RM

Providing Resource Management Services to PVM Applications

Jim Pruyne
Miron Livny

Computer Sciences Department
University of Wisconsin — Madison
Madison, Wisconsin
{(pruyne,miron)@cs.wisc.edu}

Our view of the world

- ***Dynamic Applications***

  Some parallel applications can benefit from an increase in the number of hosts while they run and can survive a decrease in the number of hosts while they run.

- ***Dynamic Cluster***

  Some of the workstations in the cluster may leave/enter the pool of resources available for batch processing at unpredictable times.

Our view of the world (Cont.)

- ***The Resource Manager is the Boss***

  Everything (application processes and message passing system) has to run under the control of the Resource Management System (Condor).

- ***Layered Architecture***

  The message passing system is a layer between the OS and the Resource Management system that provides message passing services to the application and the manager.

Approach

- ***RM Tasks***

  All resource management services are provided by one or more Resource Management (RM) tasks that use the underlying message passing system to communicate with application tasks, daemons of the message passing system, and other RM tasks. RM tasks are special tasks. Each host used by the application has an RM task that serves requests generated by the host.

Approach (cont.)

☐ *Asynchronous RM*

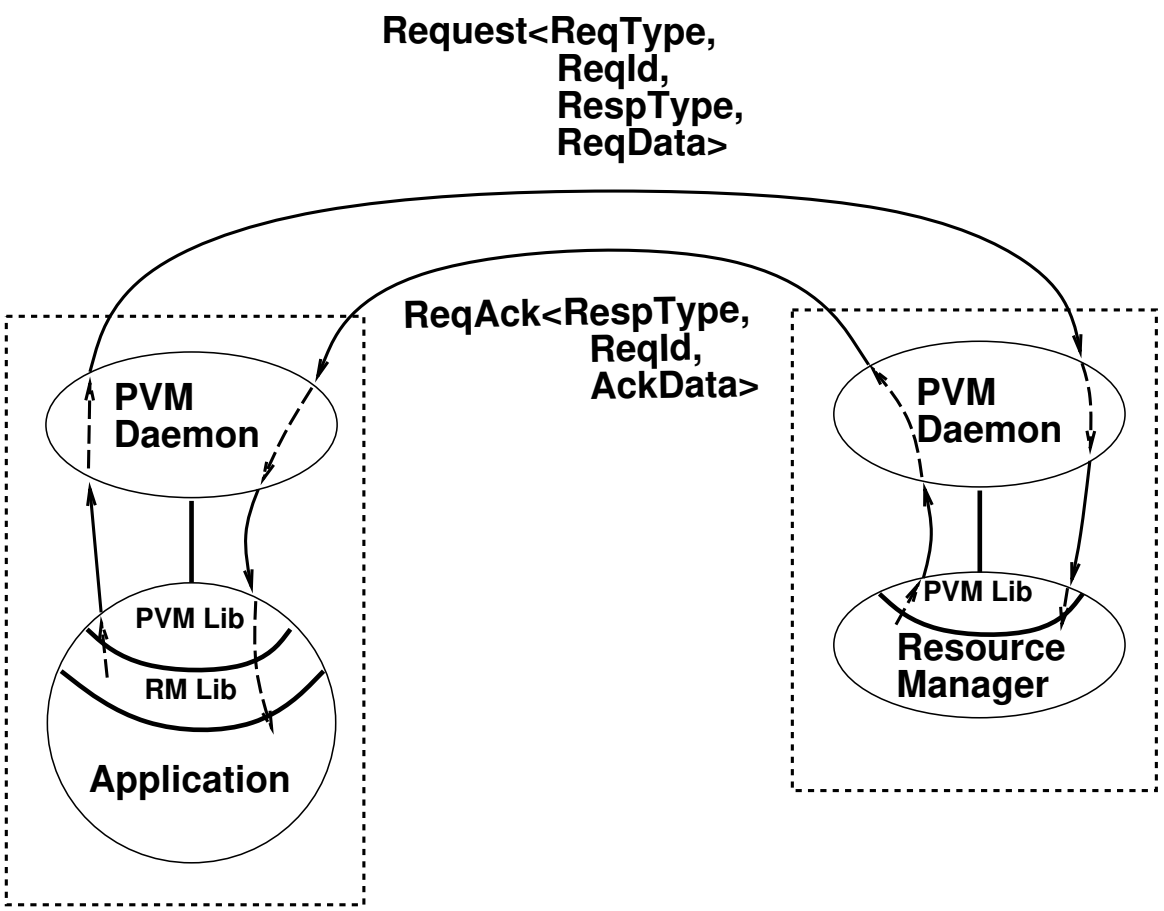
All Resource Management services are asynchronous.

- All run-time services are provided by the RM Library that translates RM requests to messages that are forwarded to the appropriate RM task.
- The RM replies to these requests by sending a message to the task that requested the service.

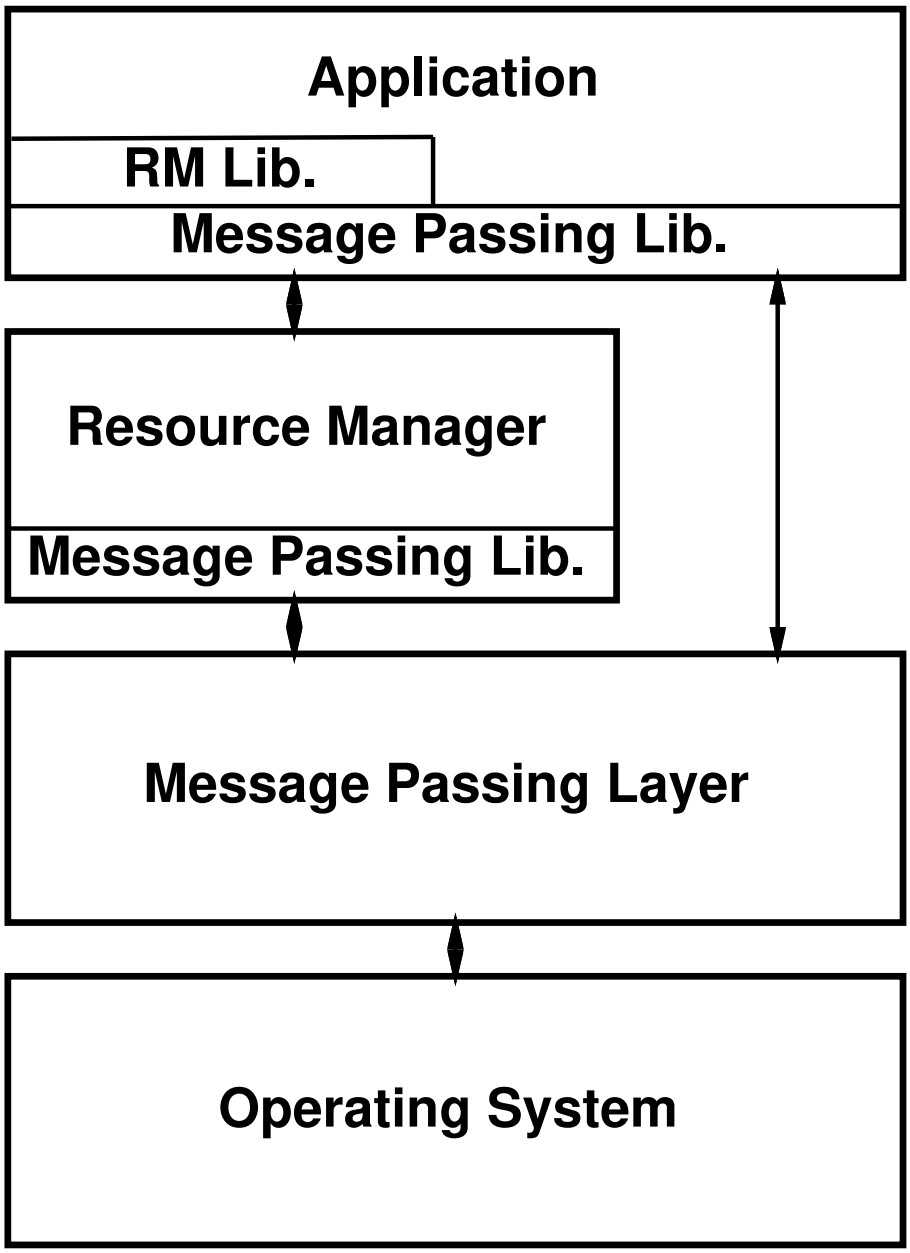
RM

- Every service request is assigned a unique sequence number. Every reply message sent by an RT task includes the sequence number of the service request that triggered it.
- Submit-time requests may be provided by the Batch system, one of the RM tasks, or both.

RM Tasks


Request<ReqType,
ReqId,
RespType,
ReqData>
ReqAck<RespType,
ReqId,
AckData>
PVM
Daemon
PVM Lib
RM Lib
Application
PVM
Daemon
PVM Lib
Resource
Manager

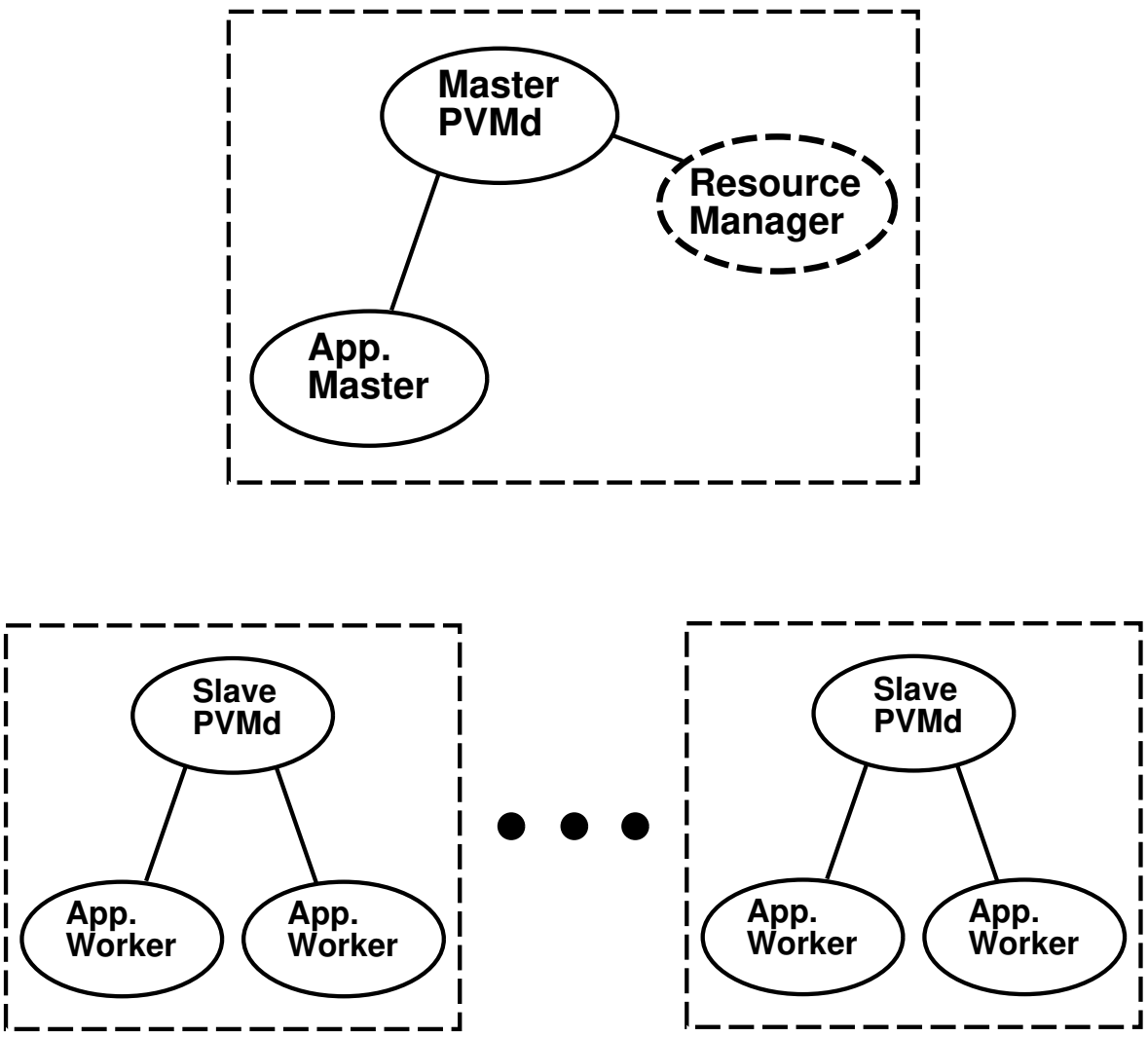
Layers


Application
RM Lib.
Message Passing Lib.
Resource Manager
Message Passing Lib.
Message Passing Layer
Operating System

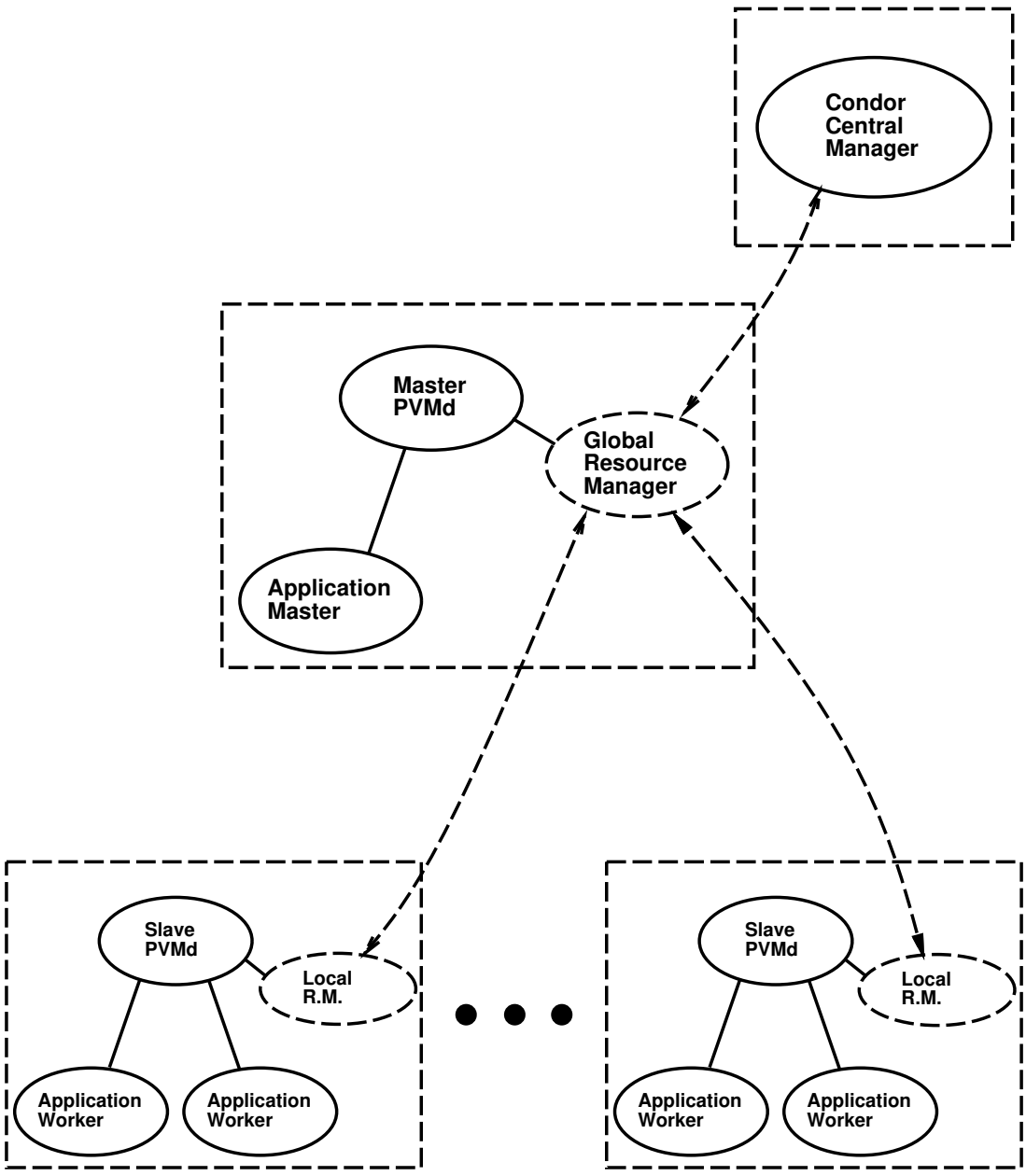
☐ *Host Classes*

Hosts are grouped into classes. Host classes can be defined at submit-time or at run-time. Requests to allocate hosts to the application are made in terms of these classes. Once a host has been allocated to the application, it can be specifically addressed by Resource Management services.

PVM Architecture



PVM + Condor Architecture



Changes to PVM daemon

- Introduced the notion of resource manager (RM) tasks.
- Each PVM daemon has an RM task associated with it.
- Provided a new function (pvm_register_rm()) so that a task may become the RM for its local daemon.

RM

- Modified task start-up so that when a new task is born, it inherits the PVM daemon's RM.
- Introduced a new message class for messages between the RM and PVM daemons or the PVM library.

Changes to PVM library

- Remove all Resource Management routines from the PVM library.
- Implement all Resource Management routines as asynchronous requests to the RM task associated with the requester and placing them in the RM library.

RM

- pvm_addhosts() does not use names of hosts to be added. Instead it uses counts and host classes.
- New notifications for hosts suspending and hosts resuming have been added.
- Added a get_rm_id() function to the message passing library.
- Include request sequence numbers in the header of messages.

RM

Profile of PVM Job #544.0

Wall Time 47:15:17

CPU Time 1142:47:17

Occupancy Time 1240:18:08

Machines Lost 257

Max. # Of Machines 27

Lost Due to Suspend 62:06:49

What Did We Learn?

- It works!!!
- We can support more than one host class.
- In most cases we can get a new host up and running in less than 30 seconds.
- It is important to know when a host is suspended.
- We need better tools to monitor this kind of jobs.

What Next

- Test system on some real applications.
- Other message passing systems.
- Scheduling Parallel Applications.
- Add Process Migration.