Scheduling PVM on Workstation Clusters using Condor

Miron Livny
Jim Pruyne

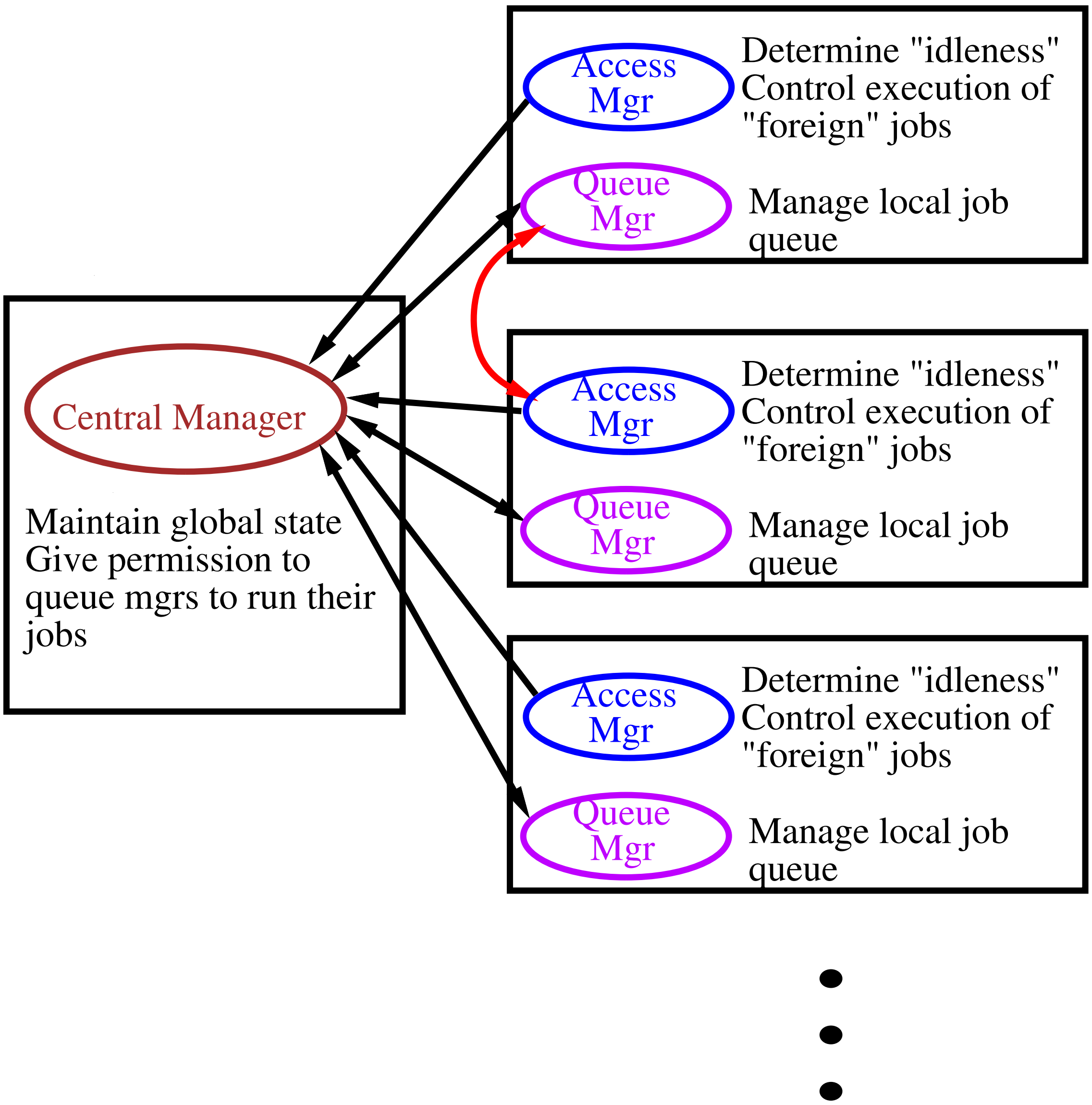
Computer Sciences Department
University of Wisconsin — Madison
Madison, Wisconsin
{miron,pryune@cs.wisc.edu}

Condor

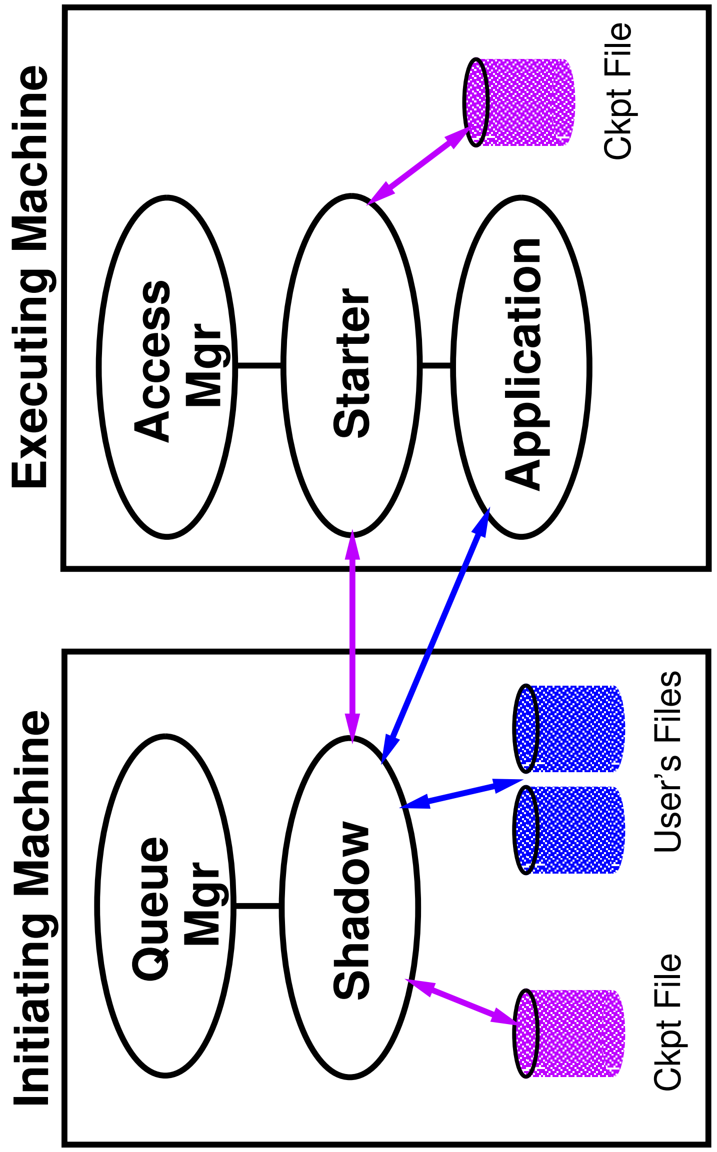
A Batch system for a cluster of privately owned UNIX workstations.

- Available (in source code) from ftp.cs.wisc.edu.

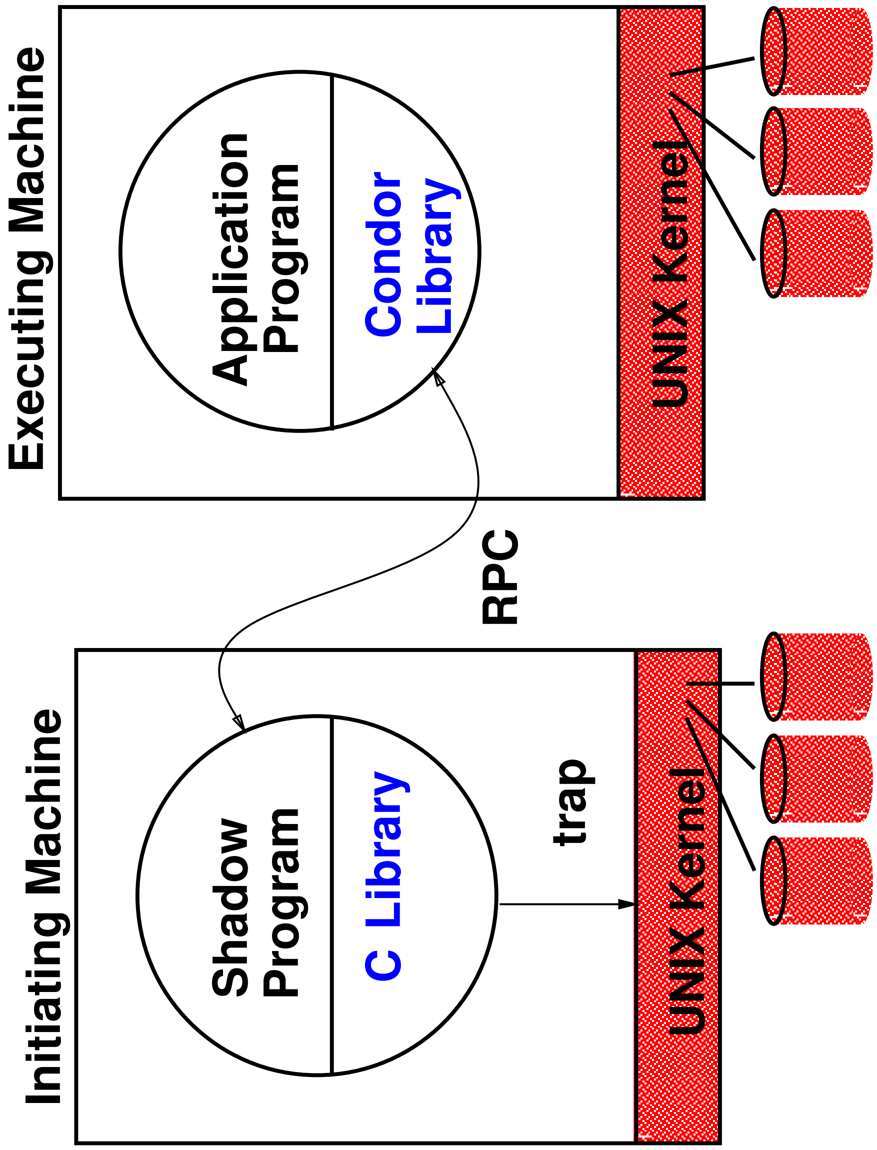
Condor Control Structure



Remote Execution



Remote File Access



Does not assume

Uniform accounts and a uniform file system across cluster. In other words, user may not have an account on executing machine and executing machine may not have access to the files of the user.

Our view of the world

- ***Dynamic PVM***

  Some PVM 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 at unpredictable times.

- ***Condor is the Boss***

  Everything (application processes and PVM daemons) has to run under the control of Condor.

PVM <-> Condor Interface

- ***Submit Time***
  - User -> Condor
- ***Run Time***
  - Application <-> Condor
  - PVM system <-> Condor

Submit Interface

- Inform the scheduler that this is a PVM job.
- Use Condor expressions to define classes of workstations. A workstations belongs to a class if its resources satisfy the conditions of the expression that defines the class.
- For each class provide the number of workstations required to run the job and the number of workstations desired.
- Executable of the "master" process.

Runtime Interface

☐ *Application -> Condor*

- Add hosts - provide class of workstations and number. (asynchronous)
- Delete hosts - provide class of workstations and number.
- Spawn - provide executable and class of workstation.
- Notify - request notification for asynchronous events (Host added, Host revoked, Process terminated).

☐ *Condor -> Application*

Notification messages on asynchronous events.

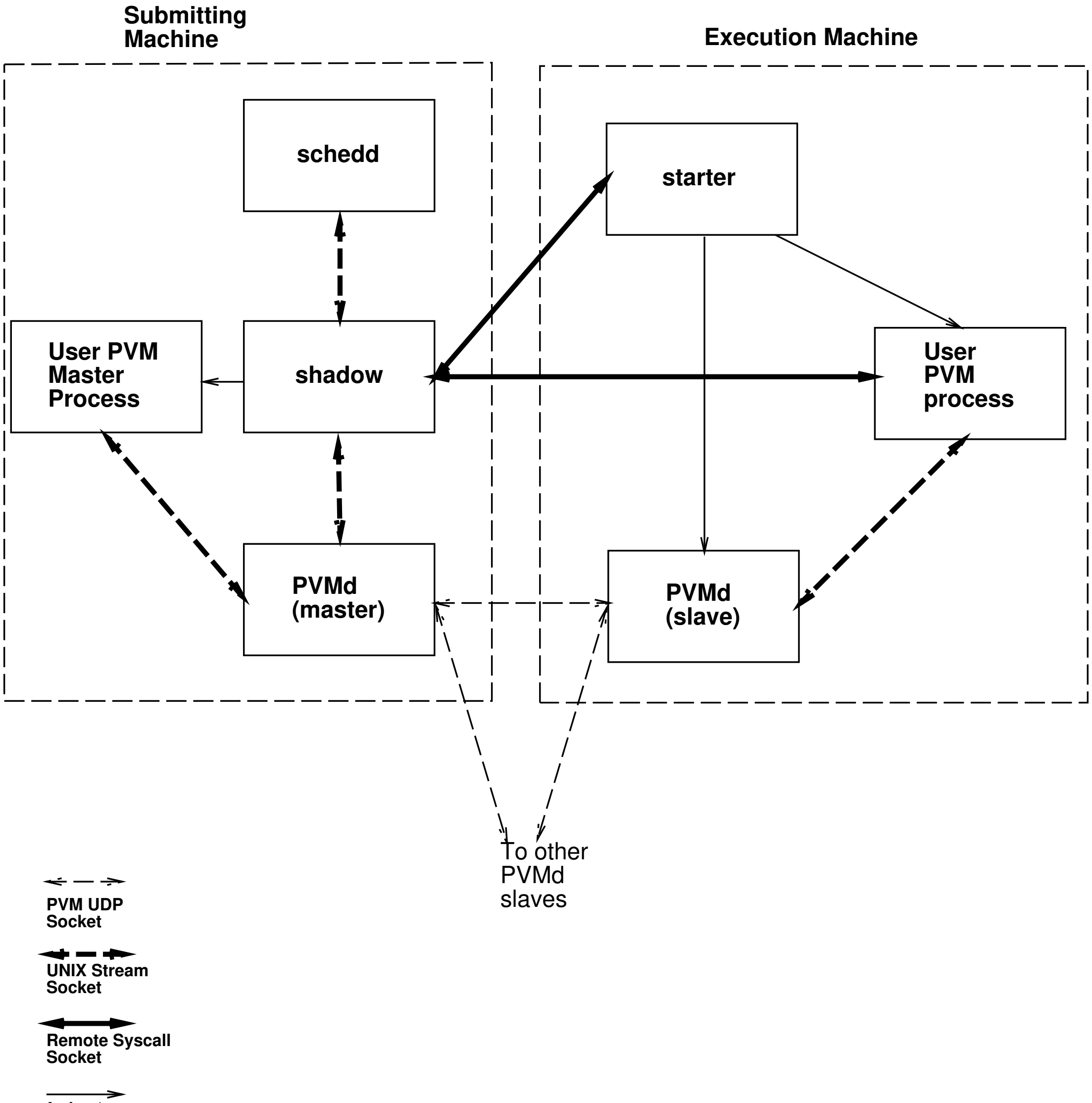
Runtime Interface (Cont.)

- ***PVM -> Condor***
  - Spawn, add host, delete host.
  - Process connected/terminated.
  - Host connected/removed.

- ***Condor -> PVM***

  Condor does not initiate any communication with PVM.

System Architecture



Scheduling

Based on the Up-Down allocation policy of Condor. Each user has a priority that depends on past and current resource usage and requests.

- Machines are reserved for one scheduling interval (5 minutes).
- Users with reserved machines lose priority.
- Users with pending requests gain priority.

Status Report

- Integrated system is running on a small Condor cluster of dedicated DecStations.
- A number of ’real life’ PVM application are used to test and evaluate the system.
- In 2-4 weeks we plan to install the integrated system on our production cluster (> 250 workstations)
- Over the Summer we plan to evaluate the system and the scheduling policy.