Exascale Process Management Interface

Ralph Castain
Intel Corporation
rhc@open-mpi.org

Joshua S. Ladd
Mellanox Technologies Inc.
joshual@mellanox.com

Artem Y. Polyakov
Mellanox Technologies Inc.
artemp@mellanox.com

David Bigagli
SchedMD
david@schedmd.com

Gary Brown
Adaptive Computing
gbrown@adaptivecomputing.com

PMIx – PMI exascale

Collaborative open source effort led by
Intel, Mellanox Technologies, and Adaptive Computing.

New collaborators are most welcome!

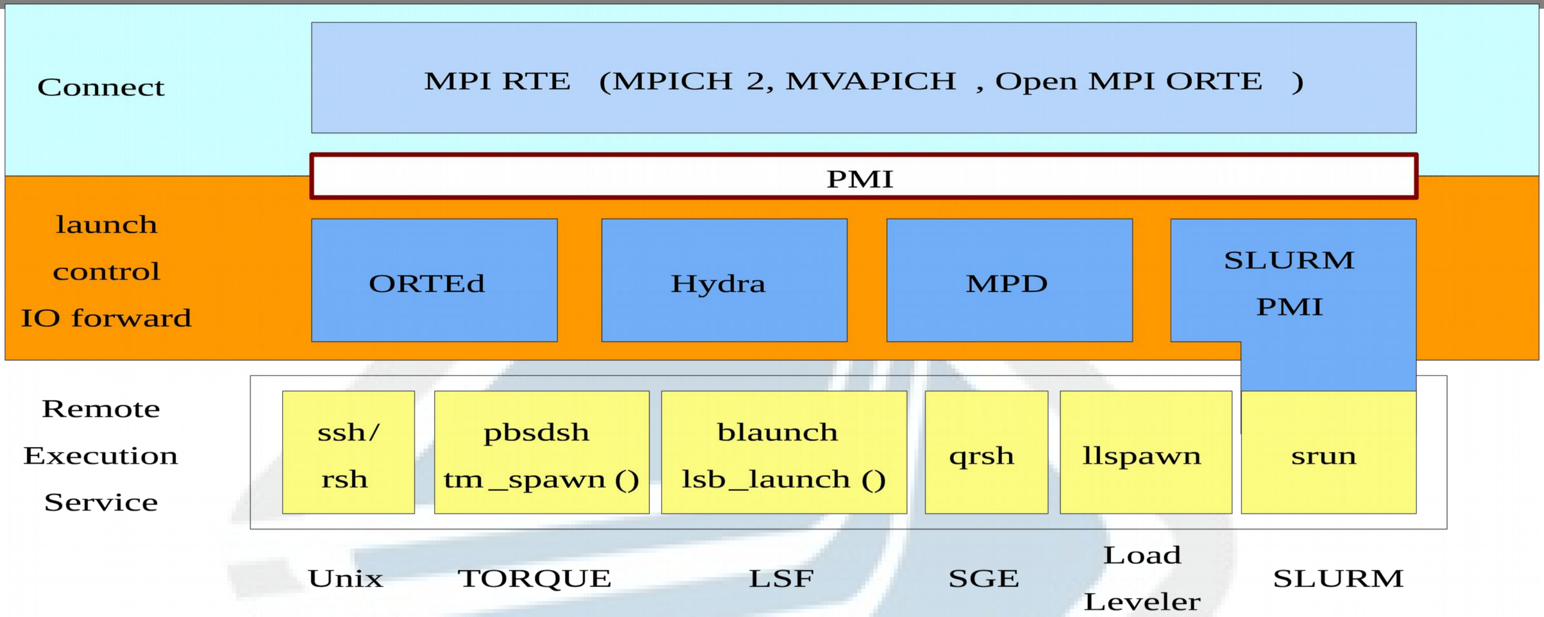
intel®

Mellanox®
TECHNOLOGIES

SchedMD

Adaptive
COMPUTING

Process Management Interface – PMI



- PMI is most commonly utilized to bootstrap MPI processes.
- Typically, MPI processes “put” data into the KVS data base that is intended to be shared with all other MPI processes, a collective operation that is a logical **allgather** synchronizes the database.
- PMI enables Resource Managers (RMs) to use their infrastructure to implement advanced support for MPI application acting like RTE daemons.
- SLURM supports both PMI-1/PMI-2 (http://slurm.schedmd.com/mpi_guide.html)

PMIx – PMI exascale

(What and Why)

- **What is it?**
  - Extended Process Management Interface.
- **Why?**
  - MPI/OSHMEM job launch time is a **hot topic**!
  - Extreme-scale system requirements: 30 second job launch time for $O(10^6)$ MPI processes.
  - Scaling studies have illuminated many limitations of current PMI-1/PMI-2 interfaces at **extreme scale**.
  - Tight integration with Resource Managers can **drastically reduce** the amount of data that needs to be exchanged during MPI_Init.

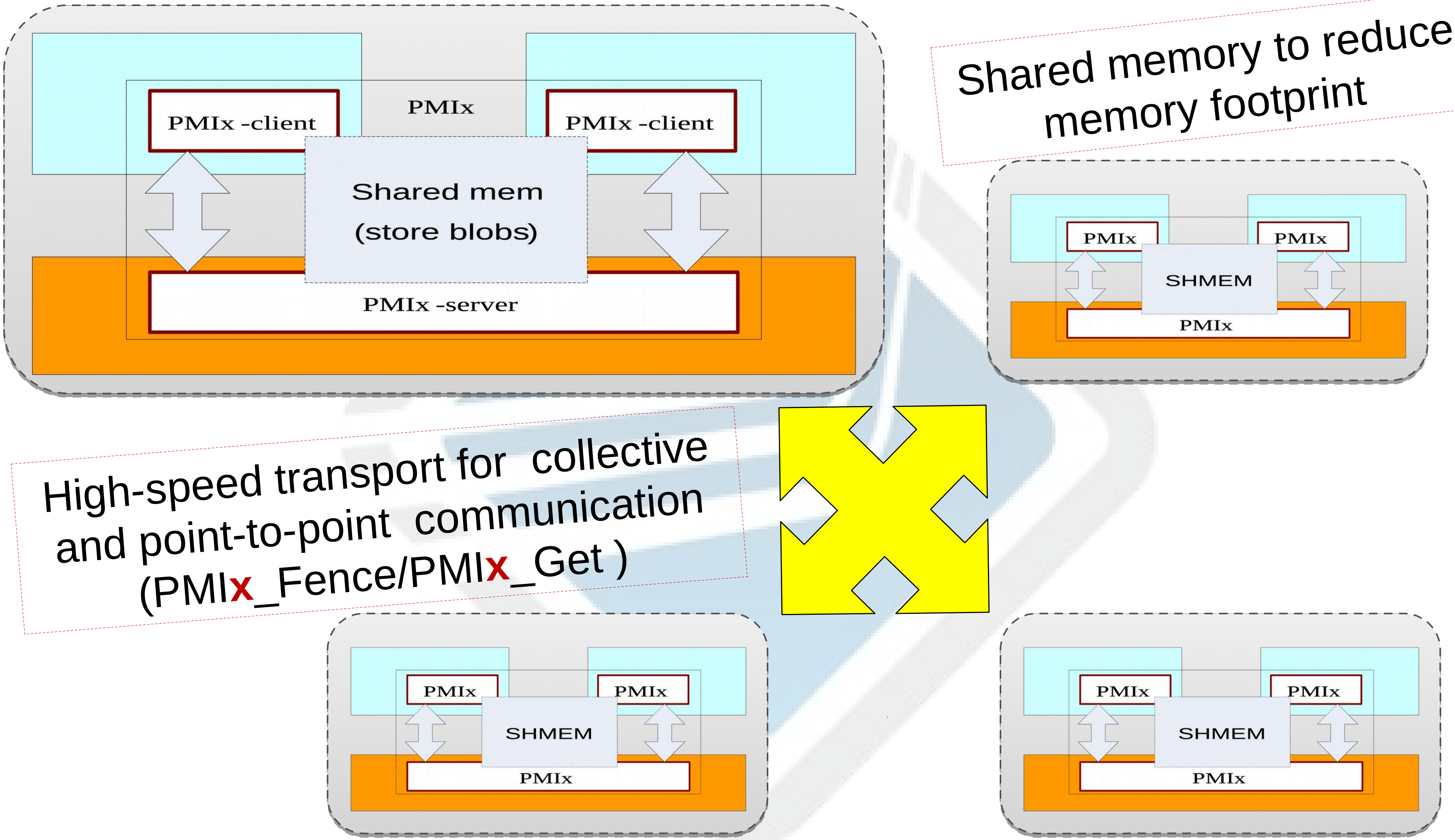
PMI**x** is a new process management interface
that has been designed to address these limitations

PMIx – PMI exascale

(Technical Goals)

- Reduce the **memory footprint** from $O(N)$ to $O(1)$ by leveraging *shared memory* and *distributed databases*.
- Reduce the **volume** of data **exchanged** in collective operations with *scoping hints*.
- Provide the ability to **overlap** communication with computation with *non-blocking* collectives and get operations.
- Support both *collective* communication modes of **data exchange** and *point-to-point* "direct" data retrieval.
- Reduce the amount of local messages exchanged between application processes and RTE daemons (**many-core nodes**).
- Use high-speed **HPC interconnects available on the system** for the data exchange.
- Extend "Application – Resource Manager" interface to support fault-tolerance and energy-efficiency requirements.

PMIx implementation architecture



Shared memory to reduce memory footprint

High-speed transport for collective and point-to-point communication (PMIx_Fence/PMIx_Get)

PMIx v1.0 features

- Data scoping with 3 levels of locality: *local*, *remote*, *global*.
- Communication scoping: PMIx_Fence under arbitrary subset of processes.
- Full support for *point-to-point* "direct" data retrieval well suited for applications with sparse communication graphs.
- Full support for non-blocking operations.
- Support for “binary blobs”: PMIx client retrieves process data only once as one chunk reducing intra-node exchanges and encoding/decoding overhead.
- Basic support for MPI dynamic process management;

PMIx v2.0 features

Performance enhancements:

- One instance of database per node with "zero-message" data access using shared-memory.
- Distributed database for storing Key-Values.
- Enhanced support for collective operations.

Functional enhancements:

- Extended support for dynamic allocation and process management suitable for other HPC paradigms (not MPI-only.)
- Power management interface to RMs.
- File positioning service.
- Event notification service enabling fault tolerant-aware applications.
- Fabric QoS and security controls.

SLURM PMIx plugin

PMIx support in SLURM

- Implemented as a new MPI plugin called "pmix".
- To use it:

a) either set as a command line command line parameter:

```
$ srun –mpi=pmix ./a.out
```

b) or set PMIx plugin as the default in slurm.conf file:

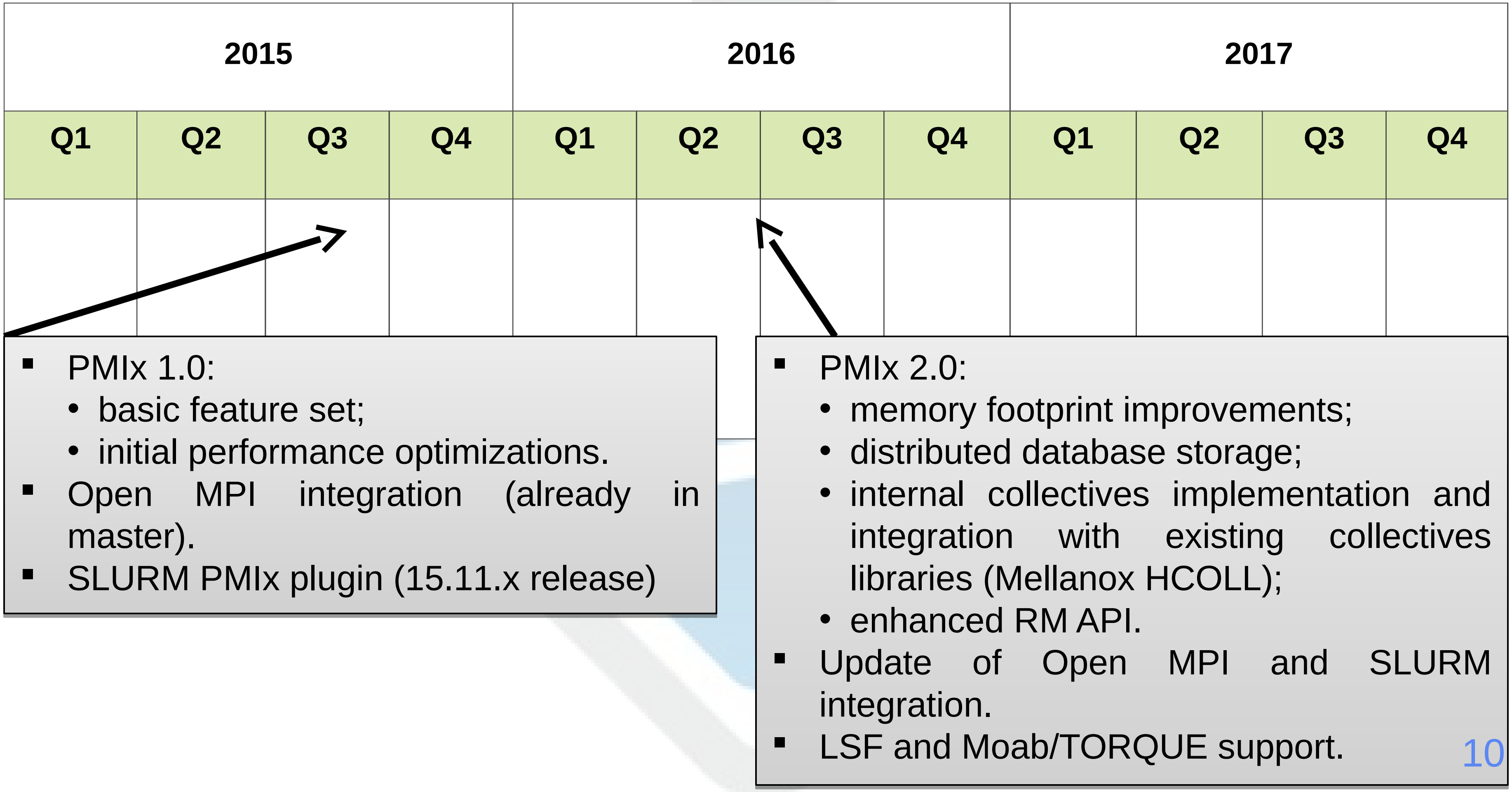
```
MpiDefault = pmix
```

- Development version of the plugin is available on github:

https://github.com/artpol84/slurm/tree/pmix-step2

- Beta version of PMIx plugin will be available in the next SLURM major release (15.11.x) at SC 2015.

PMIx development timeline



2015				2016				2017			
Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4

- PMIx 1.0:
  - basic feature set;
  - initial performance optimizations.
- Open MPI integration (already in master).
- SLURM PMIx plugin (15.11.x release)

- PMIx 2.0:
  - memory footprint improvements;
  - distributed database storage;
  - internal collectives implementation and integration with existing collectives libraries (Mellanox HCOLL);
  - enhanced RM API.
- Update of Open MPI and SLURM integration.
- LSF and Moab/TORQUE support.

Contribute or Follow Along!

- **Project:** **https://www.open-mpi.org/projects/pmix/**
- **Code:** **https://github.com/open-mpi/pmix**

Contributions are welcomed!