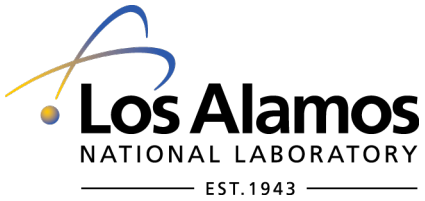
Open MPI State of the Union Community Meeting SC'08

Dave Montoya



Jeff Squyres

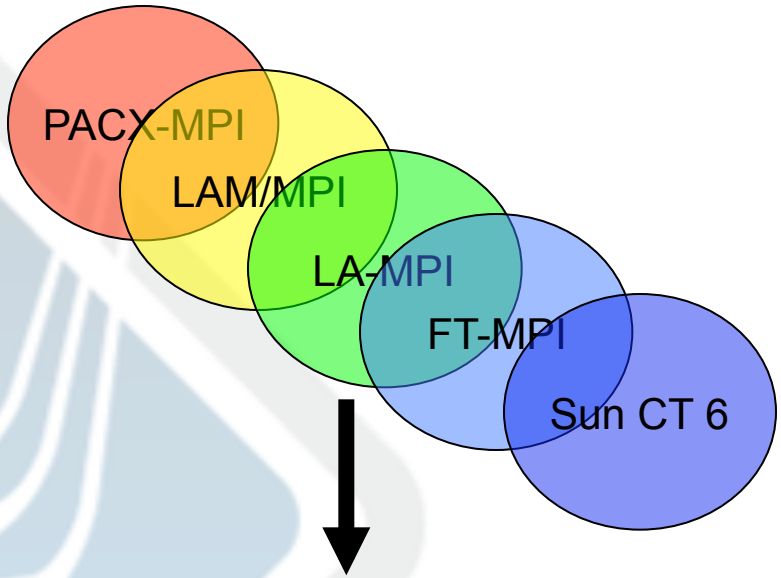
George Bosilca

Agenda

- Open MPI Project / Community
- Current Status: v1.3
- Los Alamos / Petaflop
- Roadmap
- Upcoming Challenges
- HPC Community Feedback

Open MPI Is…

- Evolution of several prior MPI's
- Open source project and community
  - Production quality
  - Vendor-friendly
  - Research- and academic-friendly
- All of MPI-1 / MPI-2

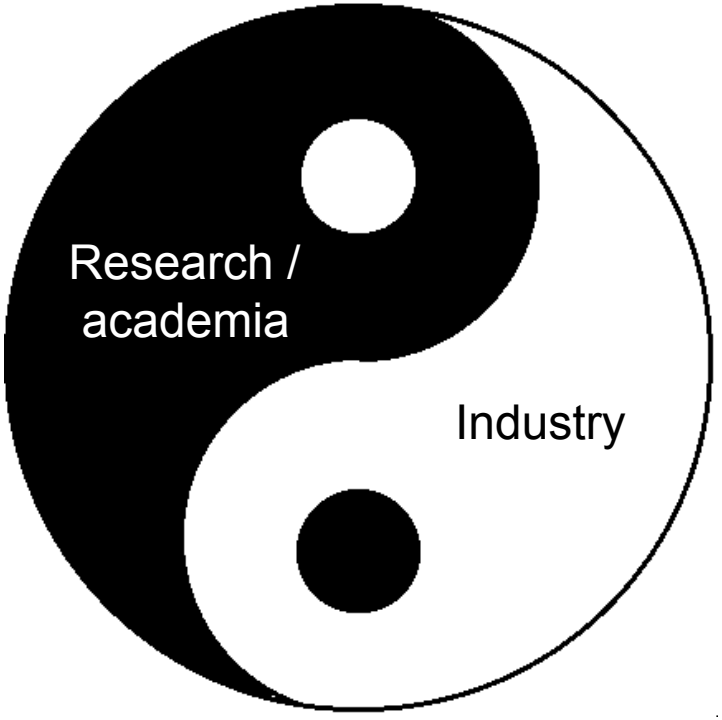


OPEN MPI

15 Members, 9 Contributors, 2 Partners

Why Does Open MPI Exist?

- Maximize all MPI expertise
  - Research / academia
  - Industry
  - …elsewhere
- HPC / MPI is not free
  - Need government, academic, and industry backing
- The sum is greater than the parts



Why Does Open MPI Work?

- How does the project stay together?
  - Different organizations
  - Different biases
  - Different goals
  - …these are exactly what make us strong
- Open MPI Project is like a marriage
  - It takes a lot of work
  - You need to find a good balance
  - But the end result can be really, really great

Current Status: v1.3

Open MPI v1.3

- Release Managers:
  - Brad Benton (IBM)
  - George Bosilca (UTK)
- Expected as soon as possible after SC'08!

Open MPI v1.3

- First planning meeting: Feb 2007
- Aiming for beginning of 2008
- Possible features (subject to change / light grey indicates "unlikely")
  - ConnectX XRC support
  - End-to-end data reliability
  - More scalability improvements
  - More compiler, run-time environment support
  - Fine-grained processor affinity control

Open MPI v1.3

- MPI 2.1 compliant
- The notifier framework
- Documentation (?!)
- More architectures, more OSes and more batch schedulers, and more compilers
- Thread safety:
  - Support included for some devices
  - Only the point-to-point support have been tested
- MPI_REAL16 and MPI_COMPLEX32

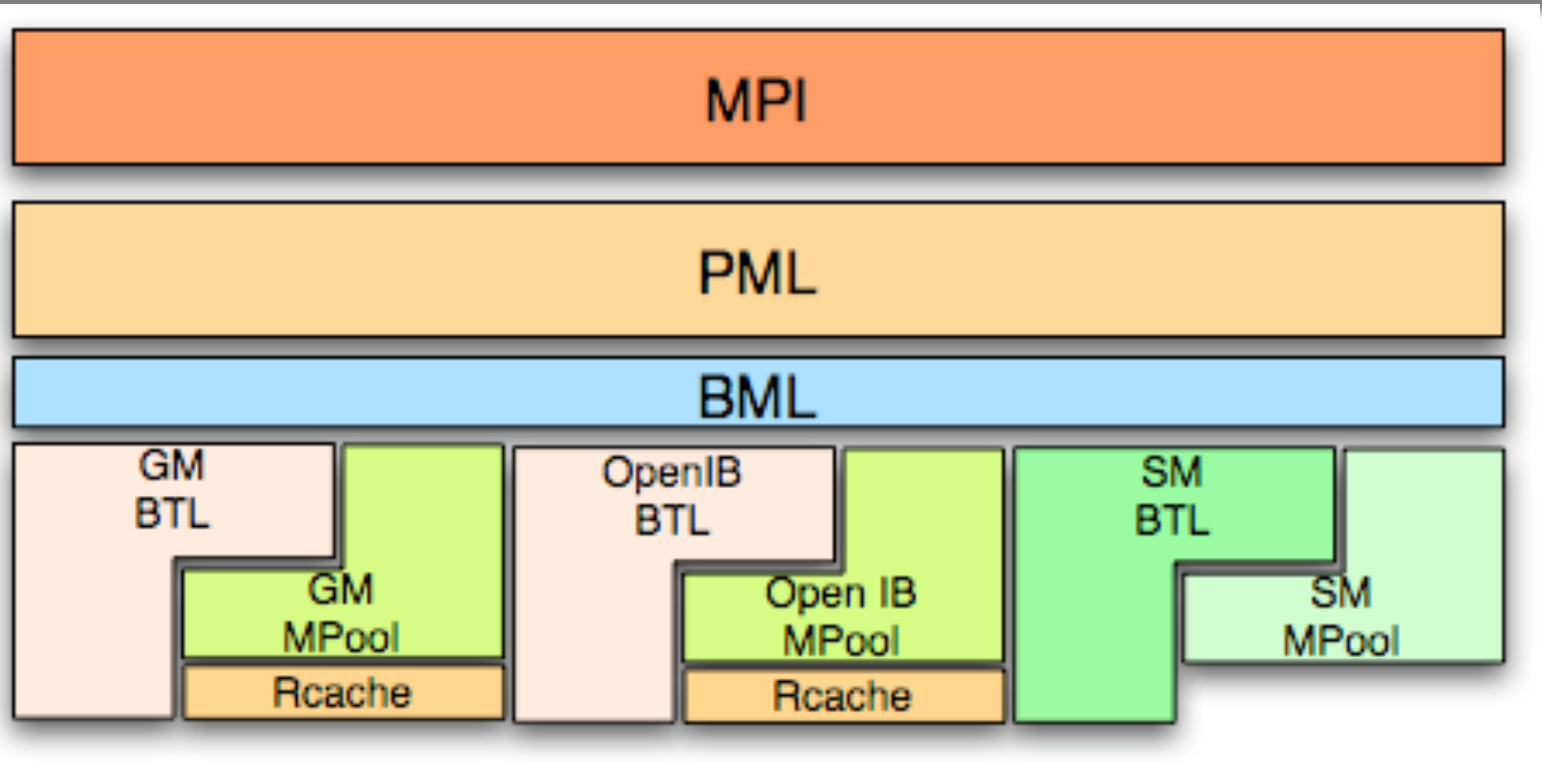
Open MPI v1.3

- Many (many) improvement to the MPI C++ bindings
- Valgrind support (memchecker)
- Update ROMIO to the version from MPICH2 1.0.7
- Condensed error messages
- Many little improvements

Open MPI v1.3

- Scalability
  - Keep the same on-demand connection setup as prior version
  - Decrease the memory footprint
    - Sparse groups and communicators
    - Less data in the business card
  - And a lot of improvements in the Open MPI RTE (our runtime system).

Open MPI v1.3


MPI
PML
BML
GM
BTL
GM
MPool
Rcache
OpenIB
BTL
Open IB
MPool
Rcache
SM
BTL
SM
MPool

Open MPI v1.3

- Point-to-point Message Layer (PML)
  - Improved latency
  - Better adaptive algorithms for multi-rail support
  - Smaller memory footprint
- Collective Communications
  - More algorithms, more performance
  - Special shared memory collective
  - Hierarchical Collective active by default

Open MPI v1.3 (OpenFabrics)

- Many performance enhancements
- Added iWARP support
- "Bucket" SRQ support
- XRC support
- Message coalescing
- Asynchronous error events
- Automatic Path Migration (APM)
- Improved processor / port binding
- uDAPL enhancements
  - Multi-rail support
  - Subnet checking
  - Interface include/ exclude capabilities

Low Level Devices (BTL) Status

Network	Dynamic Processes	Threading support
Self		
Shared Memory		
TCP		
Myrinet (MX)		
Myrinet (GM)		
Infiniband (openib)		
Infiniband (ofud)		
Elan		
Sicortex		
Portals		
uDAPL		
SCTP		

- All BTL devices support MPI 1 (pt-to-pt) and MPI 2 (RDMA) communications
- All devices support PERUSE
- Table on left shows BTL dynamic / threading status

NOTE: MTL components do not support threading
  - Use BTL equiv. (if available)
  - MX, Portals, PSM

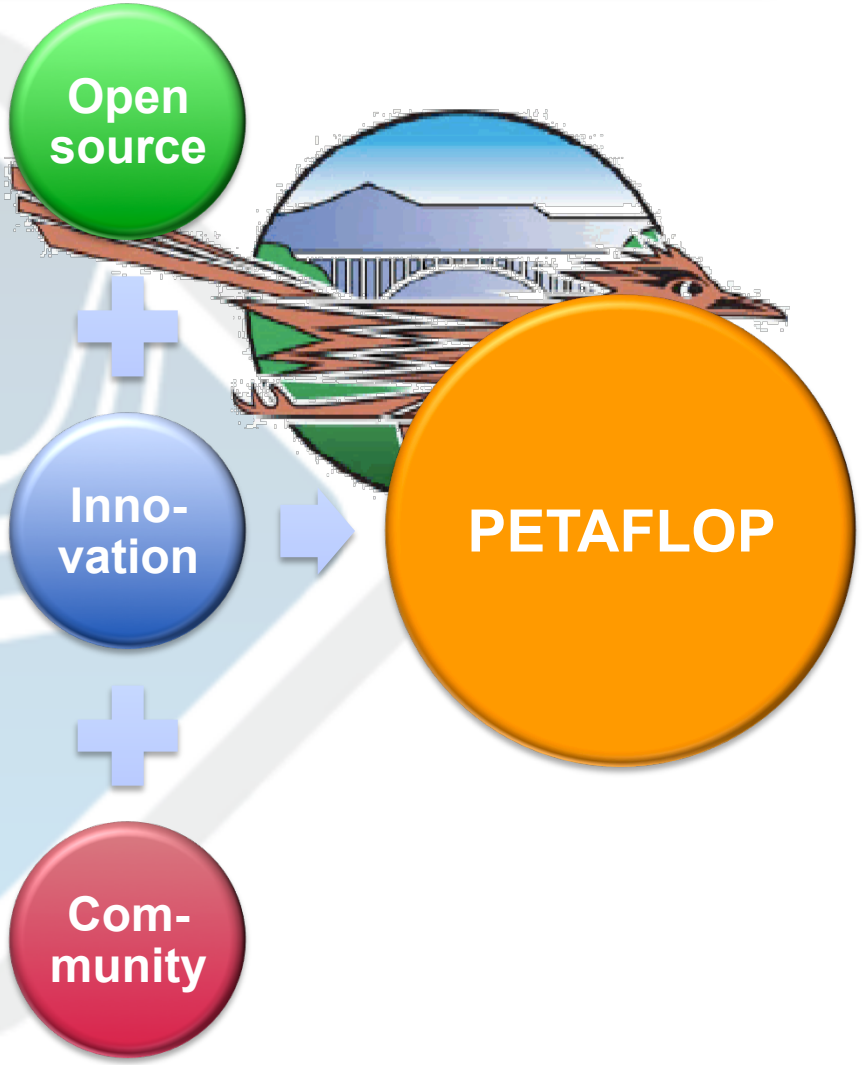
Open MPI v1.3

- Fault Tolerance
  - Coordinated checkpoint/restart
  - Support BLCR and self
  - Able to handle real process migration (i.e. change the network during the migration)
    - MX, IB, TCP, SM, self
  - Improved Message Logging (under 5% overhead).

Los Alamos + Scalability = Petaflop

Petaflop!!

- Los Alamos Road Runner
- #1 on Nov. 2008 Top500
  - 1.1 petaflops
- Powered by Open MPI
  - Significant community achievement

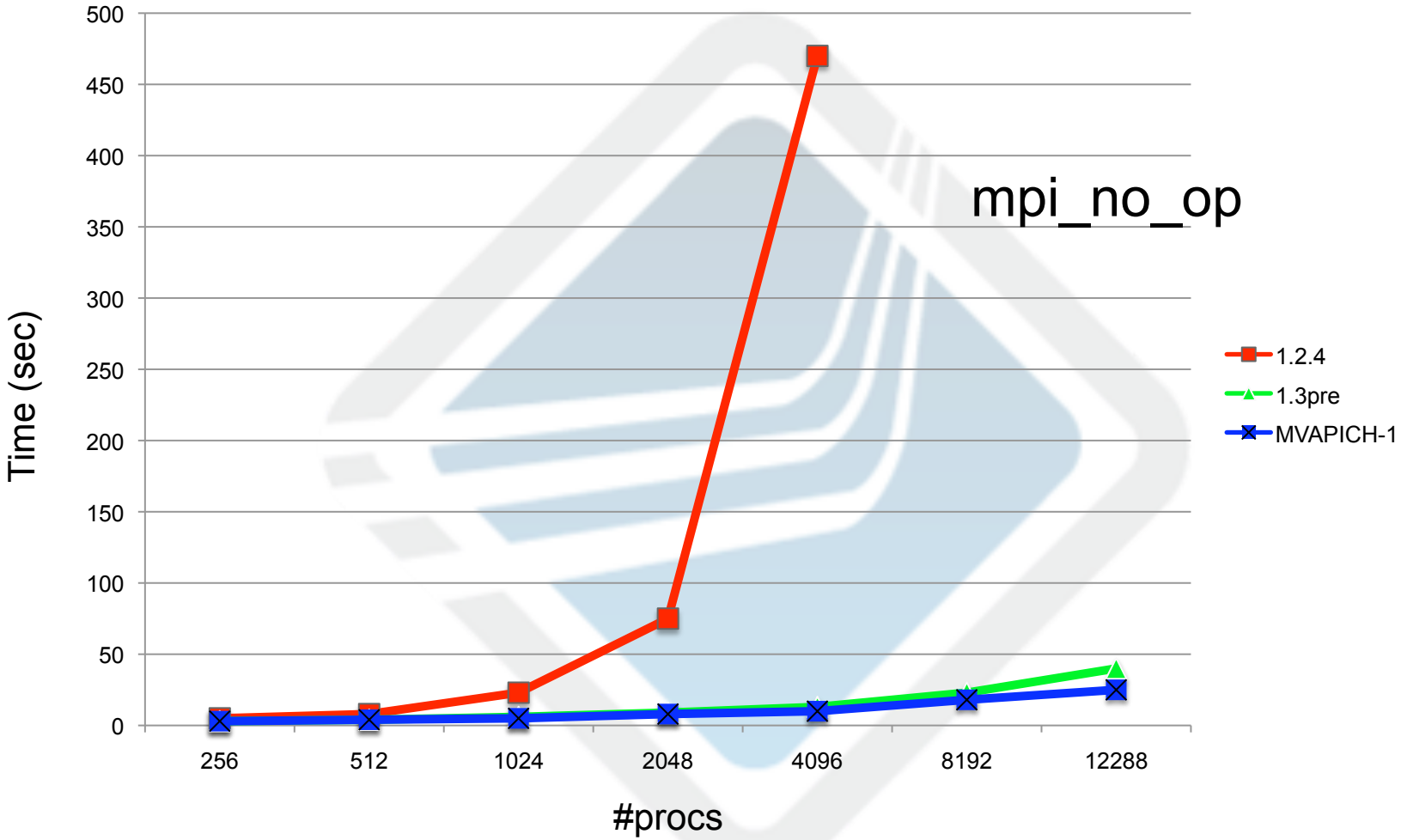


OMPI 1.3: Lean, Mean OMPI Machine

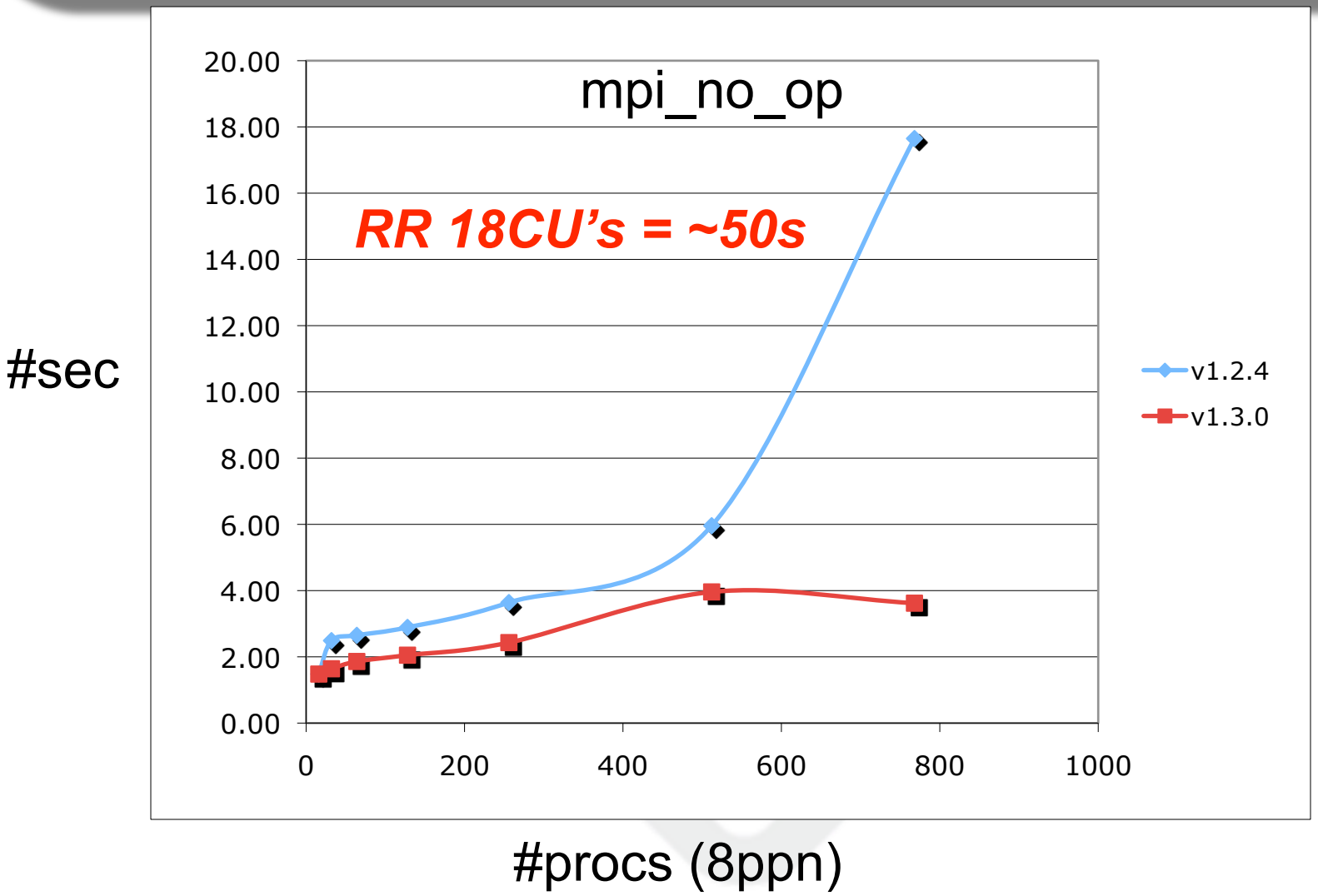
We Shared Your Pain

- Scalability
  - Reduce launch times by order of magnitude
  - Reliable cleanup, robustness
- User features
  - Simplify & combine frequently used multiple params into one option
  - Extend usability based on feedback
  - Better, easier debug and error messages
- Maintainability
  - Cleanup, simplify program flow
  - Remove everything not *required* for OMPI

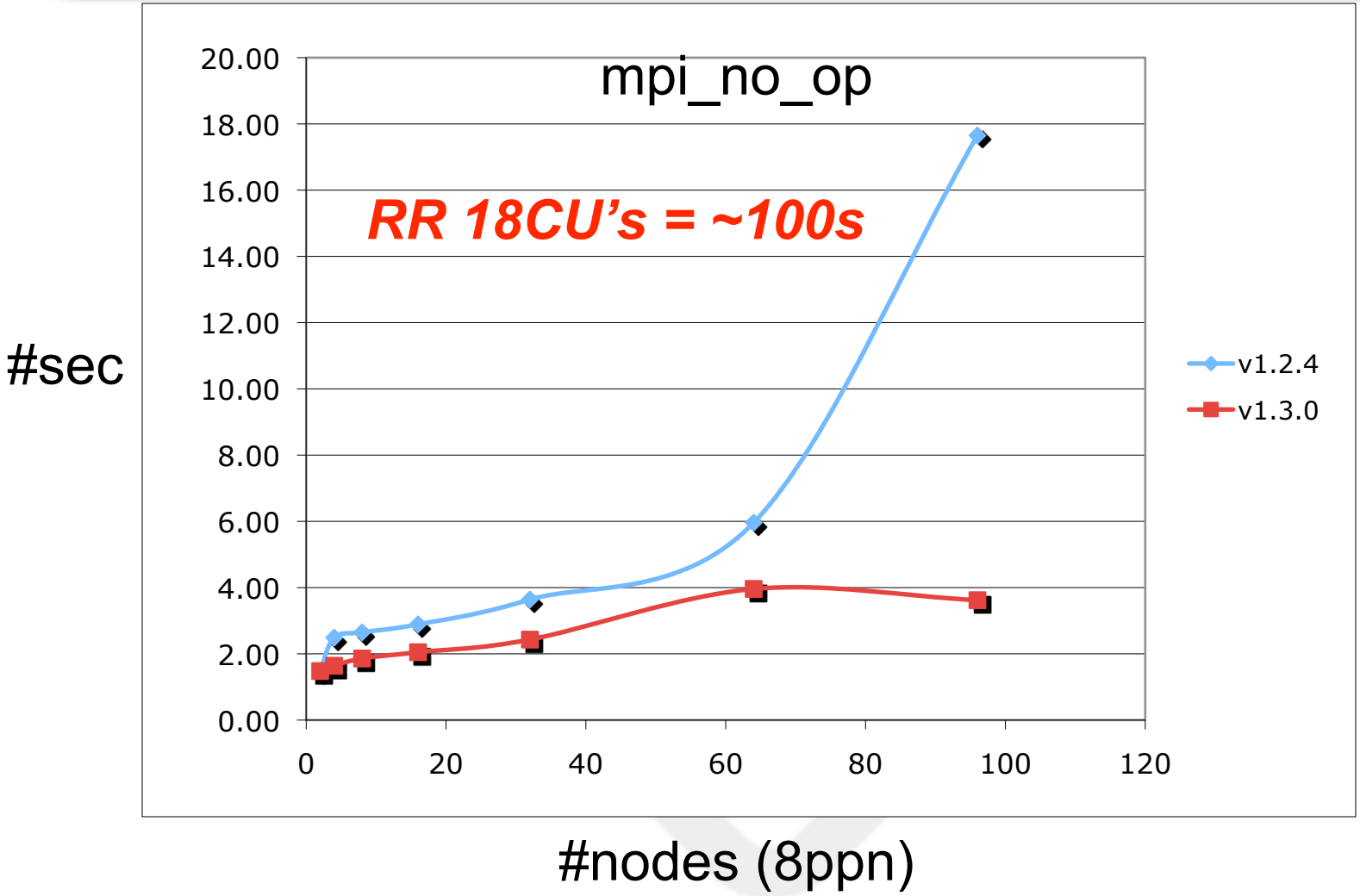
How Did We Do?


mpi_no_op
Time (sec)
500
450
400
350
300
250
200
150
100
50
0
256
512
1024
2048
4096
8192
12288
#procs
1.2.4
1.3pre
MVAPICH-1

Old vs. New (rrz)



Old vs. New (rrz)



Runtime New Features

- New mapping algorithms
  - Sequential
  - Loadbalanced
  - Rank/slot direct mapping of ranks to sockets and cores
- Resource Definition
  - Clarified hostfile, -host, RM-allocation interactions
  - Relative node indexing

Runtime New Features

- --display-map
  - Displays map of nodes and ranks
- --display-allocation
  - What nodes have been allocated to job
- --leave-session-attached
  - Maintains connection to daemons without ORTE diagnostic output

Runtime New Features

- Routed out-of-band communications
  - MPI remains point-to-point
  - OOB routes all messages through local daemon
  - Connection count on node
    - 1.2.x: #procs_on_node x #procs_in_job
    - 1.3: #nodes_in_job
    - Example: 1024 procs on 256 nodes (4ppn)
      - 4096 connections => 256 connections
- -mca ompi_show_mca_param
  - Shows what mca params are being seen by the MPI ranks themselves, and where they were set

Roadmap

Possible Future Features

- **BIG** disclaimer
  - We are in the planning phase of v1.4 only
  - Features discussed here are *possible*
  - Nothing has been fully decided yet
- Not seeing something you want?
  - Come join us!

Possible Future Features

- Run-time parameter usability options
  - So many parameters, so little time…
  - Ability to sysadmin “lock” parameter values
  - Spelling checks, validity checks
- Run-time system improvements
  - Next generation launcher
  - Integration with other run-time system

Possible Future Features

- More processor and memory affinity support
  - Usability features (a la Sun ClusterTools 6)
  - Automatic mappings, cartography discovery
  - “Topology awareness”
  - …? (manycore / networking kinds of issues)
- [More] Shared memory improvements
  - Allocation sizes, sharing
  - Scalability to manycore

Possible Future Features

- I/O redirection features
  - Line-by-line tagging
  - Output multiplexing
  - “Screen”-like features
- Error message notification flexibility
  - Communicate with network / cluster monitoring systems
  - Multiple degrees of warnings / errors

Possible Future Features

- OpenFabrics
  - Asynchronous progress for long messages
  - IBCM support (scalability)
  - Investigate UD (e.g., collectives)
  - Combine shared memory + verbs for on-host communications
  - Relaxed PCIe ordering

Possible Future Features

- Blocking progress
- MPI connectivity map
- Refresh included software
  - Libevent, ROMIO, …
- Separate BTL's into standalone entities
  - Build something other than MPI…?
- Progress thread / asynchronous progress

Upcoming Challenges

Challenges

- Fault Tolerance
  - Uncoordinated + Message Logging
  - Similar with FT-MPI approach
    - Or try to stay in sync with the MPI Forum
- Scalability
  - At the runtime level
  - And at the MPI level

Challenges

- Collective Communications
  - Deal with the size
  - Take advantage of the physical topology
  - Figure out when to switch between collective algorithms
- Point-to-point
  - Even more performance
  - And scalability (shared memory and all)

HPC Community Feedback

What do You Want From MPI?

- MPI-2.1 is complete
  - Merged MPI-1 and MPI-2 documents (yay!)
  - $22 printed books (586 pages!), HLRS booth #1353
- The MPI Forum needs your help!
  - What do you want to see in MPI-3.0?
  - What do you ***not*** want to see in MPI-3.0?

What do You Want From Open MPI?

Ernest Hemingway:

When people talk, listen completely.
Most people never listen.

(we're listening; you talk now)

What do You Want From Open MPI?

Franklin D. Roosevelt:

Be sincere; be brief; be seated.

(we're listening; you talk now)

How Important Is…

- Thread safety
  - Multiple threads making simultaneous MPI calls
- Parallel I/O
  - Working with parallel file systems
- Dynamic processes
  - Spawn, connect / accept
- One-sided operations
  - Put, get, accumulate

Come Join Us!

http://www.open-mpi.org/