MPI Has Failed Now What ?

Patrick Geoffray
Senior Software Architect
patrick@myri.com

14 September 2008
CCGSC – Flat Rock, NC

Failed ?!? Yeah, Right.

- MPI is widely used in HPC.
  - Millions of lines of code.
  - Supported by all vendors.
- MPI has reduced the cost of HPC.
  - Applications are easier to port to new machines.
  - ISVs maintain single code base.
  - Larger pool of human expertise, tools.
  - Everybody can write parallel codes, even physicists.
- MPI is a good programming interface.
  - Implicitly express locality.
  - Does not rely on compilers.
  - Simple error handling.
  - Support communication/computation overlap.

The Devil's Advocate

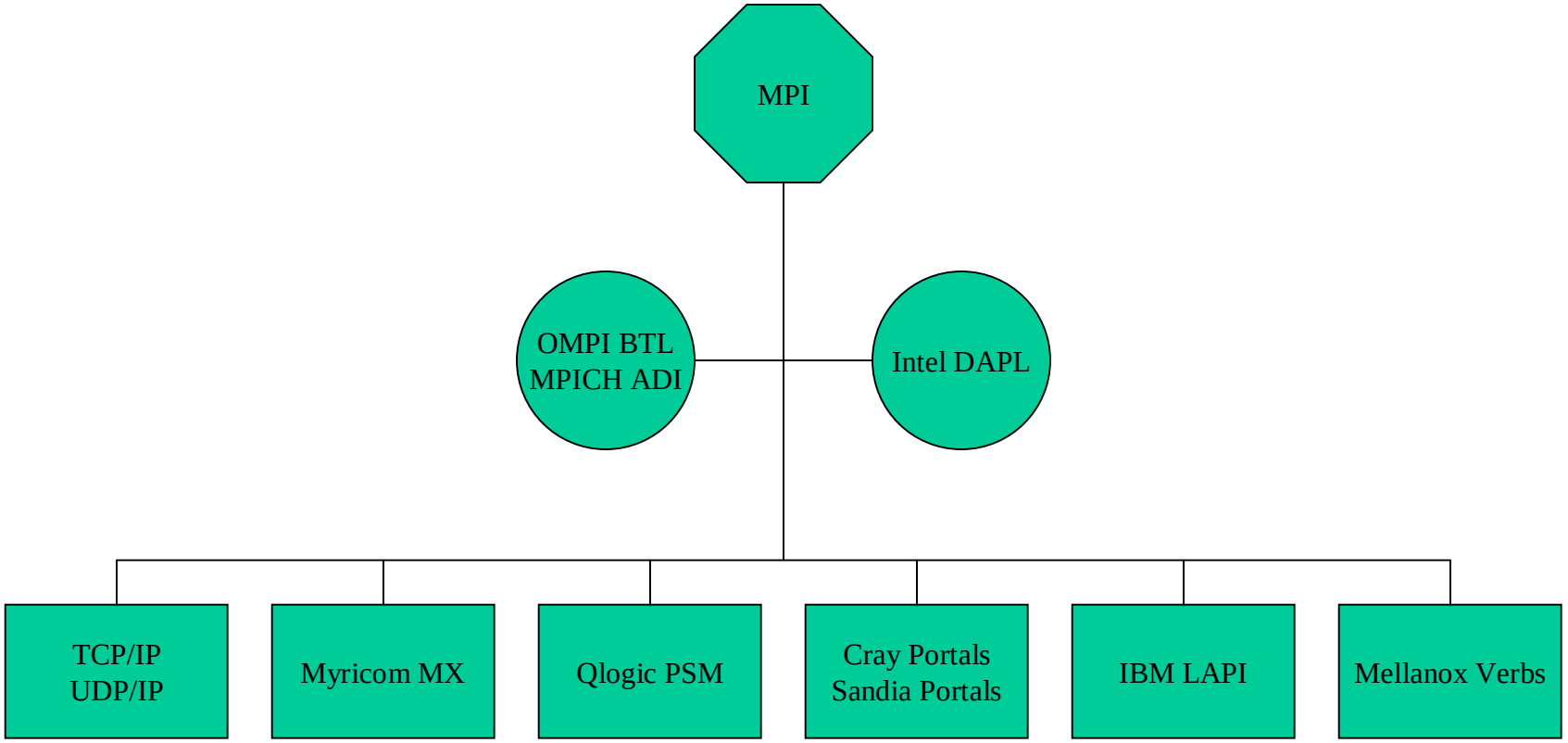
- MPI is not portable.
  - Look for "undefined" or "implementation dependent" in the spec.
  - False buffering assumption can deadlock valid MPI codes.
- MPI is not efficient.
  - Force matching and/or copy even when locality is not important.
  - Unexpected messages, matching in linear time, MPI-2 RMA.
- MPI is not scalable.
  - Everybody knows about everybody else.
- MPI is not fault-tolerant.
  - Spec assumes reliable message transmission, no async errors.
  - *Après moi le deluge* (After me comes the floods, Louis XV).
- MPI is in the HPC ghetto.
  - Real world uses Socket.

MPI Cannot Be Fixed

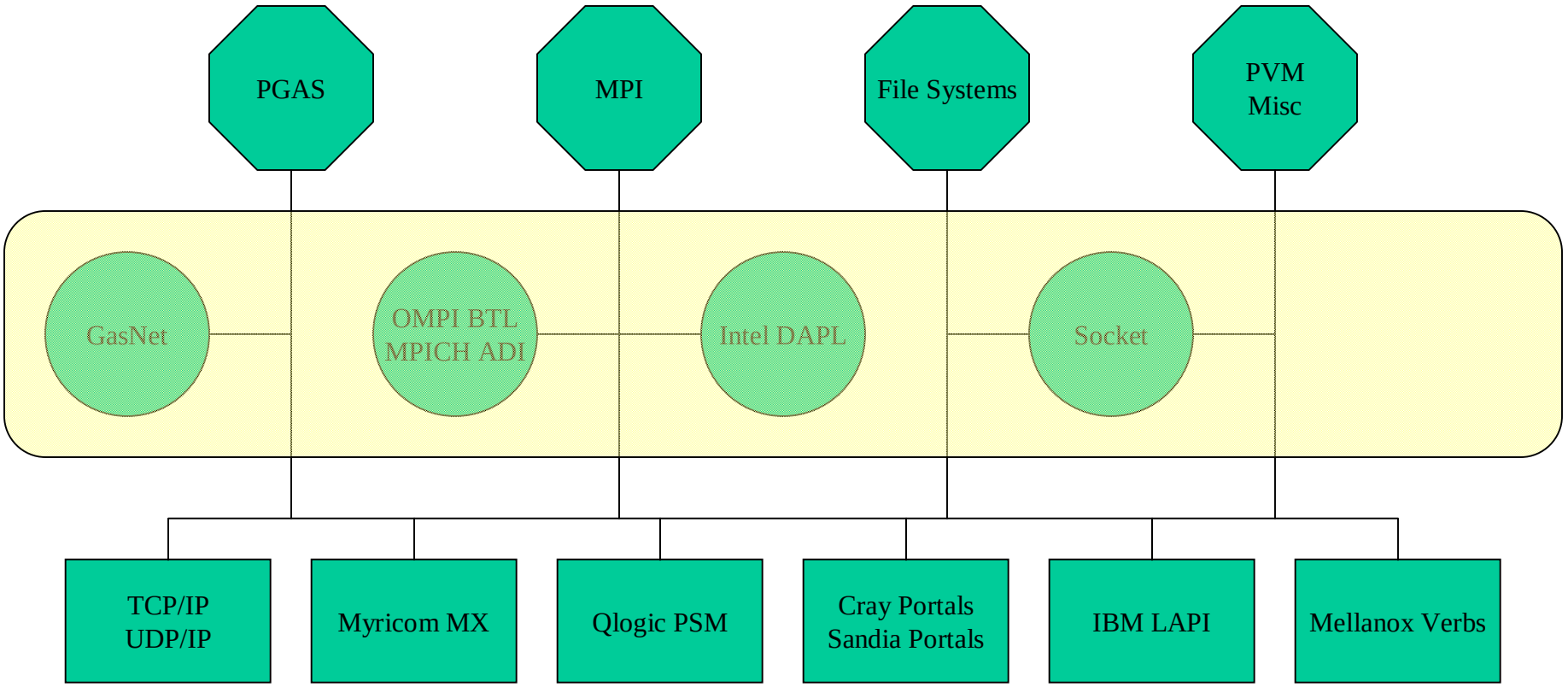
- MPI Forum can only add things, too much inertia to really reform the Interface.
- Most problems are semantic (ex. matching) or interface-based (ex. error handling).
  - Require major changes to the foundations of the Interface.
- *Sub-setting* is not a solution.
  - Narrow down some part of the Interface (for example removing ANY_SENDER).
  - Mostly targeted at performance.
- FT-MPI and other proposals are not practical.

➢ **Define a small, well-defined, fault-tolerant, scalable, efficient interface *below* MPI.**

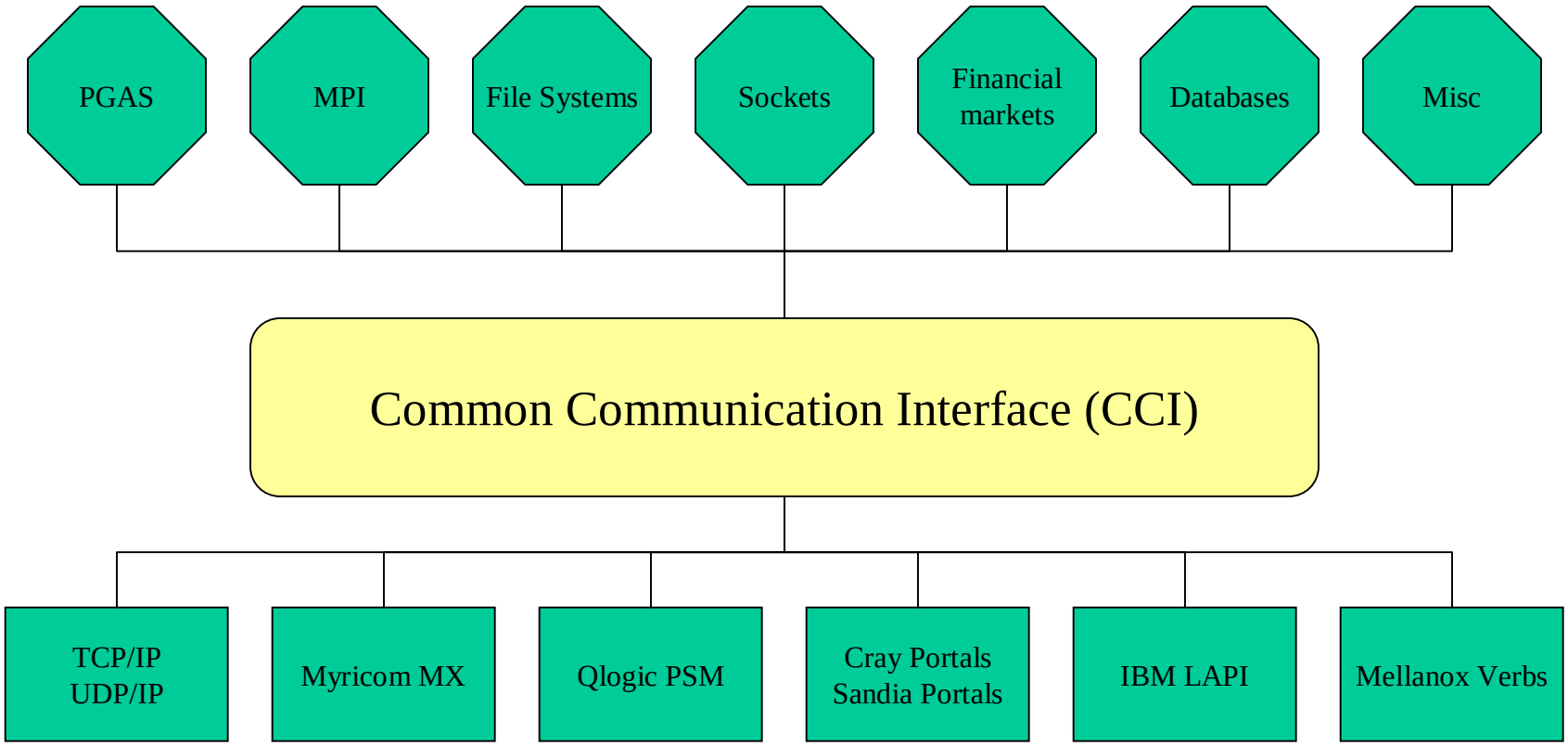
The MPI World



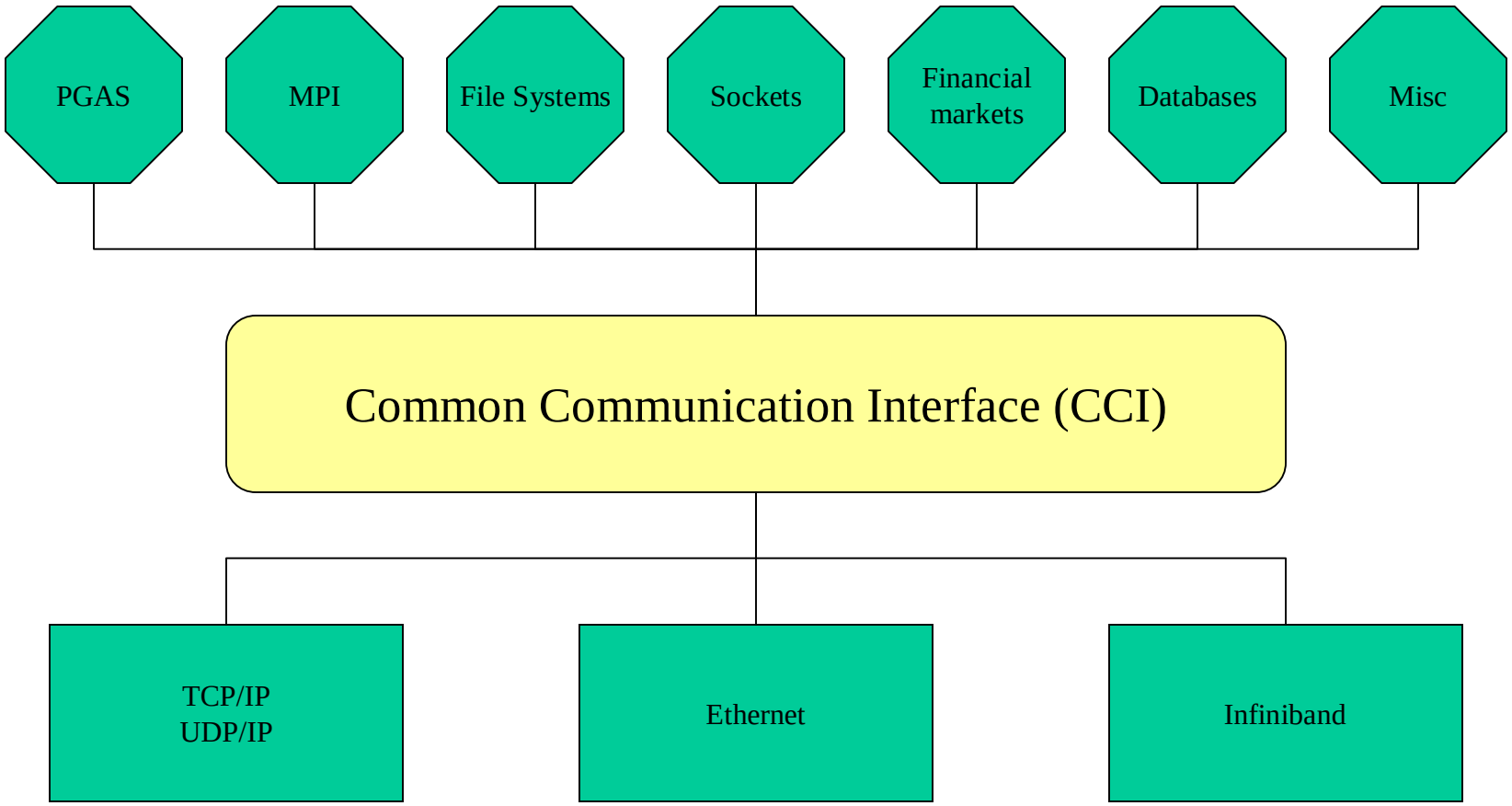
The HPC Ghetto



The Unified World



The Refined Unified World



Common Communication Interface (CCI)

- Actives Messages and RMA operations.
- Operations are always non-blocking.
  - Incentive for communication/computation overlap.
- No assumed order on the wire.
  - Allow for multi-rails and/or adaptive routing.
- Asynchronous progress.
  - Threads or event-driven runtime.
  - Explicit progress function for backup.
- No per-connection resources (no QPs), but not connection-less (state).
- Local completion == Remote completion.
  - All error notifications are synchronous.
- Callbacks on operations' state transitions.

Active Messages

- Semantics:
  - Always buffering on send side.
  - Separate Header and Data.
  - Message size limited to MTU (no order on wire).
  - Handler called on receive side, asynchronously or in progress function.
  - Access to Header and Data buffers limited to handler lifetime.
  - Possibility to “borrow” Data buffer for zero-copy deferred access.

- Implementation benefits:
  - Simple MTU-sized send and receive rings.
    - Implicit flow-control.
  - Handler can be executed on a host, a NIC, a GPU.

Remote Memory Access Operations

- Semantics:
  - One-sided operations.
  - Never buffering on local side.
  - Explicit memory regions management.
    - Virtual memory regions for custom mappings.
  - Express dependencies between groups of operations (including Active Messages).
  - Regular non-contiguous access (n-dimension stride) local/remote.
  - Atomic operations.

- Implementation benefits:
  - Can be implemented on top of Active Messages.
  - Dependencies (order) can be enforced on remote side.
  - Virtual regions piggyback IOMMU for global memory allocator.

Status

- Technical spec in still in early development.
- CCI is not a Public Forum.
  - Right now, a more or less formal group of experts.
  - Public input later in the process.
- CCI is gaining momentum.
  - A number of vendors have joined or expressed great interest.
  - Some middlewares maintainers cried of joy.
- Proof is in the pudding.
  - Early implementations of CCI over common low-level vendor interfaces.
  - Early ports of various middlewares on top of CCI.
- Ethernet unification is a great drive.

Thank you! Questions?