Metacomputing

An evaluation of emerging systems

Presented by Dr David Cronk

Overview

- Introduction
- Human Factors
- Site Autonomy
- System Functionality
- Future Growth
- Running MPI jobs
- Performance
- Conclusions

Introduction

- What is metacomputing?
  - Metacomputing consists of connecting geographically distributed high-performance computing resources in a seamless manner.
    - Hides the complexity of resource management (login and job submission) from the user.
    - Allows multiple high-performance resources to be used to solve problems that are too large to solve with traditional systems.
    - A single application can take advantage of different architectures for different tasks.
    - Facilitates sharing of input and output data.
  - There are many metacomputing systems in various stages of development.

Globus

- Being developed at ANL and ISI USC
- Grew out of the I-WAY project (1995)
- Toolkit consists of 6 modules
  - Resource location and allocation - Scheduling and initialization
  - Communication - Based on the Nexus communication library
  - Unified resource information service - Real-time system status
  - Authentication interface - Used to authenticate users and resources
  - Process creation - Initiates comp. on resource after location and allocation
  - Data access - Provides high speed remote access to persistent storage

Legion

- University of Virginia
- Object Oriented
- Designed as a metacomputing *framework*
- Legion objects are persistent
- File system presented to the user has no concept of physical location

Human factors

- **Installation and maintenance**
  - Globus: Multiple systems must be installed (LDAP, SSLEAY,MPICH-G). Must interact with Globus developers for certs. Maintain grid-mapfile and gatekeeper on EVERY machine.
  - Legion: Can use pre-compiled binaries. Install on a single machine and then add machines. Only add machines from machine Legion originally installed on. Full installation required following an ungraceful termination..
- **Ease of use**
  - Globus: For MPI programs, very simple with no code alteration.
  - Legion: No code alteration necessary, but new compilations must be registered. I/O is more difficult and requires some modification.

Site Autonomy

- **Security**
  - Globus: GSI focuses on authentication, not authorization, and is based on GSS. Users must get certs from CA and have local accounts on resources used.
  - Legion: Provides simple security implementations, but more sophisticated implementations are encouraged. Security policies can be enforced externally. Objects are identified by LOIDs, which use X509 certs.

Site Autonomy

- **Resource Management**
  - Globus: Uses RSL to communicate requests. Uses resource *brokers*, *co-allocators*, and *managers* (GRAM). GRAM can interface with local batch schedulers. Users must have local accounts AND be in grid-mapfile to use resource.
  - Legion: Completely decentralized, uses jurisdictions and magistrates. Magistrates and schedulers are simple by default, but users can implement more sophisticated policies.

System Functionality

- File System
  - Globus: Provides mechanisms for automatic staging of executables and copying data files. GASS provides mechanisms for remote I/O
  - Legion: The user sees the files system as a single name space. Context space is globally named and globally accessible. Executables must be registered. Input and output files can be specified at runtime.
- Language Support
  - Globus: C, Fortran, MPI, Compositional C++, Fortran M, nPerl
  - Legion: C, Fortran, Mentat, MPI, PVM, BFS

Future Growth

- Globus
  - Users must have an account on every machine. Decentralized directory service. Scalable fault detection.
  - LDAP and RSL are extensible. Incomplete data model.
- Legion
  - Scalability was a primary design goal. Jurisdictions and Magistrates improve scalability. File system uses a configuration database.
  - Extensibility was a key design issue. Built as a framework.

MPI: Globus

- Make sure MPICH-G is installed on your system
- Make sure your environment is setup to use the MPICH-G executables (mpicc, mpif77, and mpirun)
- Change file open/close calls if needed
- Recompile you application using the MPICH-G executables
- Setup a “machines” file in the working directory
- Use mpirun to start the job
  - Use -globusrsl to use customized rsl script

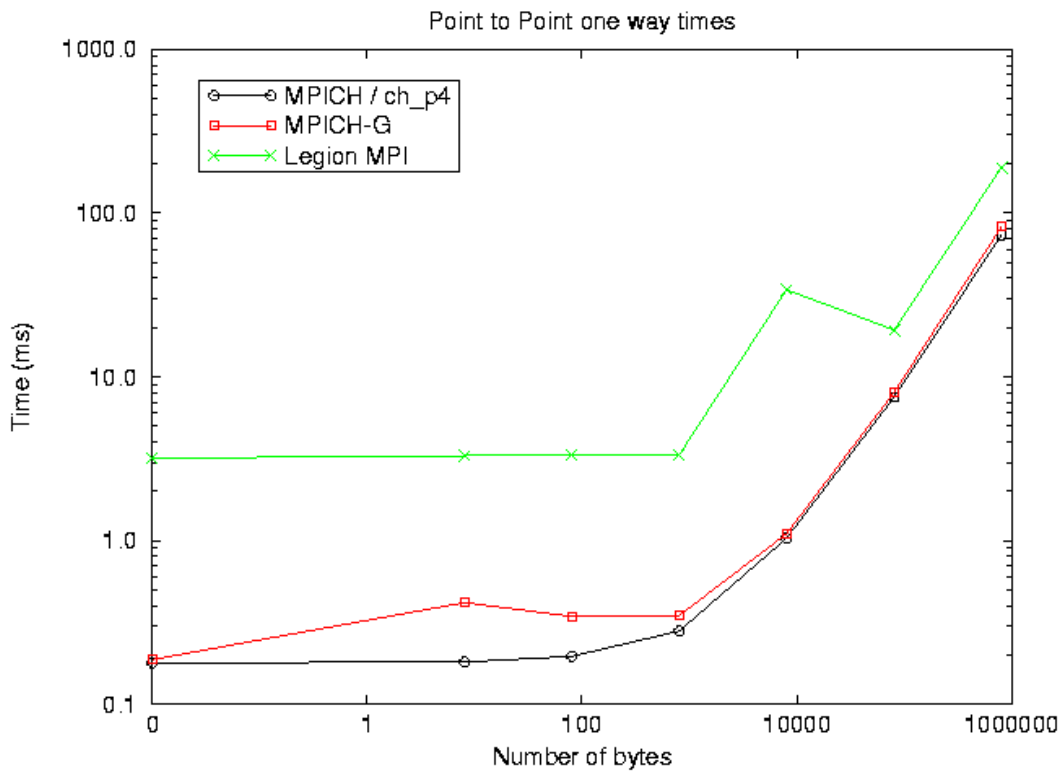
MPI: Legion

- Add Legion I/O calls if needed
- Recompile application using Legion MPI include files
- Link application using “legion_link”
- Register the executable with “legion_mpi_register”
- Import/export files as needed
- Setup legion_tty to monitor output
- Setup “hosts” context if you want to control what machines are used to run the job
- Use “legion_mpi_run to start execution

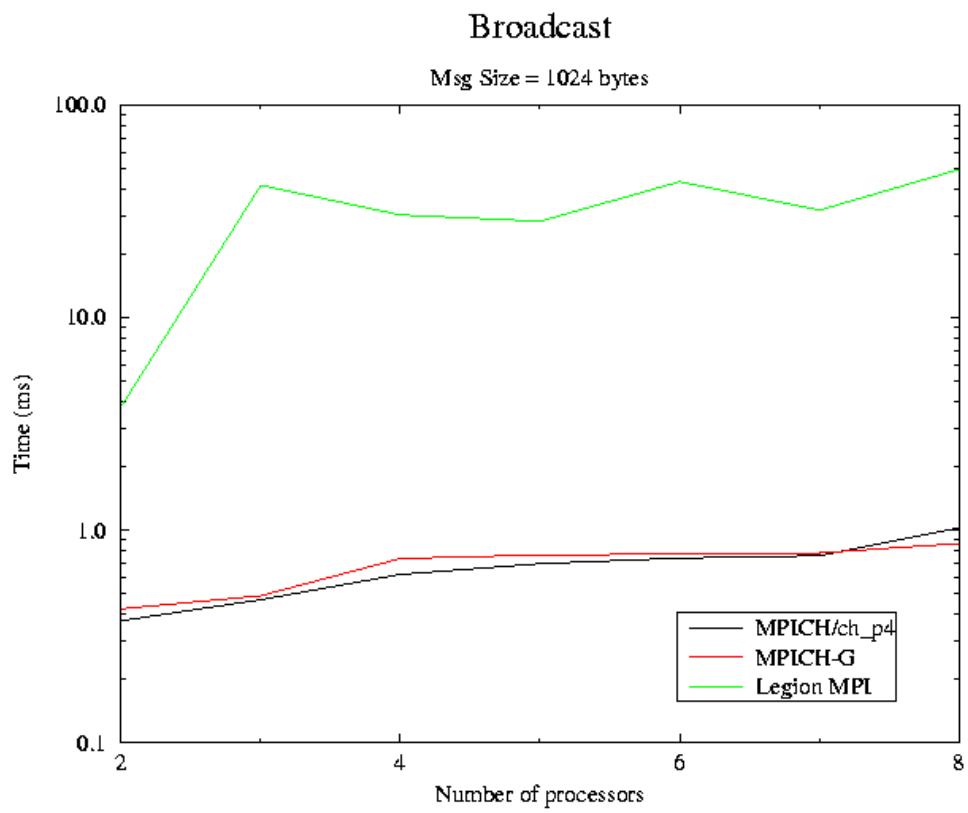
Performance

- Point-to-Point communication
- Collective operations
  - Broadcast
  - Barrier synchronization
  - All-reduce
- LU Factorization (ScaLAPACK)
- Linpack

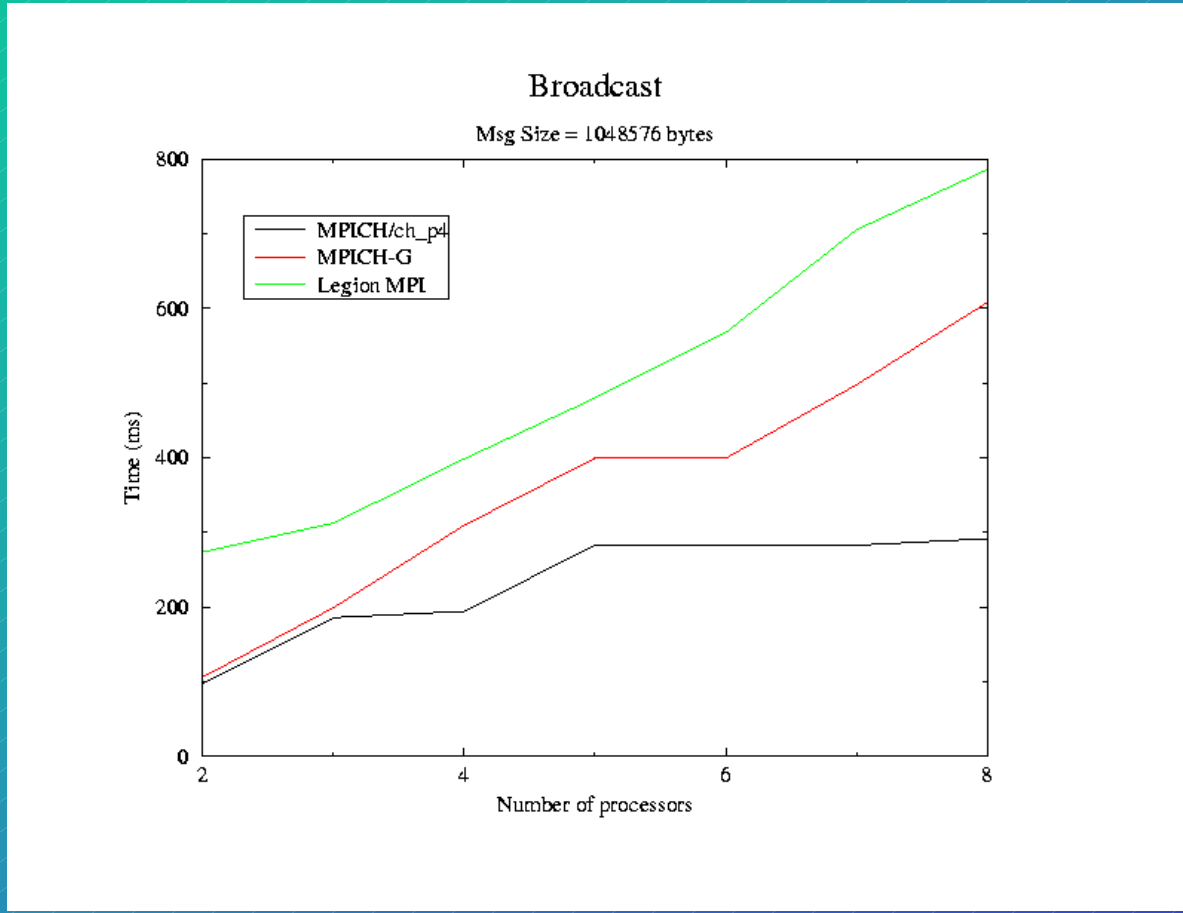
Point-to-Point Communication



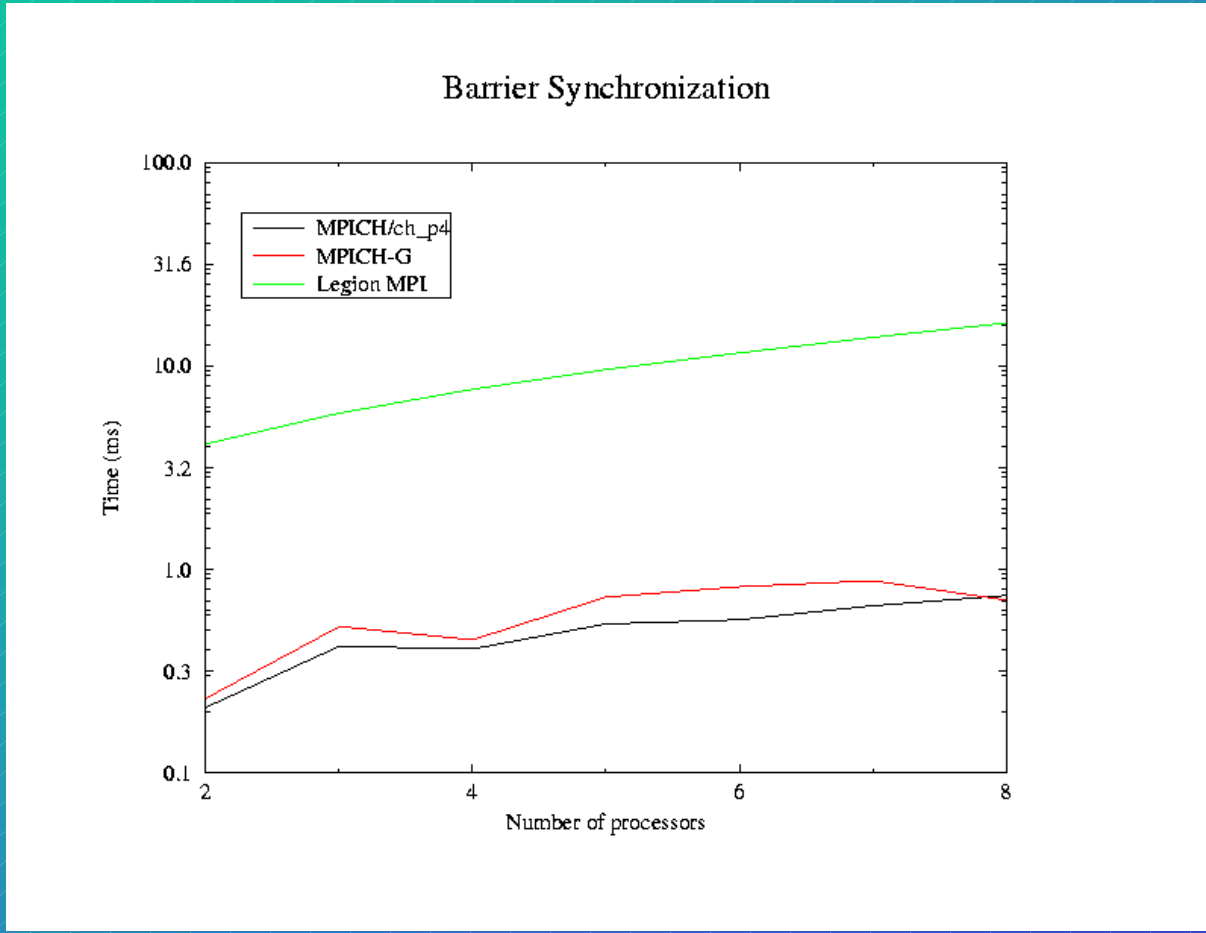
Broadcast (1K)



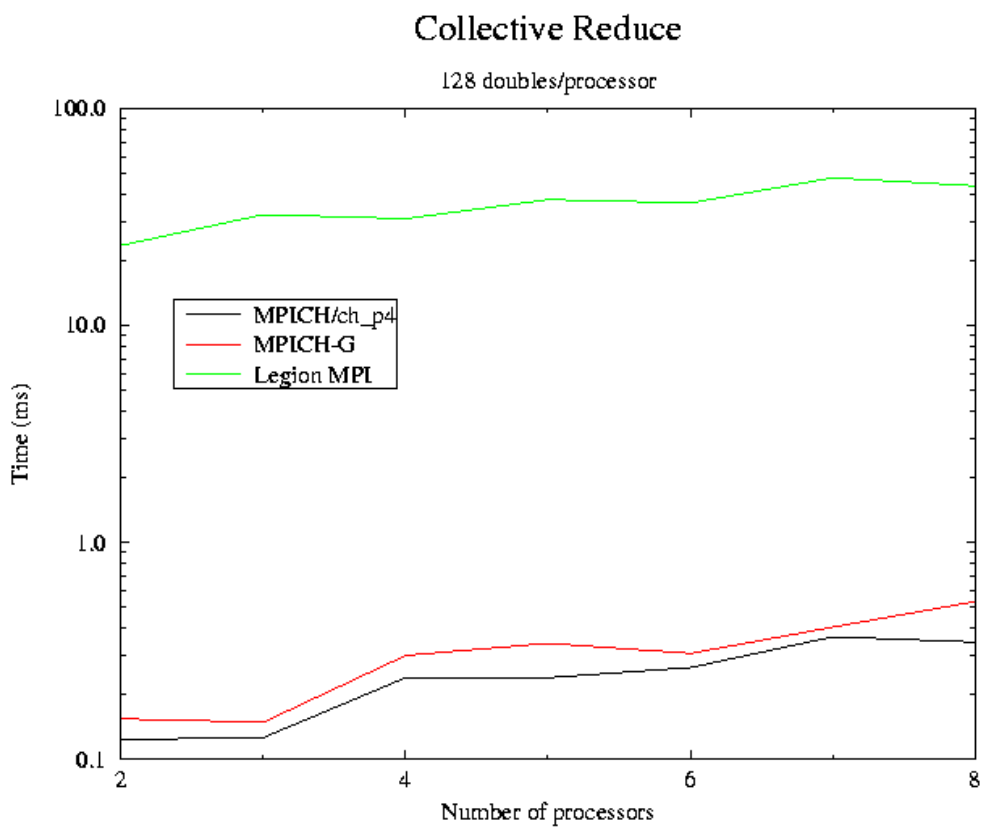
Broadcast (1 Meg)



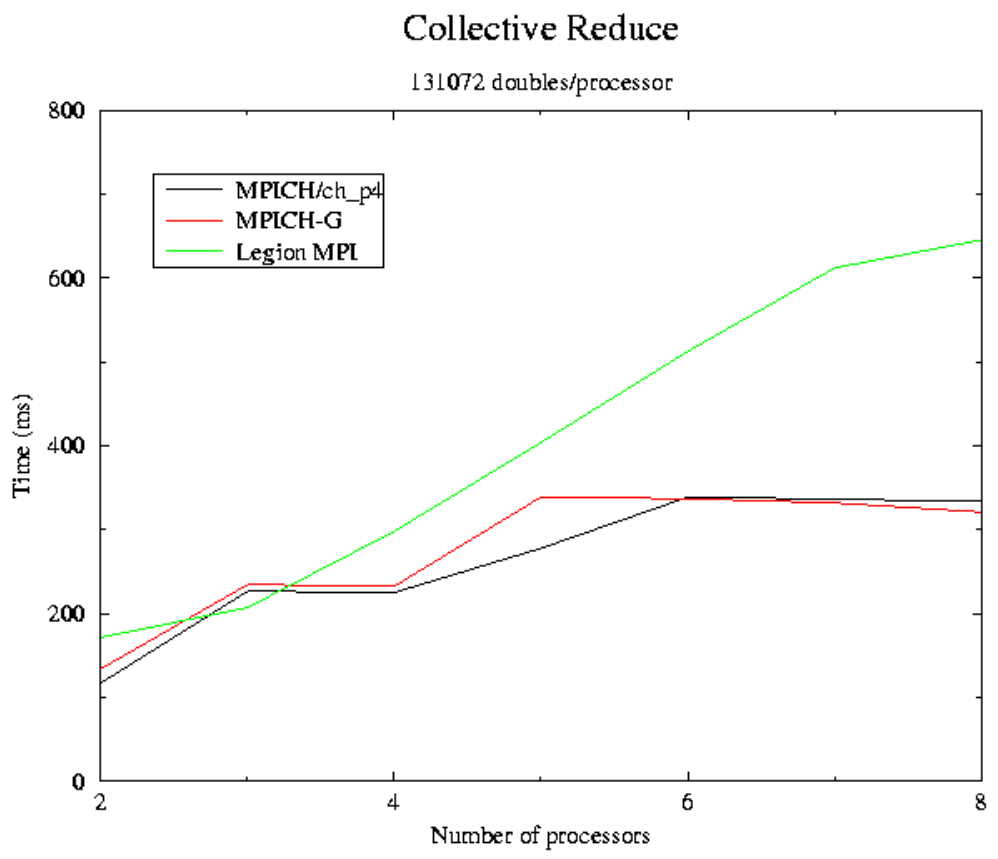
Barrier Synchronization



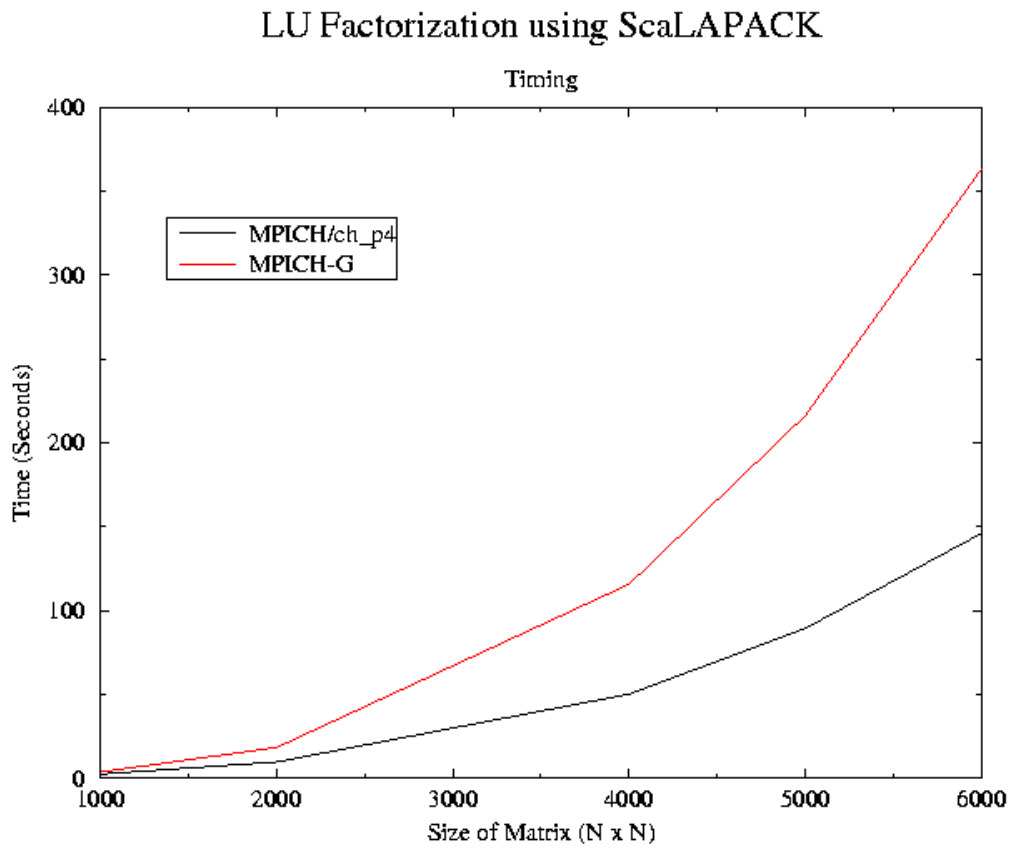
Collective Reduce (128)



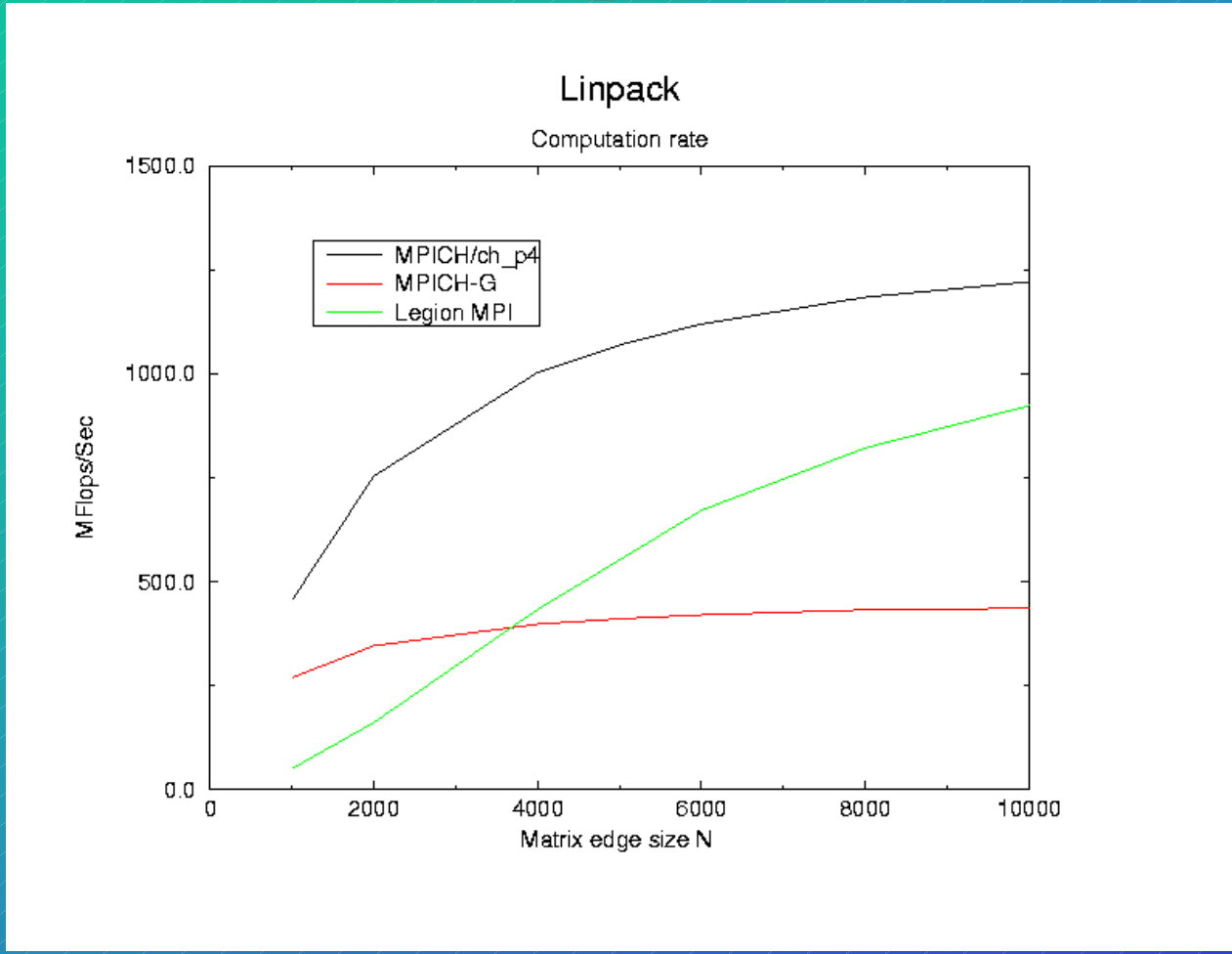
Collective Reduce (128K)



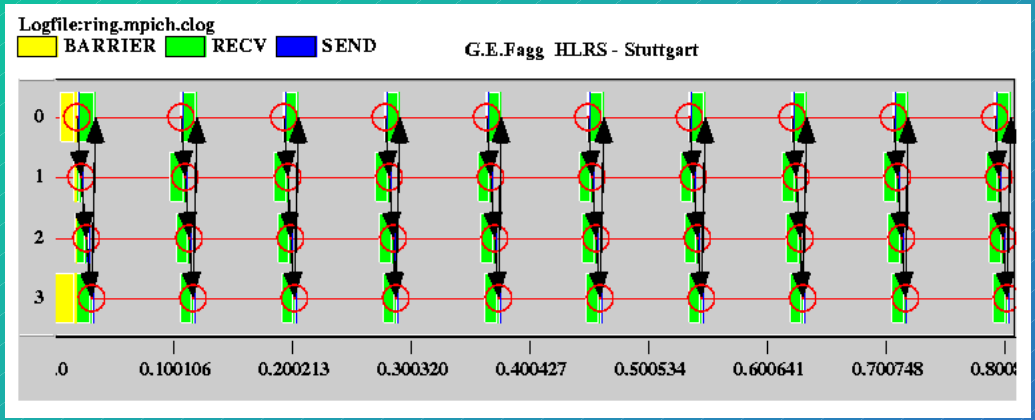
LU Factorization

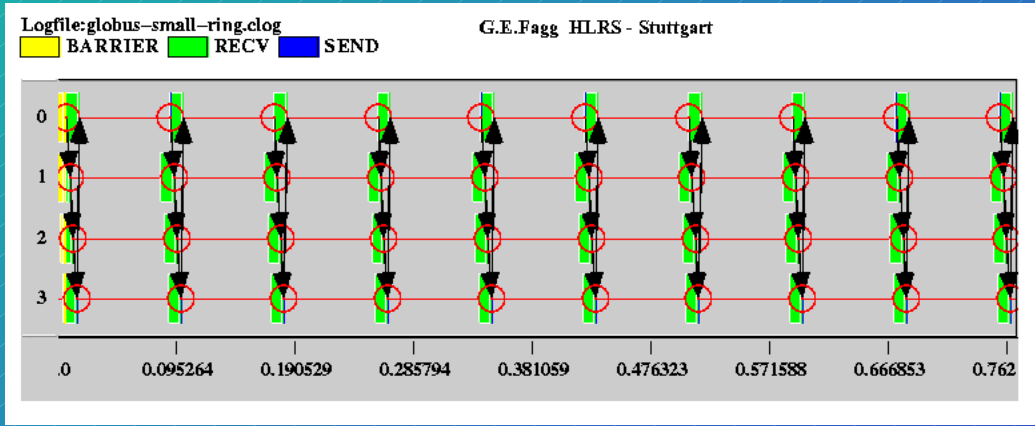


Linpack

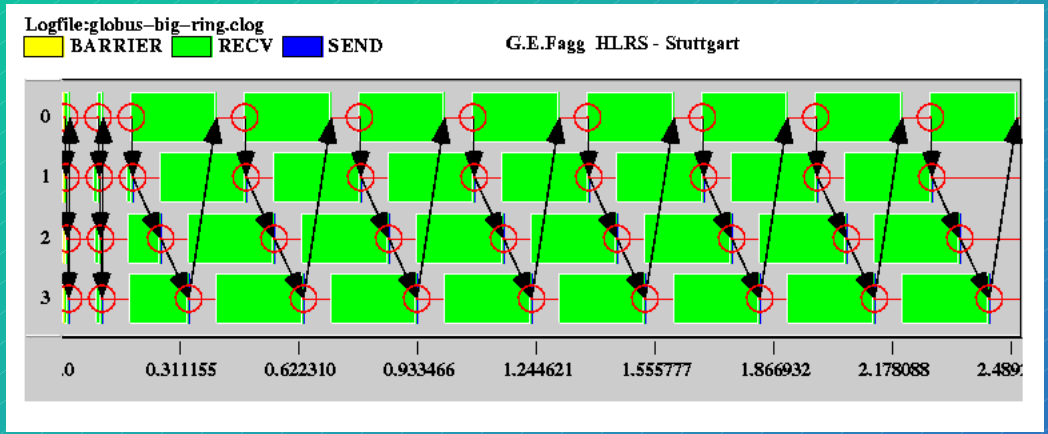

Linpack
Computation rate
1500.0
1000.0
500.0
0.0
MFlops/Sec
MPICH/ch_p4
MPICH-G
Legion MPI
0
2000
4000
6000
8000
10000
Matrix edge size N

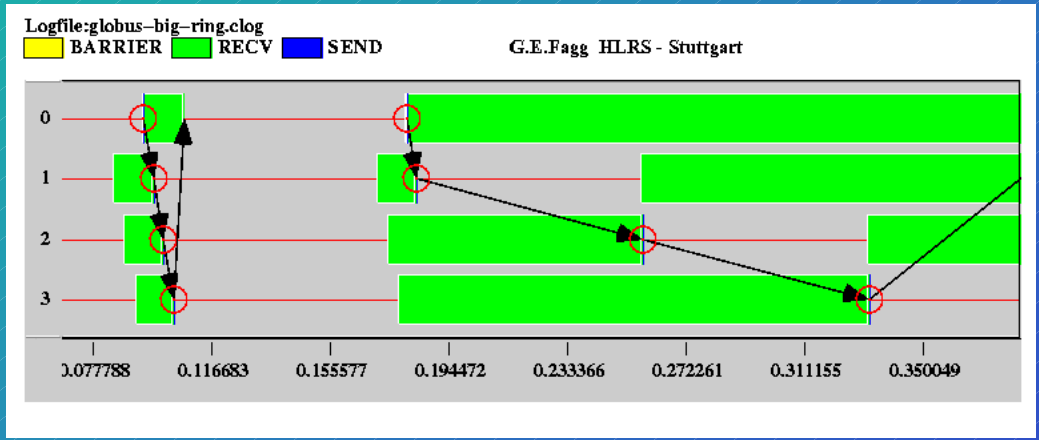
More on Globus MPI





More on Globus MPI





Conclusions

- Power users needs:
  - Cross-site authentication (currently available)
  - Cross-site scheduling (Available, inefficient)
  - Global file access (Not currently supported)

Conclusions (cont)

- Cross-site authentication
  - Both Globus and Legion can be integrated to existing infrastructure
- Cross-site scheduling
  - Both Globus and Legion can provide efficient schedules that filter down to the individual site and machine queuing systems
  - Performance issues are preventing users from using this functionality to run jobs on multiple MPPs (True Metacomputing)
- Global file access
  - Both Legion and Globus require moderate code alteration
  - Globus requires a file to be closed before another application can open it (no streaming pipeline)
  - Legion allows multiple copies with extensive caching for performance, but files must be imported into the Legion context space

Summary

- Both Globus and Legion provide benefits in terms of file/data management
- Both Globus and Legion provide some additional job scheduling functionality
- Both systems come with additional cost
  - Globus: A lot of extra sys. Admin. Support
  - Legion: More responsibilities for the user
- Reports from real world experiences will tell when, if ever, these systems are ready to be deployed in a production code environment.
- There are currently several groups continuing to work with both systems in a test environment