Introduction

Efforts to build Java messaging systems based on the MPI standard have typically followed either the JNI approach or the pure Java approach. Experiences gained with these implementations suggest that there is no 'one size fits all' approach. The reason is that applications implemented on top of Java messaging systems can have different requirements. For some, the main concern could be portability, while for others high-bandwidth and low-latency could be the most important requirement. Portability and high-performance are often contradictory requirements. High performance can be achieved by making use of the specialized communication hardware, but only at the cost of compromising portability that Java offers. Keeping both in mind, the key issue at present is not to debate the JNI approach versus the pure Java approach, but to provide a flexible mechanism for applications to swap communication protocols.

MPJ Express

To address this issue, we are developing MPJ Express (MPJE) - an implementation of the MPI bindings for Java. This software follows a layered architecture based on the idea of device drivers; which are analogous to UNIX device drivers. The ability to swap devices at runtime helps mitigate the contradictory requirements of the applications. In addition, we are developing a runtime system that bootstraps MPJE processes over a collection of machines connected by a network. Though the runtime system is not part of the MPI specifications, it is essential to run MPJE processes across various platforms. Figure 1 shows the layered design of MPJE.

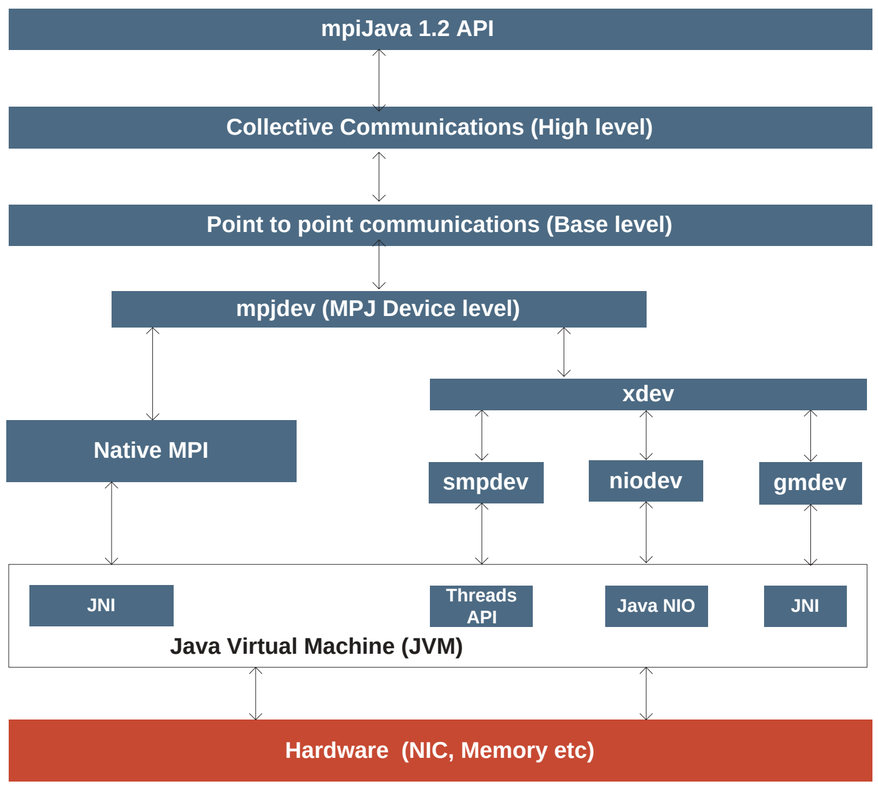

Figure 1: MPJ Express Design

Performance Evaluation

We present the the performance evaluation of MPJE against mpiJava 1.2.5, MPICH-1.2.5, and LAM/MPI-7.0.6 on Fast Ethernet. MPJE is using an initial implementation of a pure Java device based on Java New I/O (NIO) package. We have also evaluated MPJE against mpiJava on Gigabit Ethernet and found that MPJE performs similar to mpiJava on GE.

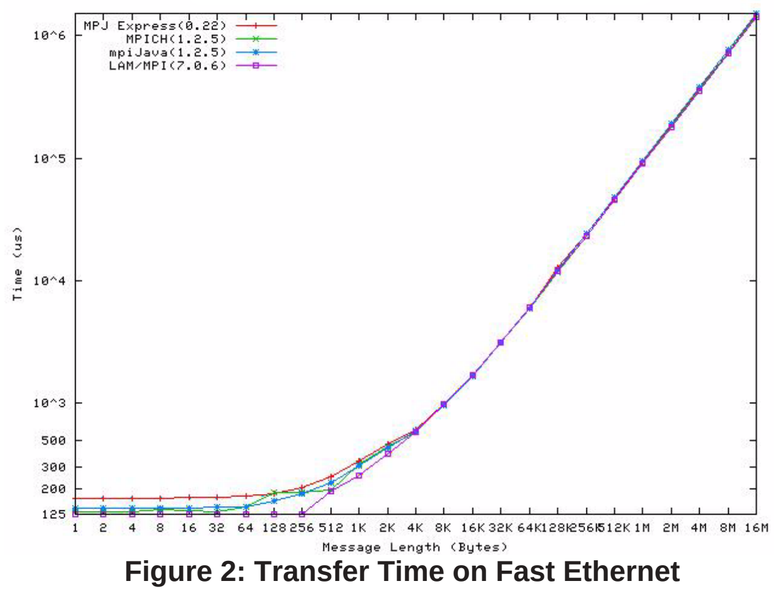

Figure 2: Transfer Time on Fast Ethernet

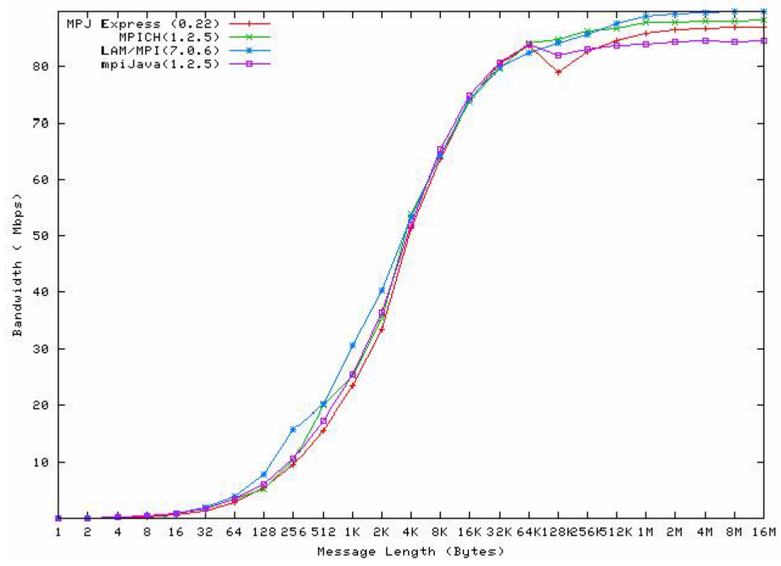

Figure 3: Throughput on Fast Ethernet

Demo Applications

MPJE follows the same API as mpiJava 1.2.x, which makes it possible to port existing parallel applications to this software. To support our claim, we have ported two mpiJava applications to MPJE. The first application is based on the Potts model which encompasses a number of problems in statistical physics and lattice theory. The application uses Metrops algorithm that starts with an initial configuration and repeats itself. The second application simulates a 2-D inviscid flow through an axisymmetric nozzle.

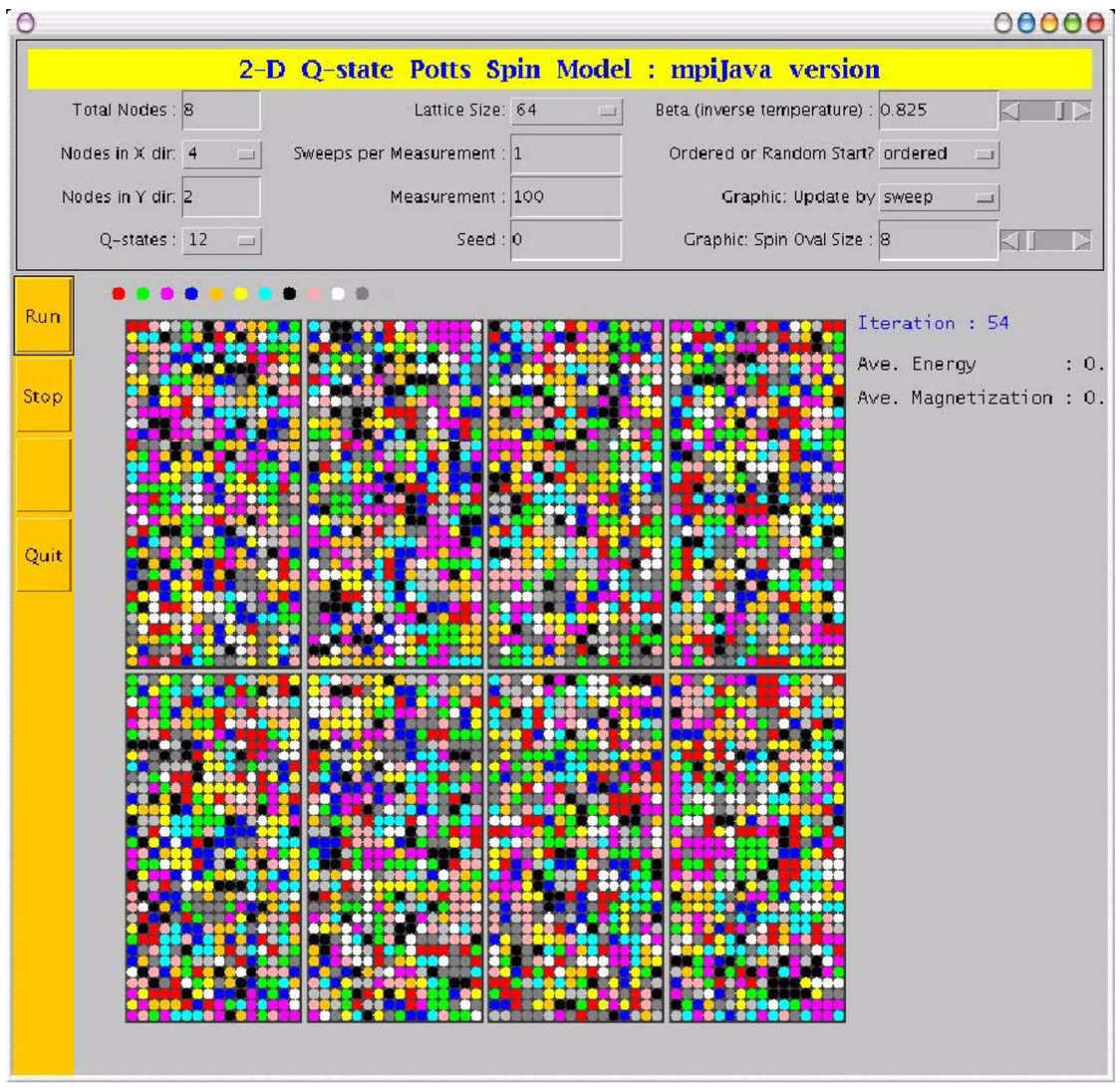

Figure 6: Statistical Physics Demo Application

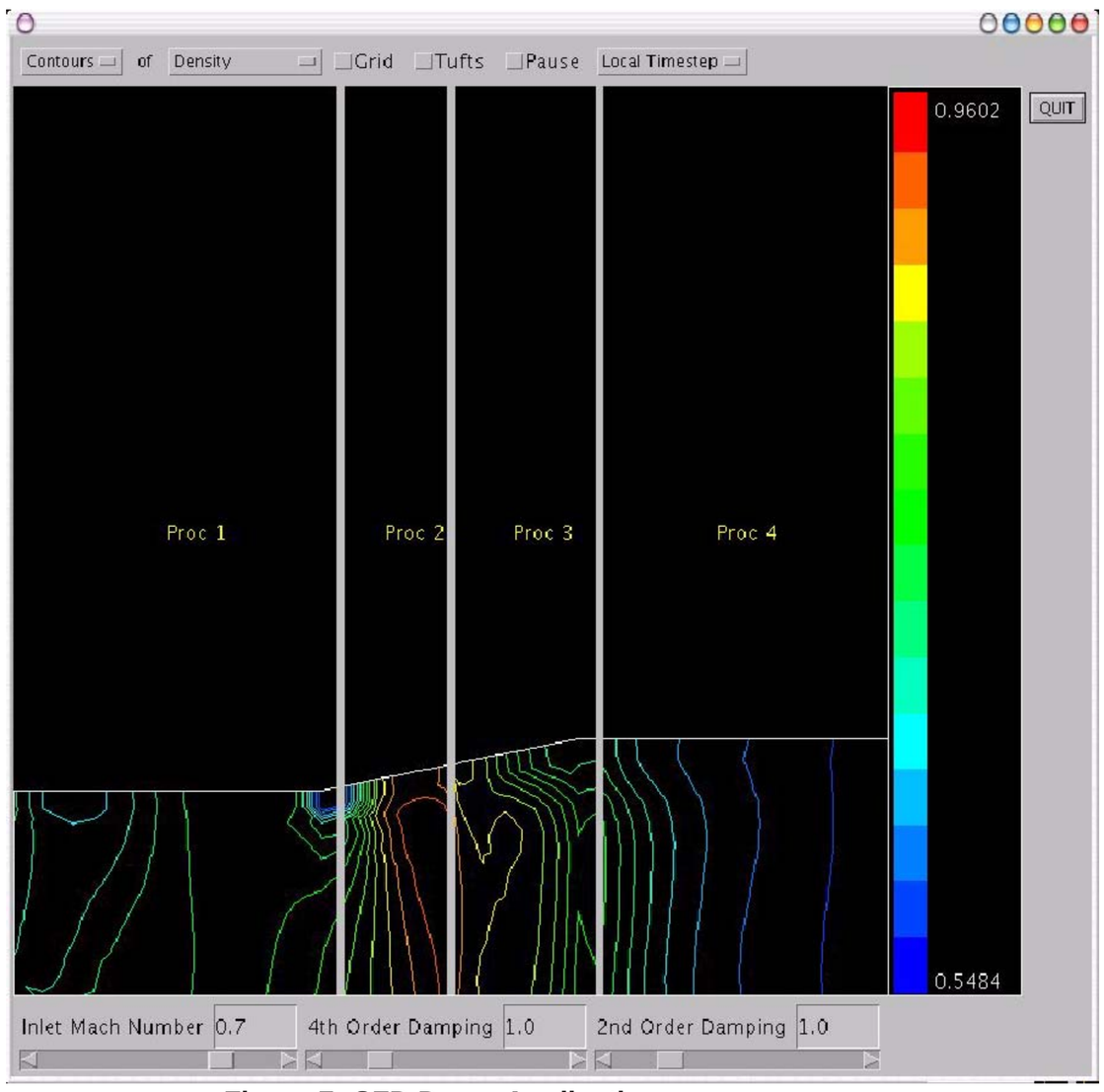

Figure 7: CFD Demo Application

Summary and Current Status

MPJE is an implementation of the MPI bindings for Java based on a modular design that allows swapping in or out various communication protocols. This software follows the same API as mpiJava, making it possible to port existing parallel Java applications to MPJE. The MPJE runtime infrastructure makes it possible to start Java applications on heterogeneous distributed platforms. A beta version of MPJE has been released and can be downloaded from the project's web site. Currently, we are developing a native MPI and a Myrinet device for MPJ Express.

Distributed Systems Group: http://dsg.port.ac.uk
MPJ Express: http://dsg.port.ac.uk/projects/mpj

Aamir Shafi -- aamir.shafi@port.ac.uk
Bryan Carpenter* -- dbc@ecs.soton.ac.uk
Mark Baker -- mark.baker@computer.org
*** Open Middleware Infrastructure Institute**