Message Passing in Java (MPJ)* API.¹ MPJE follows a layered architecture that uses device drivers, which are analogous to Unix device drivers. The ability to swap devices at runtime helps mitigate the applications' contradictory requirements. In addition, we're implementing a runtime system that bootstraps MPJE processes over a collection of machines connected by a network. Though the runtime system isn't part of the MPI specifications, it's essential to spawn and manage MPJE processes across various platforms.

MPJE's design is layered to allow incremental development and provide the capability to update and swap layers in or out as needed. **Figure 1** shows a layered view of the messaging system. The high and base levels rely on the MPJ device² and *xdev* level for actual communications and interaction with the underlying networking hardware. One device provides JNI wrappers to the native MPI implementations, and the other (*xdev*) provides access to Java sockets, shared memory, or specialized communication libraries. The wrapper implementation doesn't need *xdev*, because the native MPI is responsible for selecting and switching between different communication protocols. **Figure 1** also shows three implementations of *xdev*: *smpdev*, the shared memory device; *niodev*, an implementation of *xdev* using the Java New I/O package; and *gmdev*, an implementation of *xdev* using JNI to interact with the Myrinet communications library.

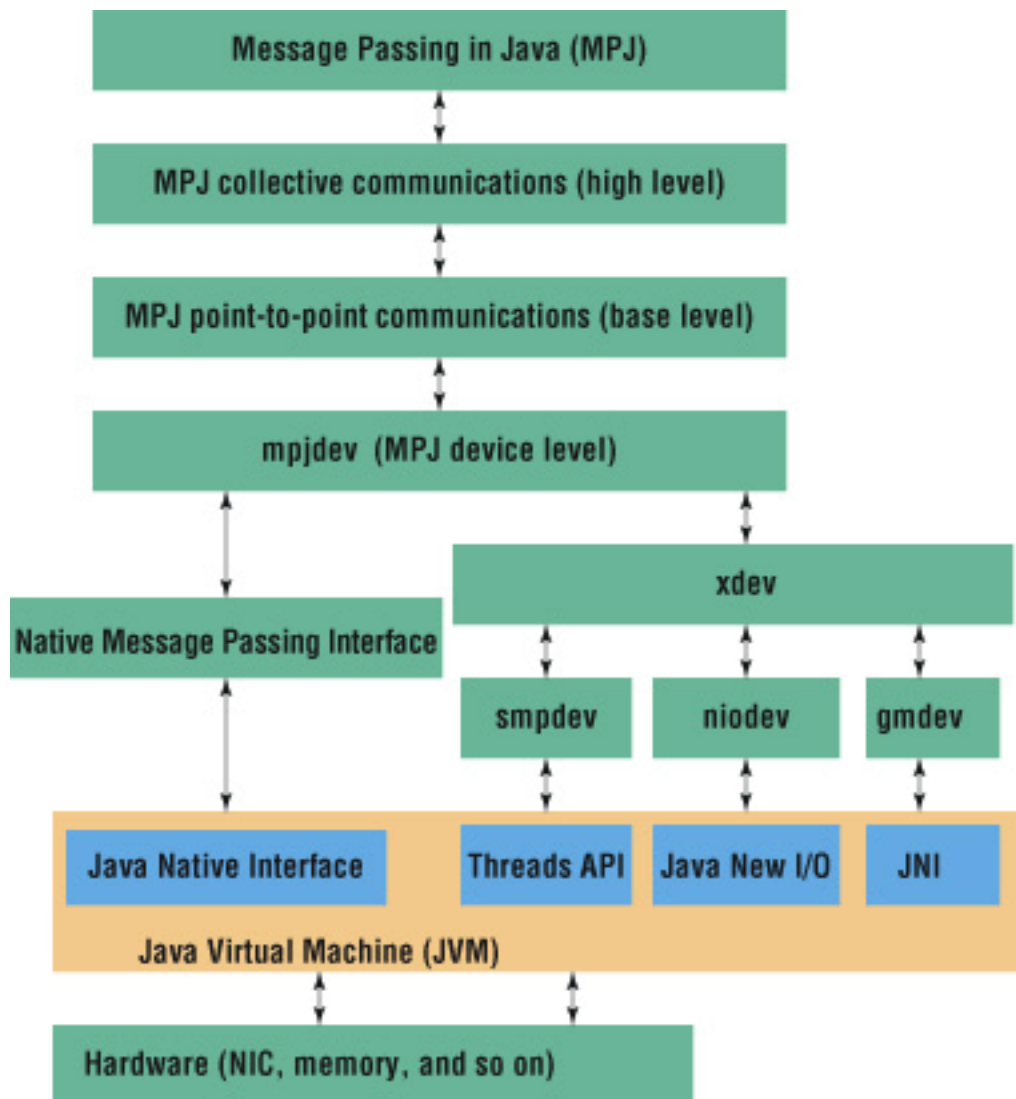


Figure 1. MPJ Express's layered design.

MPJE's initial performance evaluation on Fast and Gigabit Ethernet shows comparable performance to mpiJava, which uses JNI wrappers to interact with a native MPI implementation. We released a beta version of MPJE in early September. You can find further details of MPJE on the project's Web site (<http://dsg.port.ac.uk/projects/mpj> or email us.

References

1. B. Carpenter et. al, *MPI for Java—Position Document and Draft API Specification*, tech. report JGF-TR-03, Java Grande Forum, Nov. 1998.

2. S.B. Lim et. al, "A Device Level Communication Library for the HPJava Programming Language." *Proc. Iasted Int'l Conf. Parallel and Distributed Computing and Systems* (PDCS 2003), ACTA Press, 2003.

Mark Baker is a Reader in Distributed Systems at the University of Portsmouth, UK. Contact him at mark.baker@computer.org.

Bryan Carpenter is a senior researcher at the Open Middleware Infrastructure Institute, University of Southampton, UK. Contact him at dbc@ecs.soton.ac.uk.

Aamir Shafi is a PhD student in the Distributed Systems Group, University of Portsmouth, UK. Contact him at aamir.shafi@port.ac.uk.

Toward Intelligent, Adaptive, and Efficient Communication Services for Grid Computing

Phillip M. Dickens, *University of Maine*

What constitutes an intelligent, adaptive, highly efficient communication service for grid computing? An intelligent service can accurately assess the end-to-end system's state to determine how (and whether) to modify the data transfer's behavior. An intelligent controller could, for example, respond more aggressively to a network-related loss than to a loss caused by events outside the network domain. An adaptive communication service can either change its execution environment or adapt its behavior in response to changes in that environment. An efficient communication service can exploit the underlying network bandwidth when system conditions permit. It can also fairly share network resources in response to observed (or predicted) network contention.

A necessary milestone on the path to such next-generation communication services is the development of a *classification mechanism* that can distinguish between various data-loss causes in cluster or Grid environments. We're developing such a mechanism based on *packet-loss signatures*, which show the distribution (or pattern) of packets that successfully traversed the end-to-end transmission path versus those that did not. These signatures are essentially large

selective-acknowledgment packets that the data receiver collects and, upon request, delivers to the data sender. We refer to them as packet-loss signatures because a growing set of experimental results shows that different data-loss causes have different signatures.^{1,2} The question then is how to quantify the differences between packet-loss signatures so that a classification mechanism can identify them.

Our approach is to treat packet-loss signatures as time-series data and to apply techniques from symbolic dynamics to learn about the time series' dynamical structure. We quantify the structure in the sequence based on its complexity. We've learned that the complexity measures of packet-loss signatures have different statistical properties when the cause of such loss lies inside rather than outside the network domain. In fact, these statistical properties are different enough to let us construct, using Bayesian statistics, rigorous hypothesis tests regarding the cause of data loss.³ We're currently developing the infrastructure required to perform such hypothesis testing in real time.

Next, we plan to develop and evaluate a set of responses tailored to particular data-loss causes. We'll explore, for example, data-receiver migration and user-specified limits on CPU utilization for data loss caused by contention for CPU resources.

References

1. P. Dickens and J. Larson, "Classifiers for Causes of Data Loss Using Packet-Loss Signatures," *Proc. IEEE Symp. Cluster Computing and the Grid (CCGrid 04)*, IEEE CS Press, 2004.
2. P. Dickens, J. Larson, and D. Nicol, "Diagnostics for Causes of Packet Loss in a High Performance Data Transfer System," [http://csdl2.computer.org/persagen/DLAbsToc.jsp?Proc.18thInt'lParallelandDistributedProcessingSymp.\(IPDPS04\)](http://csdl2.computer.org/persagen/DLAbsToc.jsp?Proc.18thInt'lParallelandDistributedProcessingSymp.(IPDPS04)), IEEE CS Press, 2004.
3. P. Dickens and J. Peden, "Towards a Bayesian Statistical Model for the Causes of Data Loss," *Proc. 2005 Int'l Conf. High Performance Computing and Communications*, LNCS 3726, Springer, 2005, pp. 755-767.

Phillip M. Dickens is an assistant professor in the Department of Computer Science at the University of Maine. Contact him at dickens@umcs.maine.edu.

Grimoires: A Grid Registry with a Metadata-Oriented Interface

Sylvia C. Wong, *School of Electronics and Computer Science, University of Southampton*

Victor Tan, *School of Electronics and Computer Science, University of Southampton*

Weijian Fang, *School of Electronics and Computer Science, University of Southampton*

Simon Miles, *School of Electronics and Computer Science, University of Southampton*

Luc Moreau, *School of Electronics and Computer Science, University of Southampton*

The Grid is an open distributed system that brings together heterogeneous resources across administrative domains. Grid registries let service providers advertise their services, so users can use these registries to dynamically find available resources. However, existing service registry technologies, such as Universal Description, Discovery, and Integration (UDDI), provide only a partial solution.

First of all, such technologies have limited support for publishing semantic information. In particular, services aren't the only entities that need to be classified—for example, we would also want to define classifications for individual operations or their argument types. Second, only service operators can provide information about services, and in a large and disparate environment, it's impossible for operators to foresee all the information that users might use to find resources. Third, UDDI uses authentication techniques for security that aren't particularly suited for the large-scale nature of Grid systems.

To address these problems, we're developing a registry called Grimoires (<http://www.grimoires.org>) for the myGrid project (<http://www.mygrid.org.uk>) and the Open Middleware Infrastructure Institute (OMII, <http://www.omii.ac.uk>) Grid software release. Figure 2 shows our registry's architecture, which we've implemented as a Web service. It has two major interfaces—UDDI and metadata. The registry is UDDI v2 compliant, and you can access the UDDI interface using any UDDI client, such as UDDI4j (<http://www.uddi4j.org>). To access the metadata functionalities, you need to use a Grimoires client.

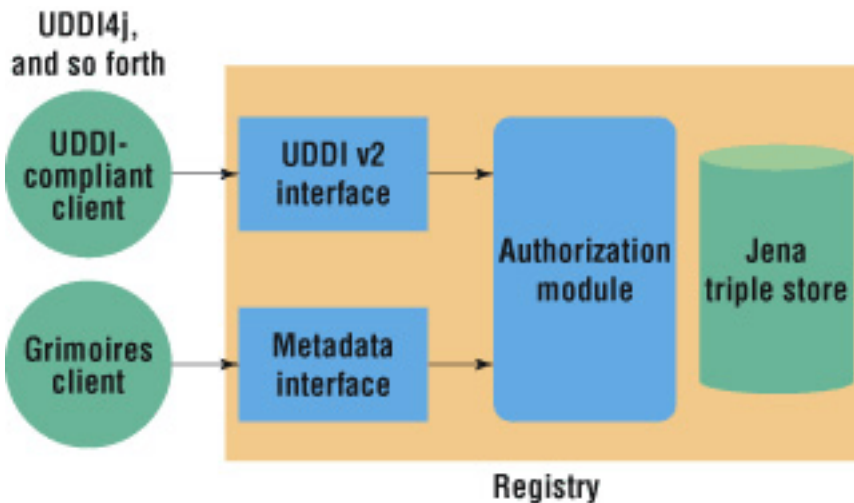


Figure 2. The Grimoires architecture. (UDDI is Universal Description, Discovery, and Integration.)

Our registry has several unique features:

- ◆ *Registration of semantic descriptions.* Our registry can publish and inquire over metadata attachments. These attachments are extra pieces of data that provide information about existing entities in the registry. Currently, the registry supports annotations to UDDI BusinessEntity, BusinessService, tModel, and BindingTemplate, and to WSDL (Web Services Description Language) operations and message parts. Thus, using Grimoires, users can annotate BusinessService with service ratings and functionality profiles and attach semantic types of operation arguments to WSDL message parts.
- ◆ *Multiple metadata attachments.* Each entity can have an unlimited number of attachments, and each piece of metadata can be updated without republishing the entity or other metadata attached to the same entity. This efficiently captures ephemeral information about services, which changes often.
- ◆ *Third party annotations.* Both service operators and third parties can publish metadata, so users with expert knowledge can enrich service descriptions in ways that the original publishers might not have conceived.
- ◆ *Inquiry with metadata.* Grimoires supports multiple search patterns. It ranges from simple searches that return a list of metadata attached to the specified entity to more complex searches that return entities that match a certain criteria.
- ◆ *Signature-based authentication.* UDDI uses a username and password credential scheme. However, Grid environments typically use certificate-based authentication.

OMII provides an implementation of SOAP message signing and verification that conforms to Web Services security standards. By deploying Grimoires in the OMII container, the registry can authenticate users using X509 certificates. This makes it easier to integrate Grimoires into existing Grid security infrastructures, and it provides an important building block—certificate-based authentication—for the single sign-on capabilities that many Grid applications require.

For more information, please visit www.grimoires.org.

Sylvia C. Wong is a research fellow in the Intelligence, Agents, Multimedia group at the School of Electronics and Computer Science, University of Southampton, UK. Contact her at sw2@ecs.soton.ac.uk.

Victor Tan is a research fellow in the Intelligence, Agents, Multimedia group at the School of Electronics and Computer Science, University of Southampton, UK. Contact him at vhkt@ecs.soton.ac.uk.

Weijian Fang is a research fellow in the Intelligence, Agents, Multimedia group at the School of Electronics and Computer Science, University of Southampton, UK. Contact him at wf@ecs.soton.ac.uk.

Simon Miles is a research fellow in the Intelligence, Agents, Multimedia group at the School of Electronics and Computer Science, University of Southampton, UK. Contact him at sm@ecs.soton.ac.uk.

Luc Moreau is a professor in the Intelligence, Agents, Multimedia group at the School of Electronics and Computer Science, University of Southampton, UK. Contact him at l.moreau@ecs.soton.ac.uk.

Cite this article:

Mark Baker, Bryan Carpenter, and Aamir Shafi, "Cluster Computing and Grid 2005 Works in Progress: A Pluggable Architecture for High-Performance Java Messaging," *IEEE Distributed*

Systems Online, vol. 6, no. 10, 2005.

Phillip M. Dickens, "Cluster Computing and Grid 2005 Works in Progress: Toward Intelligent, Adaptive, and Efficient Communication Services for Grid Computing," *IEEE Distributed Systems Online*, vol. 6, no. 10, 2005.

Sylvia C. Wong, Victor Tan, Weijian Fang, Simon Miles, and Luc Moreau, "Cluster Computing and Grid 2005 Works in Progress: Grimoires: A Grid Registry with a Metadata-Oriented Interface," *IEEE Distributed Systems Online*, vol. 6, no. 10, 2005.