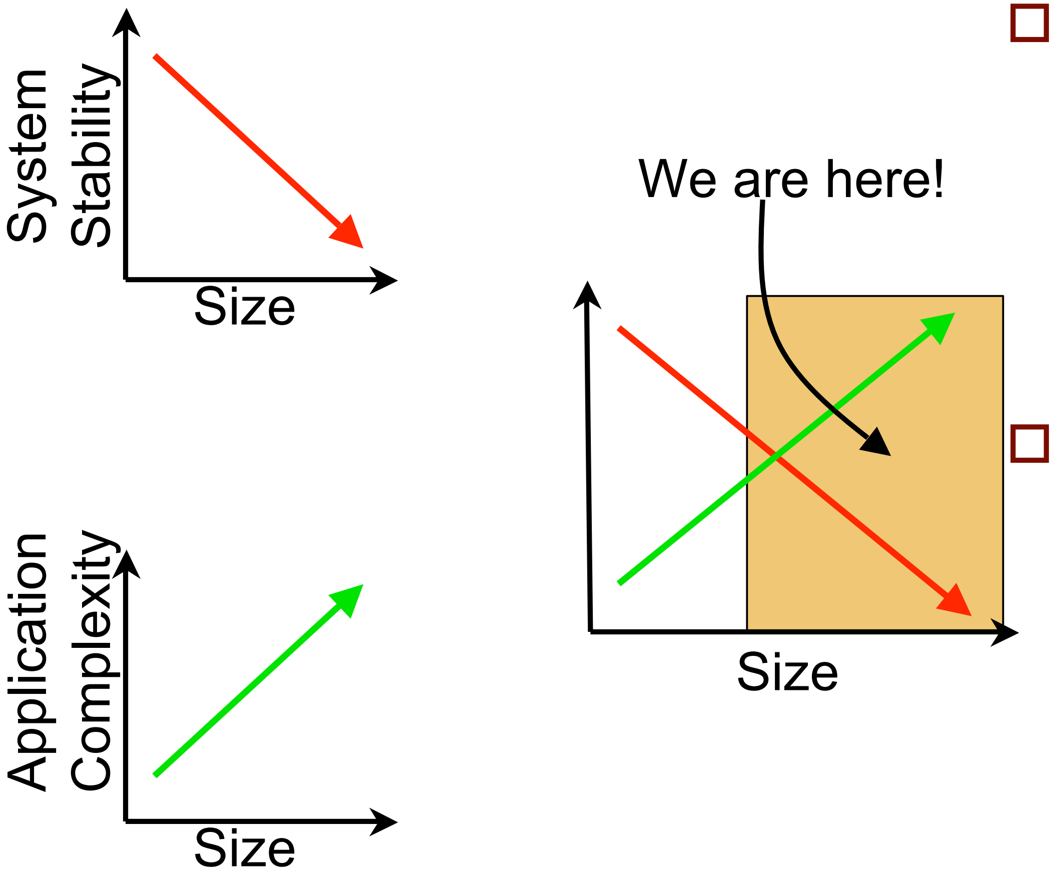
Dealing with Disaster:
Fault Tolerance in Open MPI

Josh Hursey
Open Systems Laboratory
Indiana University
jjhursey@open-mpi.org

What is there to worry about?



- HPC Systems
  - Growing in size and complexity
  - Increased frequency of component failure
- HPC Applications
  - Running longer as data sets become more complex
  - Scaling to higher degrees due to algorithmic advances.

What can we do?

- Lazy Optimism
  - Do nothing, hope for the best.
  - Scale back job submissions

- It is the Systems fault!
  The System should deal with it for me!
  - Wait for it to provide a stable, transparently fault tolerant solution.

What can we “really” do?

- Do it yourself failure handling
  - **Step 1:** Take legacy code base
  - **Step 2:** Learn about fault tolerance techniques
  - **Step 3:** Adapt code base for a set of fault scenarios
  - **Step 4:** Test and debug…
- Depend upon fault tolerance libraries and support services
  - Link with checkpoint/restart libraries
  - Use fault tolerant communication libraries

Fault Tolerance in MPI

- MPI is the *de facto* standard message passing environment for HPC applications.
  - MPI-1 and MPI-2 standards: **`http://www.mpi-forum.org/`**
  - Many implementations available

- MPI positioned to have unique knowledge of the distributed job state
  - Manage all inter-process communication
    - Must be a good steward of all data communicated
  - Detect and respond to process and node failures
  - Usually contain a distributed runtime environment

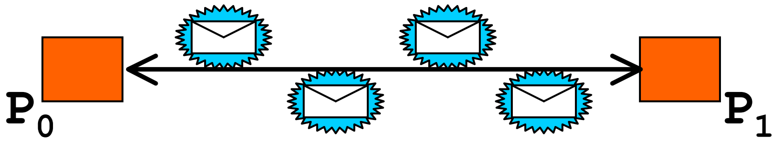
Open MPI

- Next generation MPI implementation

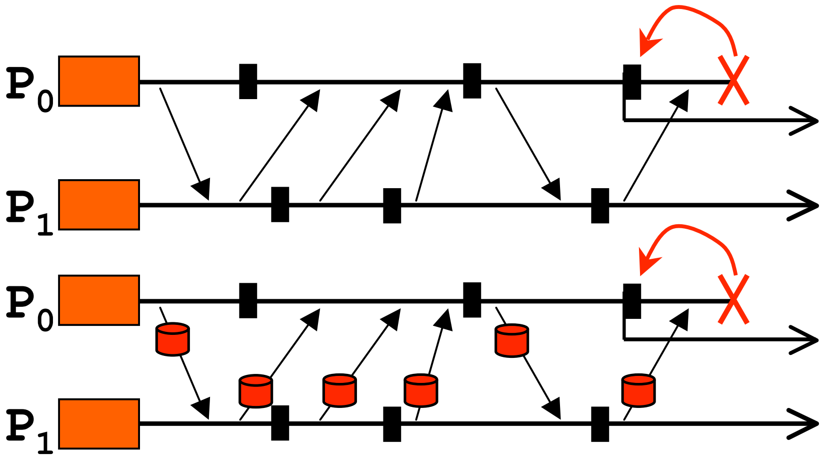
  Combine **best practices** from previous MPI implementations into a single **open source**, **production quality**, **MPI-2 compliant** MPI implementation.

Fault Tolerance Techniques

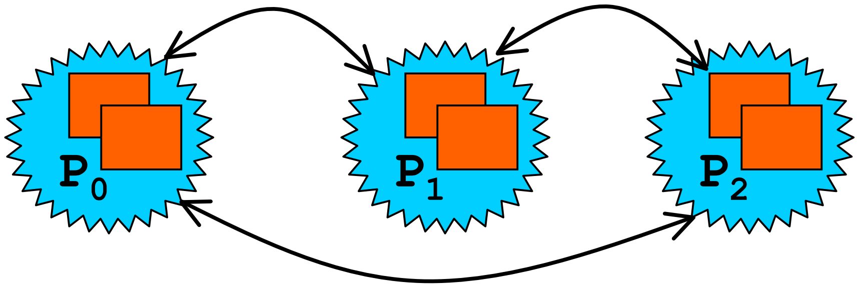
- Network failover & data reliability



- Rollback recovery
  - Checkpoint & restart
  - Message logging

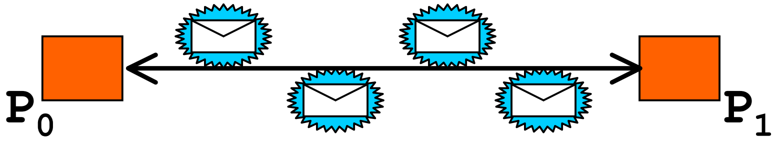


- Replication

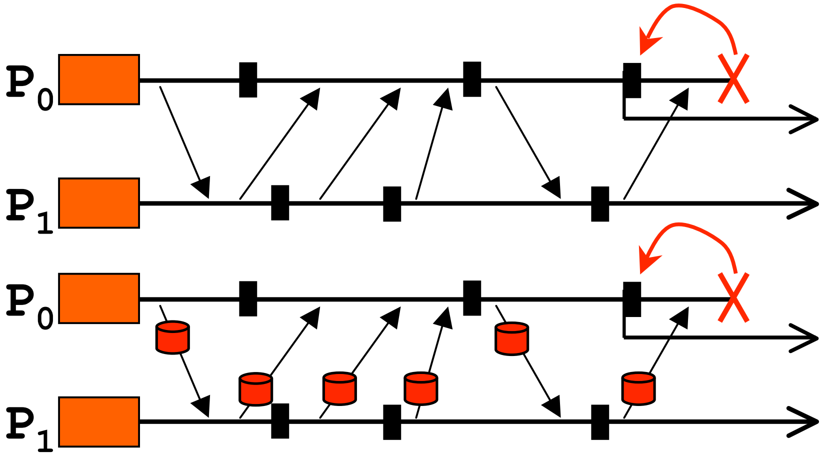


Fault Tolerance in Open MPI

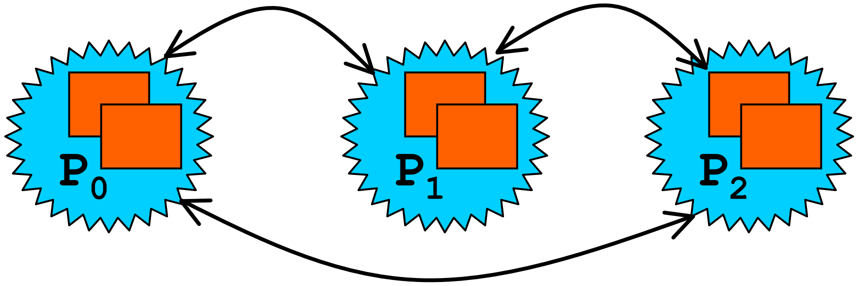
- Network failover & data reliability
  - **LA-MPI**



- Rollback recovery
  - Checkpoint & restart
    - **LAM/MPI**
  - Message logging



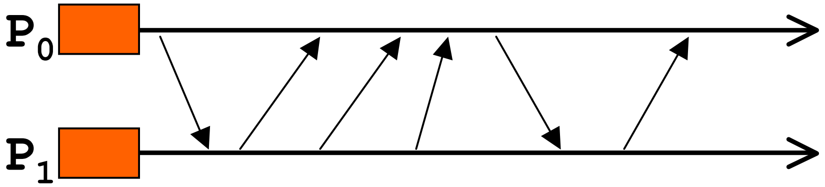
- Replication



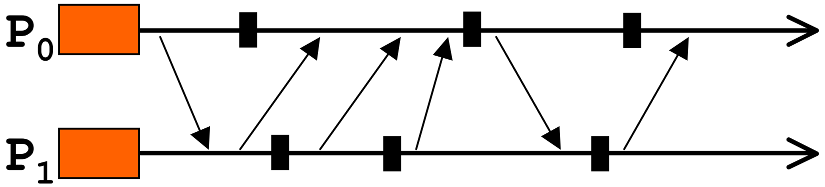
- Interactive
  - **FT-MPI**

Checkpoint/Restart

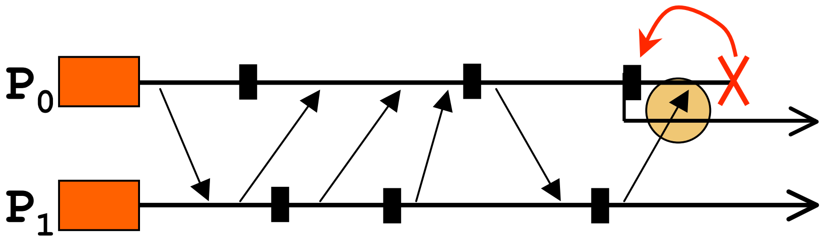
2 Processes using MPI



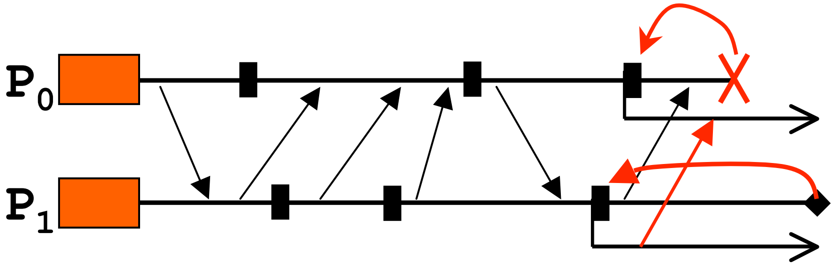
Checkpoint during failure-free execution



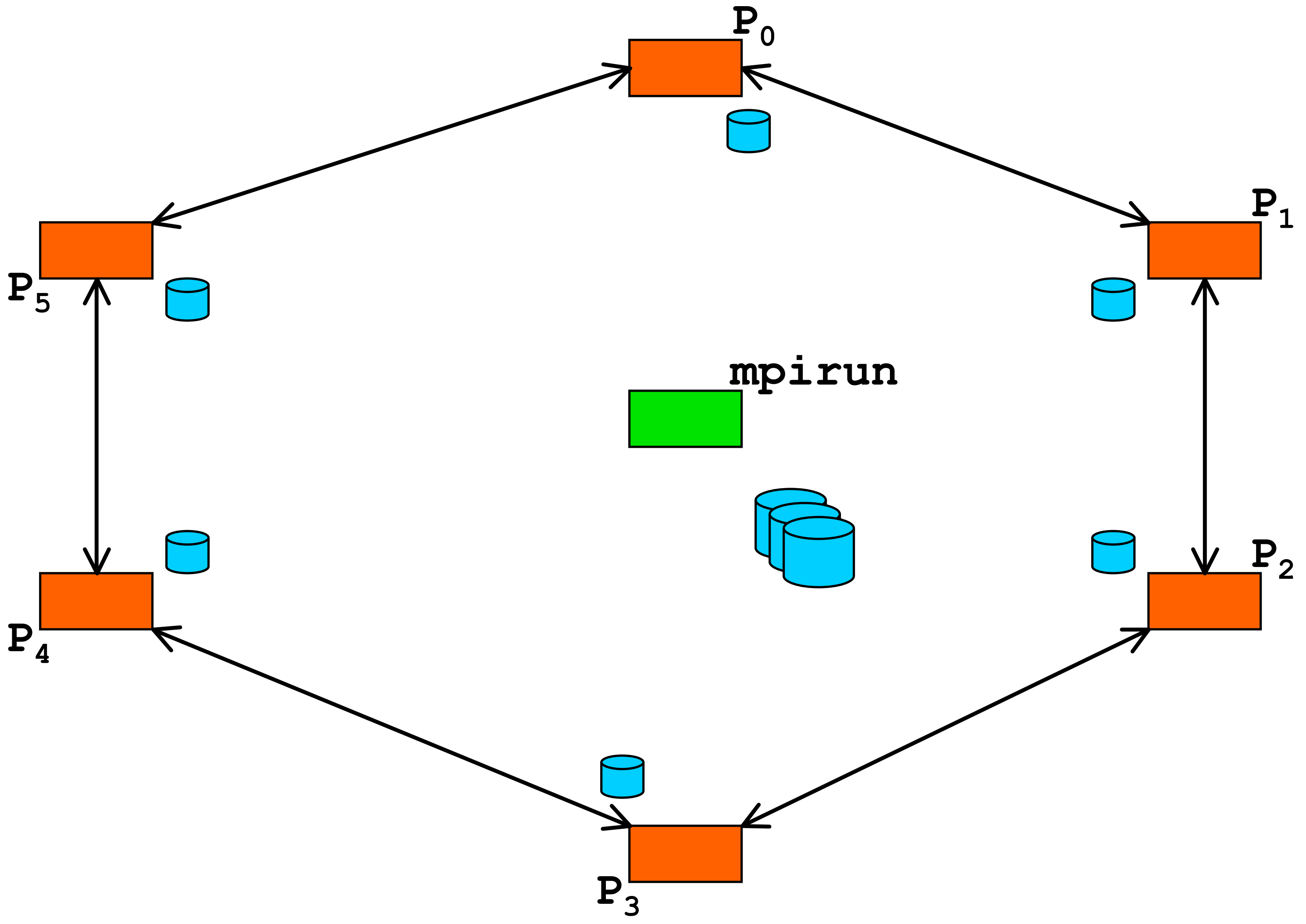
P_0 Fails
Restart it from last checkpoint



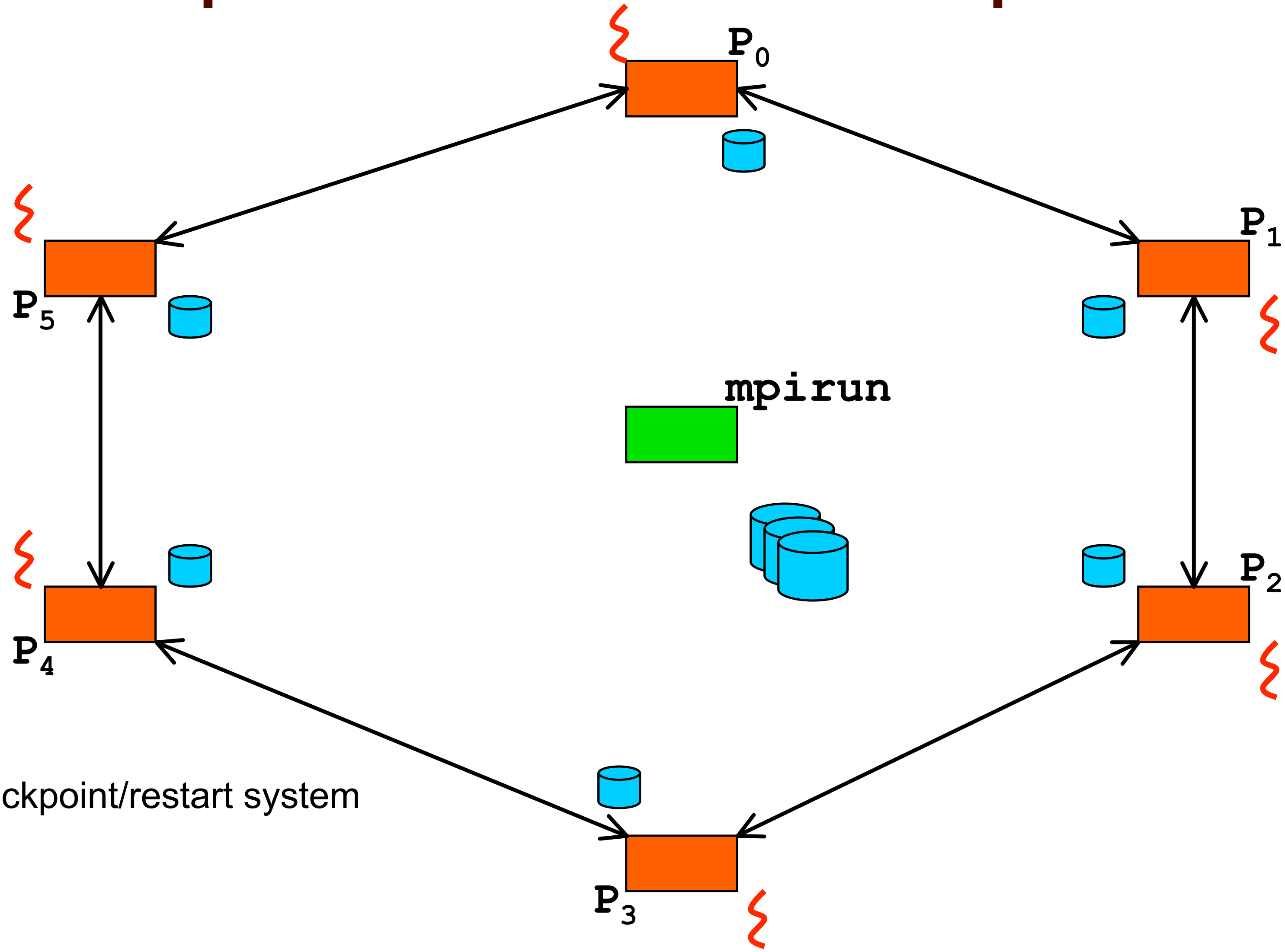
Rollback P_1 for consistency



Checkpoint/Restart in Open MPI

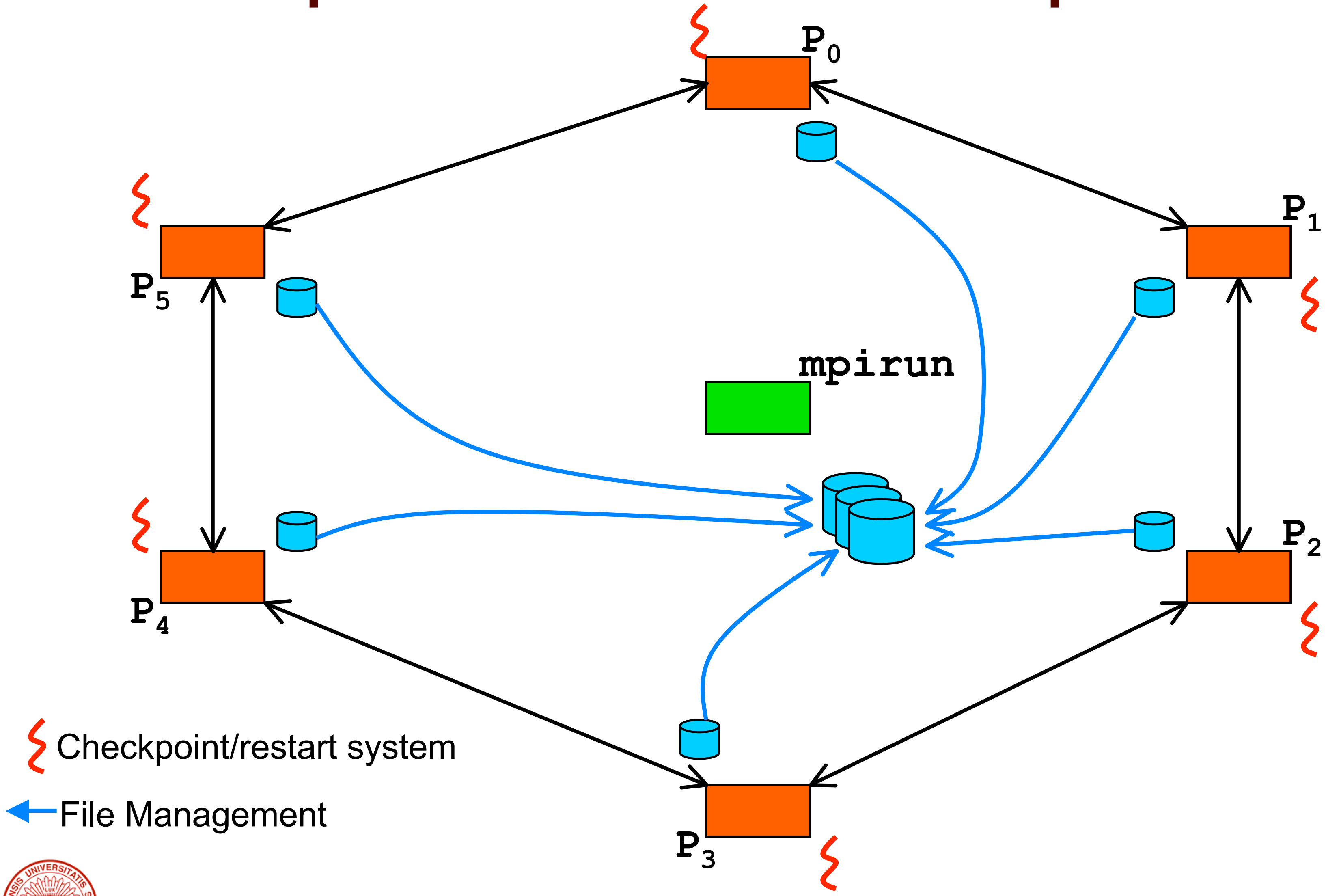


Checkpoint/Restart in Open MPI



⌇ Checkpoint/restart system

Checkpoint/Restart in Open MPI



P_0

P_1

P_2

P_3

P_4

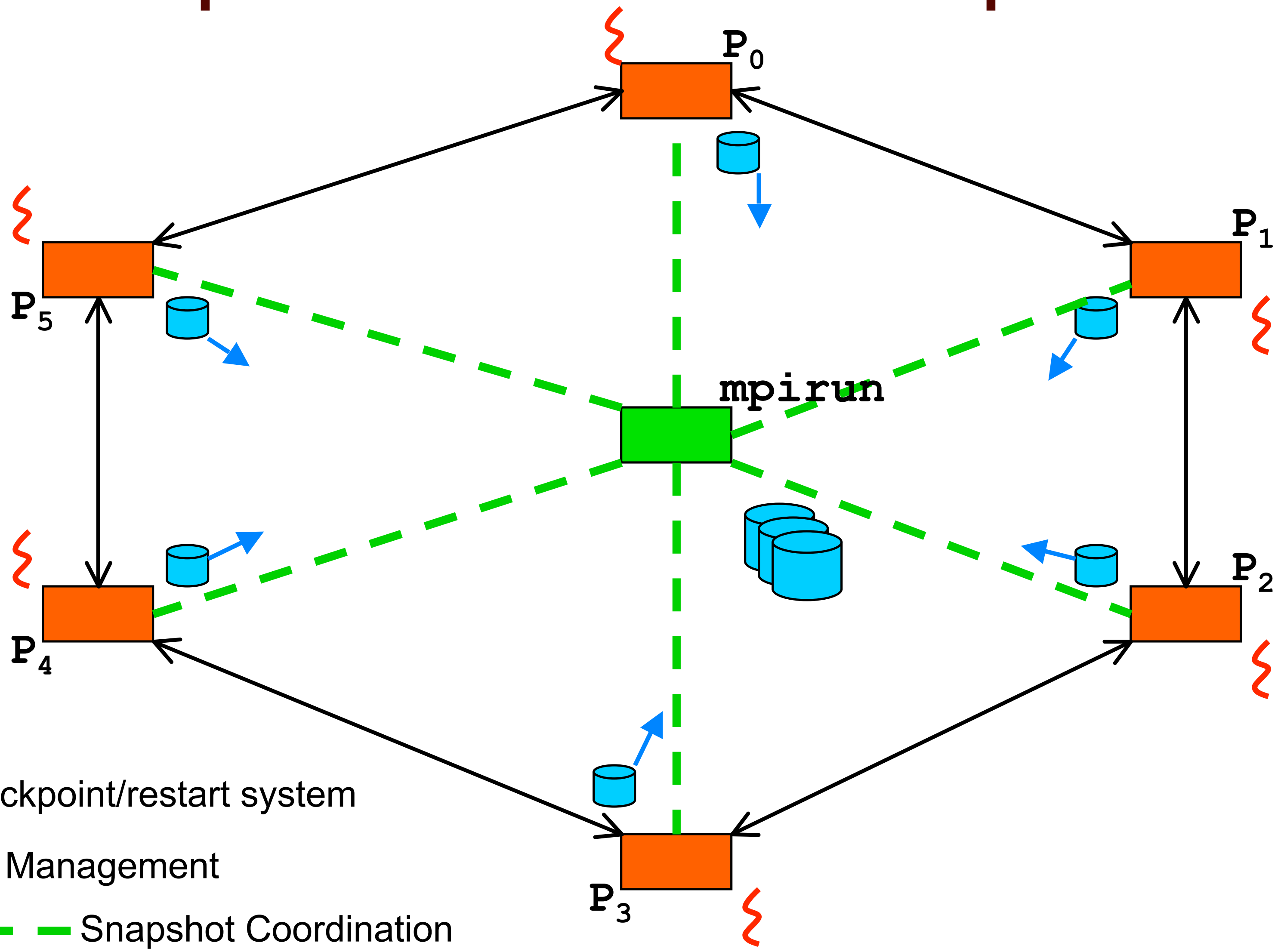
P_5

mpirun

Checkpoint/restart system

File Management

Checkpoint/Restart in Open MPI

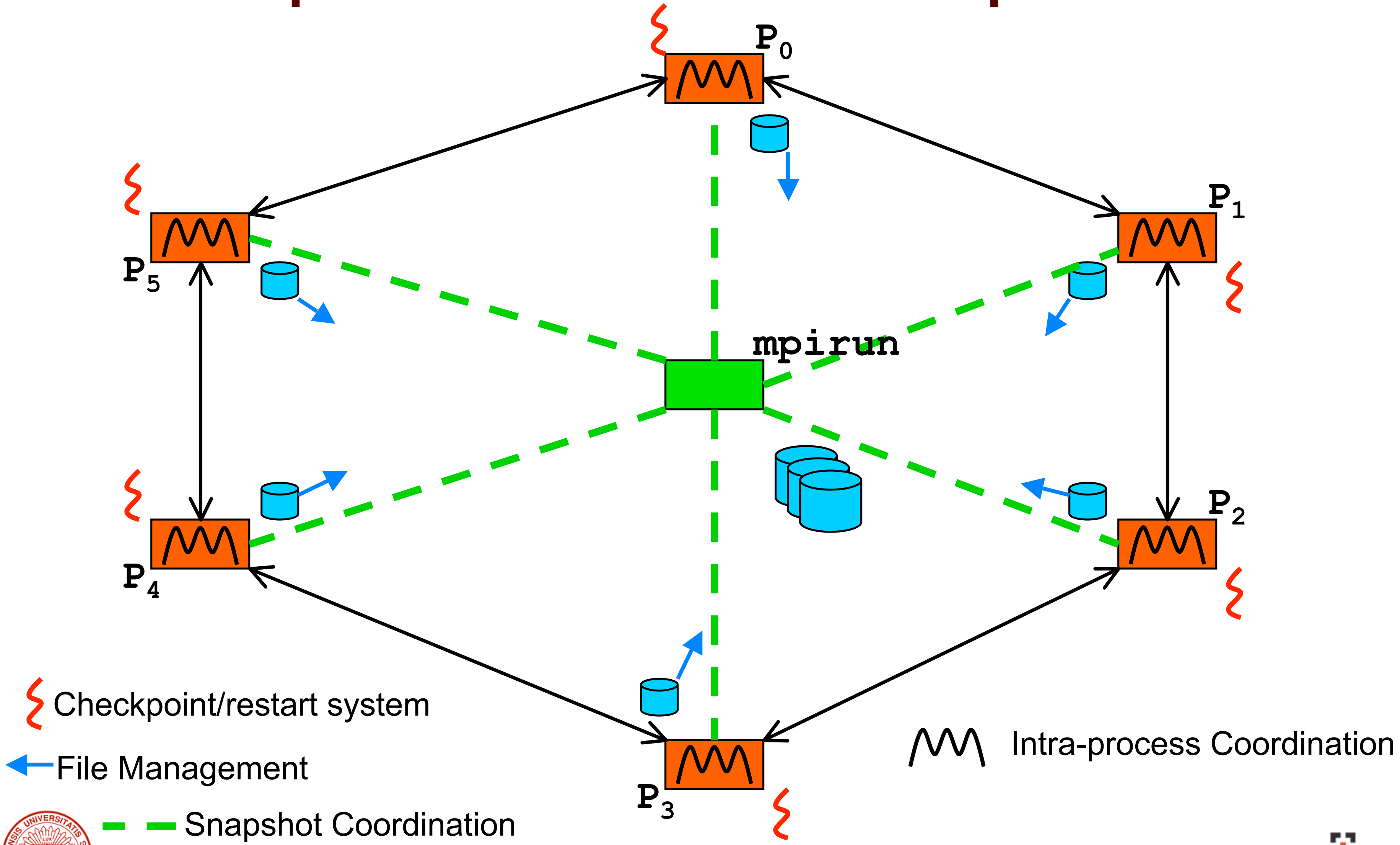


≀ Checkpoint/restart system

← File Management

- - Snapshot Coordination

Checkpoint/Restart in Open MPI



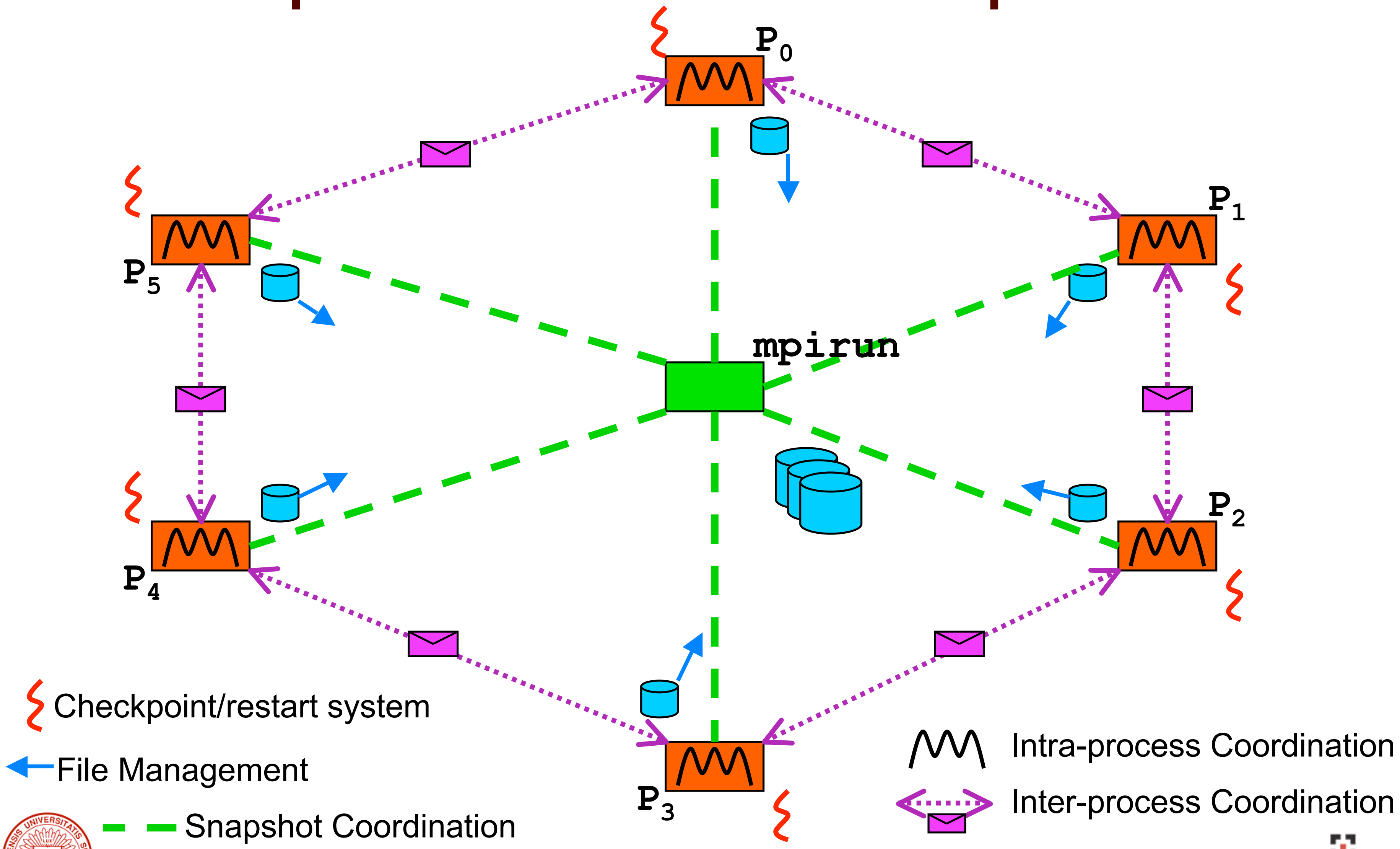
Checkpoint/restart system

File Management

Snapshot Coordination

Intra-process Coordination

Checkpoint/Restart in Open MPI



Checkpoint/restart system

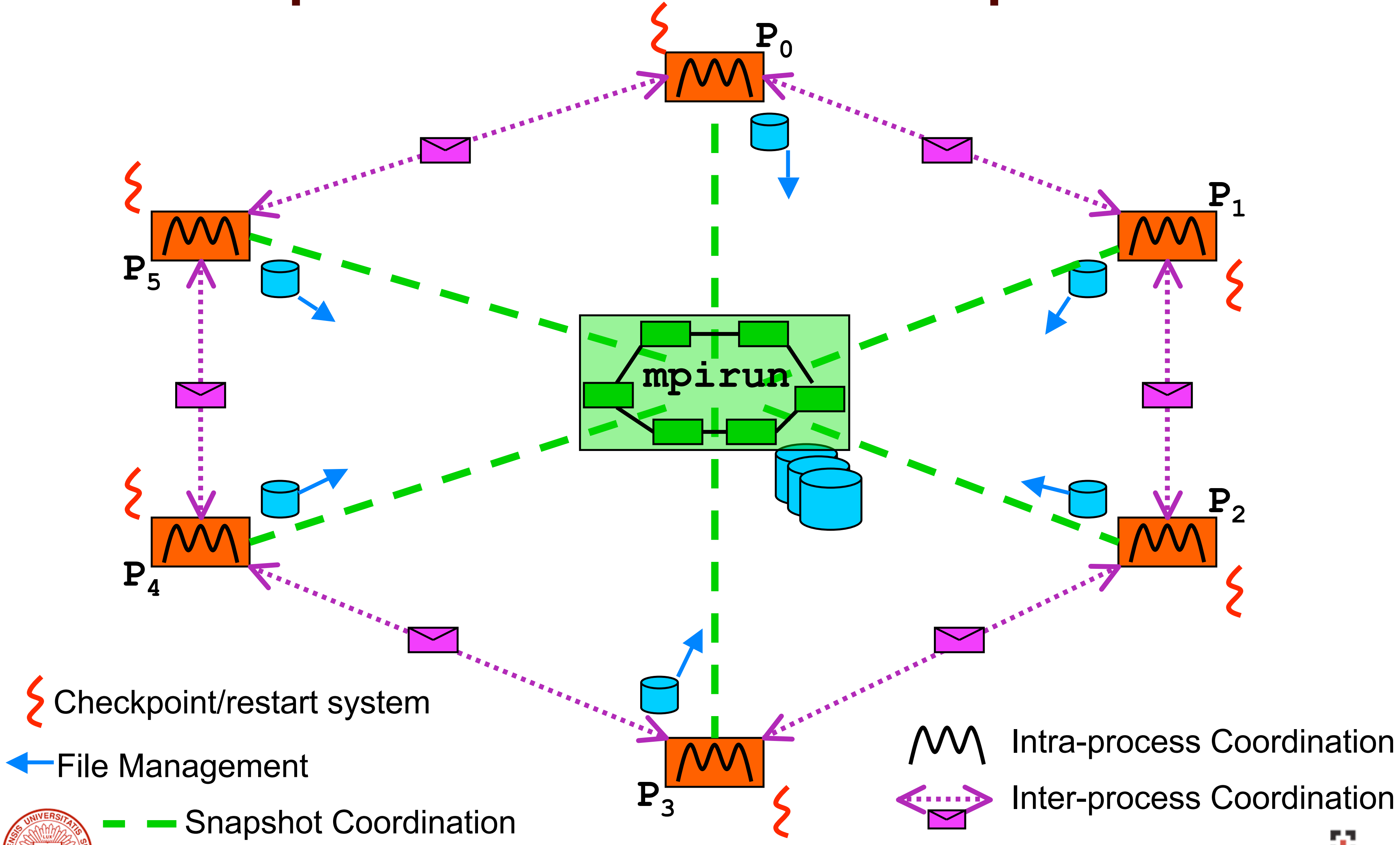
← File Management

Snapshot Coordination

Intra-process Coordination

Inter-process Coordination

Checkpoint/Restart in Open MPI



P_0

P_1

P_2

P_3

P_4

P_5

mpirun

Checkpoint/restart system

← File Management

Snapshot Coordination

Intra-process Coordination

Inter-process Coordination

Checkpoint/Restart in Open MPI

≀	Single process checkpoint/restart system (e.g., BLCR, libckpt, Condor, ‘self’)	OPAL CRS
↑	File management & movement (e.g., Unix, RSH/SSH, Out-of-band comm.)	ORTE FileM
¦	Snapshot Coordinator (e.g., Centralized, Replicated checkpoint servers)	ORTE SnapC
ʍ	Intra-process Coordinator (e.g., resolve network addresses)	INCs
↕	Inter-process Coordinator (e.g., Coordinated, Uncoordinated, Msg. Induced)	OMPI CRCP

J. Hursey, J. Squyres, A. Lumsdaine. **A Checkpoint and Restart Service Specification for Open MPI**. Technical Report TR635, Indiana University, July 2006.

J. Hursey, J. Squyres, T. Mattox, A. Lumsdaine. **The Design and Implementation of Checkpoint/Restart Process Fault Tolerance for Open MPI**. Submitted IPDPS ‘07.

What does this mean to me?

- Fault Tolerance Researcher:
  - Frameworks provide isolation
    - Benefit from progress in other areas
    - Focus on the experiment not MPI development
  - Apples-to-apples comparison of algorithms
- Application Developer:
  - Provide transparent fault tolerance solutions by default
  - Not required to know algorithmic details
  - Development hooks available for more fine grained control
- Application User:
  - Renewed focus on usable fault tolerance solutions
  - Seamless benefit from fault tolerance advancements

Demonstration

```
$ mpirun -np 2 --mca ft-enable cr my-app
At phase 1...
At phase 2...
At phase 3...
```

```
$
```

Demonstration

```
$ mpirun -np 2 --mca ft-enable cr my-app
At phase 1...
At phase 2...
At phase 3...
```

Slight pause in execution

```
$ ompi-checkpoint 1234
```

Demonstration

```
$ mpirun -np 2 --mca ft-enable cr my-app
At phase 1...
At phase 2...
At phase 3...
At phase 4...
```

Resume execution

```
$ ompi-checkpoint 1234
Ref: 0 global-snapshot-1234
$
```

Demonstration

```
$ mpirun -np 2 --mca ft-enable cr my-app
At phase 1...
At phase 2...
At phase 3...
At phase 4...
At phase 5...
$
```

Termination requested

```
$ ompi-checkpoint 1234
Ref: 0 global-snapshot-1234
$ ompi-checkpoint --term 1234
Ref: 1 global-snapshot-1234
```

Demonstration

```
$ mpirun -np 2 --mca ft-enable cr my-app
At phase 1...
At phase 2...
At phase 3...
At phase 4...
At phase 5...
$
$ ompi-restart global-snapshot-1234
At phase 6...
At phase 7...
At phase 8...
At phase 9...
```

Time passes…

```
$ ompi-checkpoint 1234
Ref: 0 global-snapshot-1234
$ ompi-checkpoint --term 1234
Ref: 1 global-snapshot-1234
```

Conclusions

- HPC applications must be prepared to handle system failure.
- MPI libraries are well positioned to provide (semi-)transparent fault tolerance solutions to HPC applications.
- Open MPI provides many fault tolerance solutions for modern HPC applications.

Wow! Where can I find this?

- **Network Failover & Data Reliability**
  - Scheduled to be released in **v1.2**
- **Rollback Recovery:** Checkpoint/Restart
  - Scheduled to be released in **v1.3**
  - First release will support:
    - MPI-1 standard point-to-point operations
    - Collective implementations layered over point-to-point operations
    - LAM/MPI-like coordinated checkpoint/restart
    - Asynchronous checkpoint/restart commands
- Watch the Open MPI mailing lists for updates:

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

Questions

[redacted]

Extra Slides