Condor for the Grid

1) Condor and the Grid. Douglas Thain, Todd Tannenbaum, and Miron Livny. In *Grid Computing: Making The Global Infrastructure a Reality*, Fran Berman, Anthony J.G. Hey, Geoffrey Fox, editors. John Wiley, 2003.

2) Condor-G: A Computation Management Agent for Multi-Institutional Grids. J. Frey, T. Tannenbaum, I. Foster, M. Livny, S. Tuecke. In *Proceedings of 10th International Symposium on High Performance Distributed Computing (HPDC-10)*, August 2001

3) http://www.cs.wisc.edu/condor/

Condor in relation to the Grid

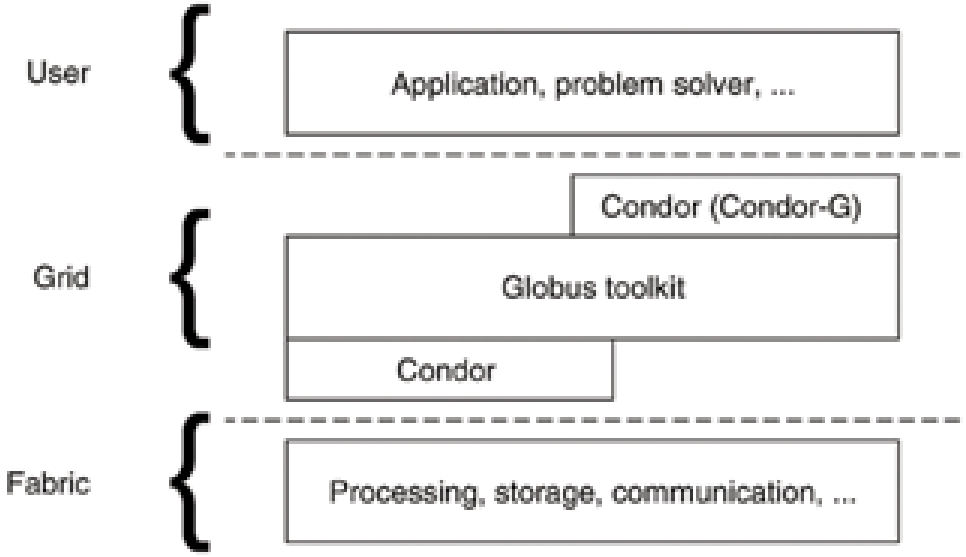


Figure 11.2 Condor technologies in Grid middleware. Grid middleware consisting of technologies from both Condor and Globus sit between the user's environment and the actual fabric (resources).

Figure 11.2 from (1)

Slight Grid Detour

- Condor-G exploits a few Grid protocols
  - GSI (Grid Security Infrastructure)
    - Essential building blocks for other Grid protocols
    - Handles authentication and authorization
  - MDS-2 (Methods for Discovering and Disseminating)
    - Resource uses GRRP (Grid Resource Registration Protocol) for notification to other Grid entities
    - Those entities use GRIP(Grid Resource Information Protocol) to obtain information about the resource status
  - GASS(Global Access to Secondary Storage)
    - Mechanisms for transferring data between a remote HTTP, FTP, or GASS Server
  - GRAM (Grid Resource Allocation and Management)

GRAM

- In 1998, Globus designed GRAM
  - GRAM (Grid Resource Allocation and Management Protocol)
  - Provides an abstraction for remote process queuing
  - Three Aspects important in relation to Condor
    - Security
    - Two-phase commit
    - Fault tolerance
- To take advantage of GRAM, the user needs a system to:
  - Remember what jobs have been submitted, where the jobs are, and what those jobs are doing
  - Handle job failure and possible resubmitting of jobs
  - Track large numbers of jobs with queueing, prioritization, logging
- Condor-G was created to do these things

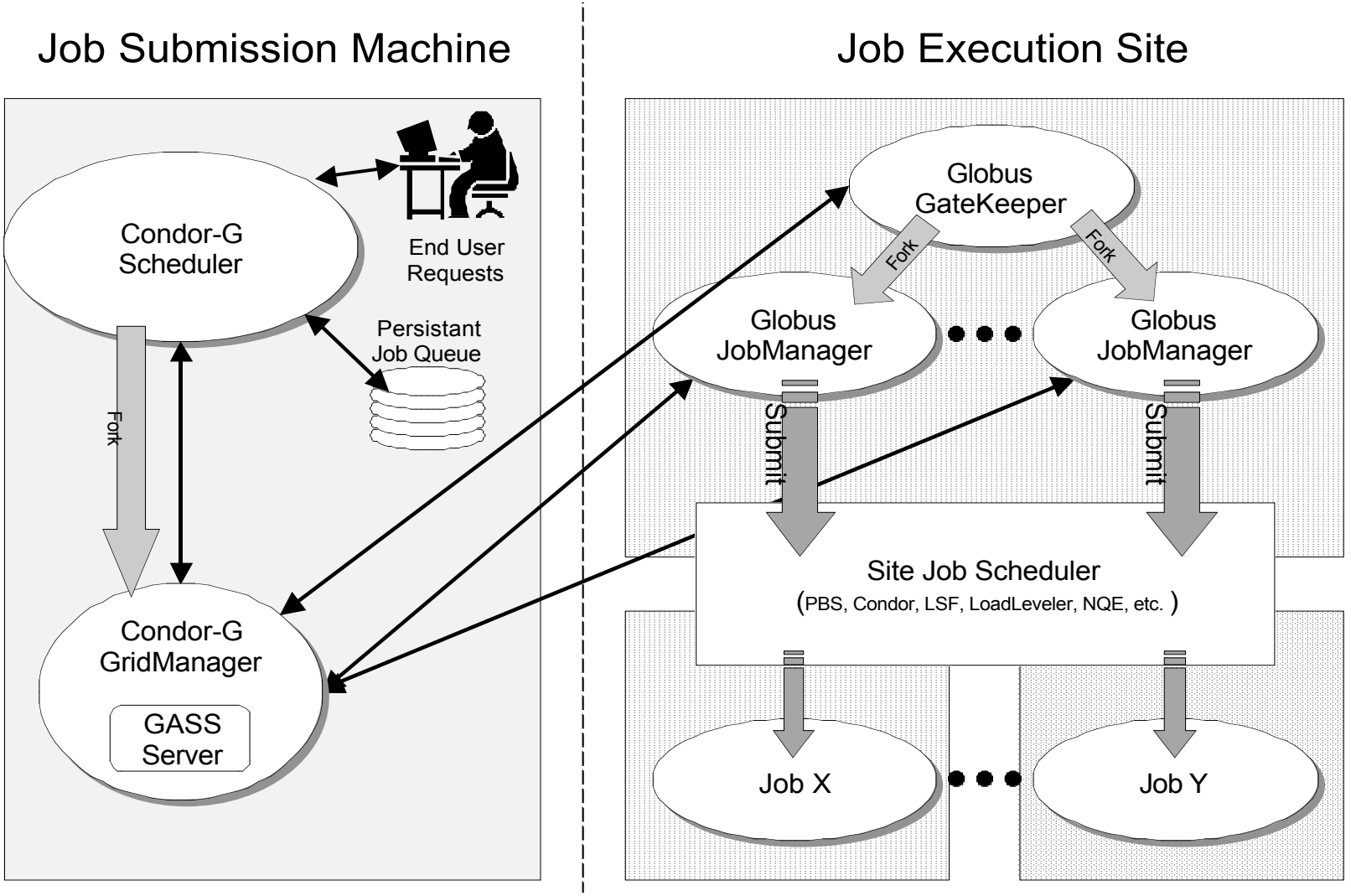
Intro to Condor-G

- Treats Grids as a local resource
- API and command line tools to
  - Submit jobs
  - Query a job's status or cancel the job
  - Be informed of job termination or problems
  - Obtain access to detailed logs
- Maintains look and feel of local resource manager

Condor-G (under the hood)

- Stages a job's standard I/O and executable using GASS
- Can use GRAM to submit jobs to remote queues
- Problems
  - Must direct job to a particular queue without knowing the availability of resources
  - Does not support the features of each system underlying GRAM
  - Many of Condor's features like checkpointing also missing
- Wait a tick … I think we can fix that last problem
- "Gliding in" allows us to have reach of Condor-G and the features of Condor

Condor without “gliding in”

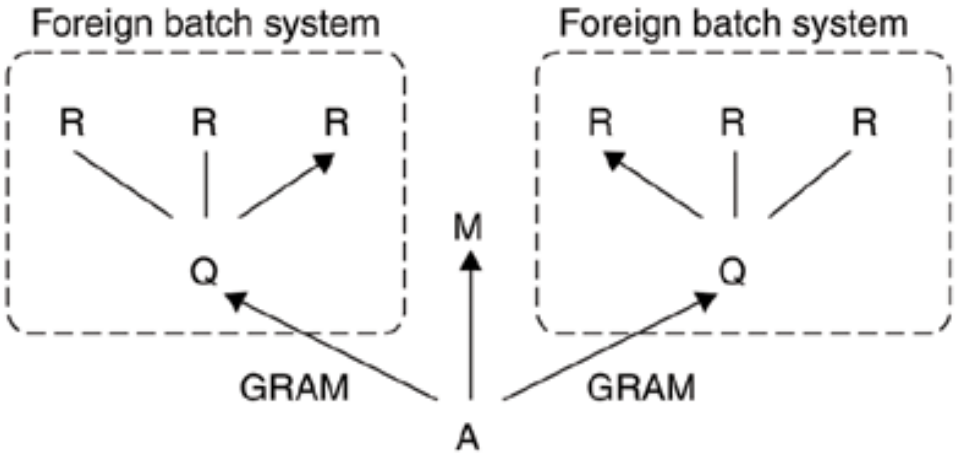


Gliding In (all figures from 11.9(3))

- Process requires three steps

Step one:

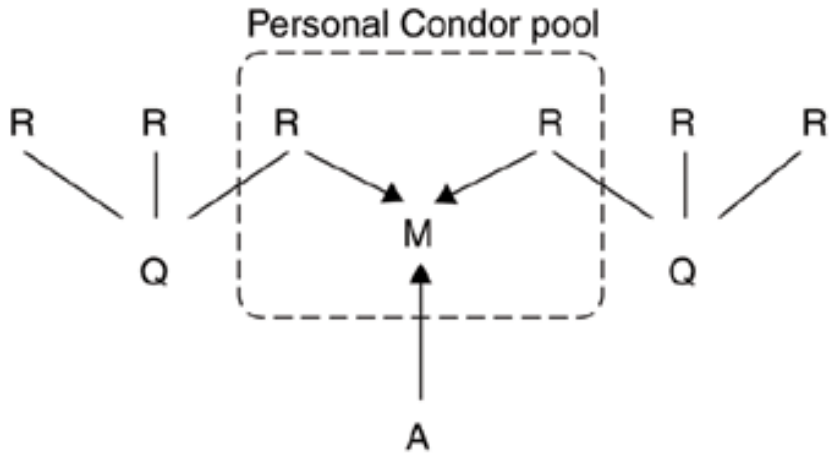
User submits Condor daemons as batch jobs in foreign systems



Gliding In - Step Two

Step two:

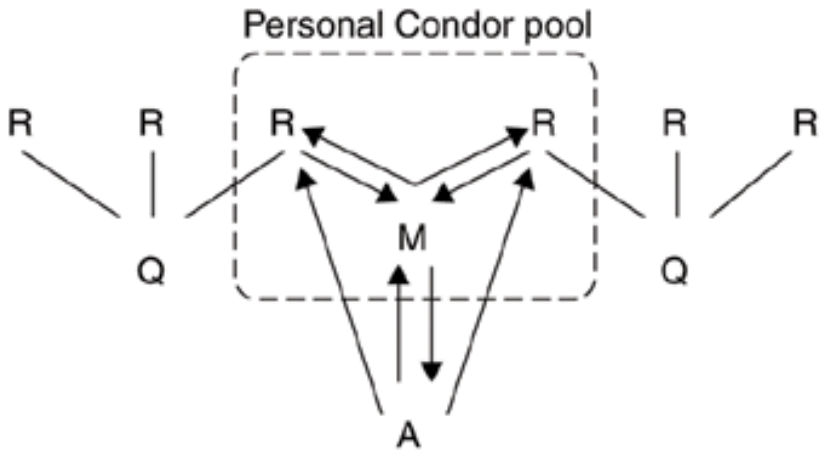
Submitted daemons form an *ad hoc* personal Condor pool



Gliding In - Step Three

Step three:

User runs jobs on personal Condor pool



Condor-G with “gliding in”

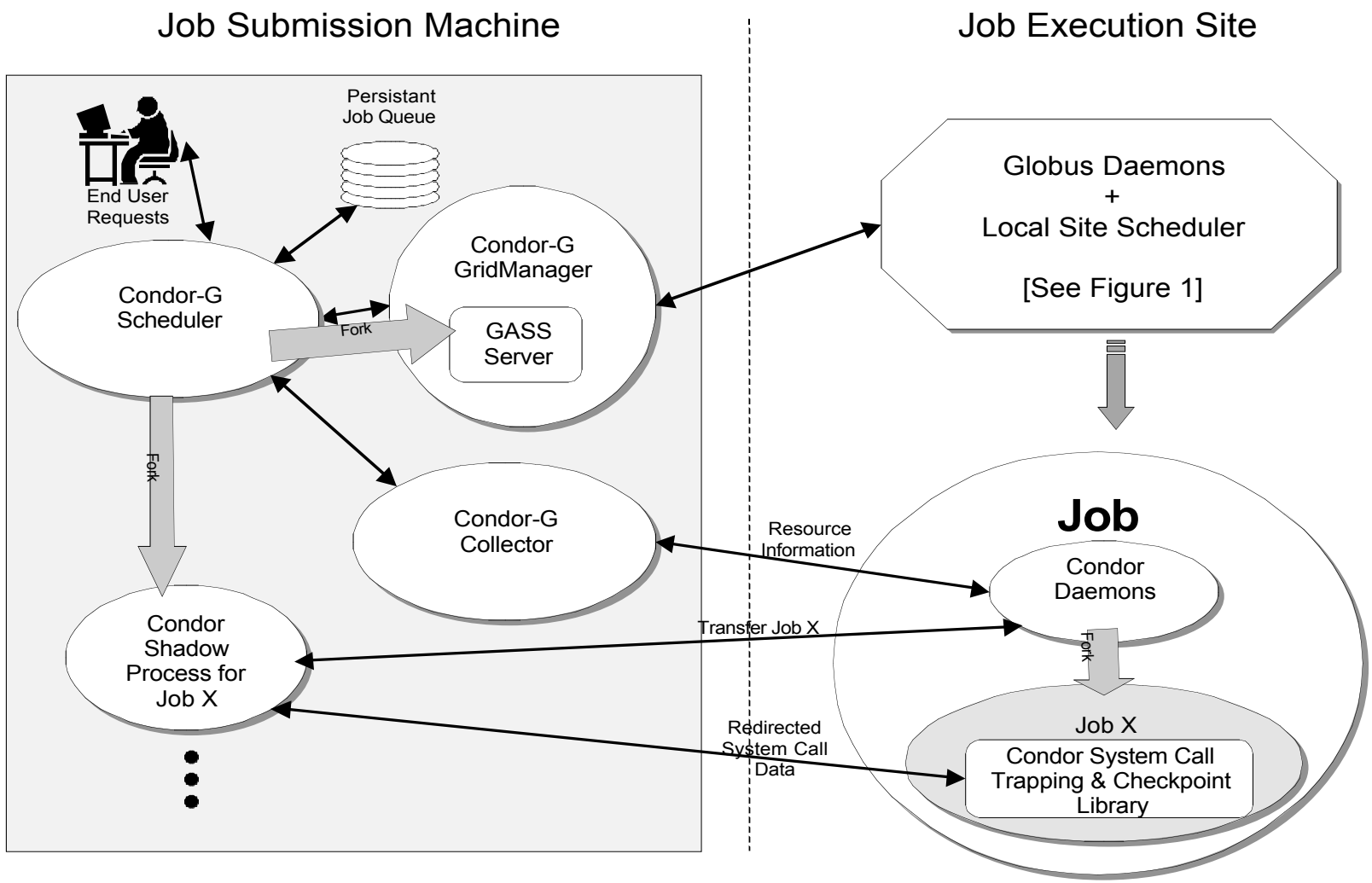


Figure 2 from (2)

More Condor-G tidbits

- Failure handling
  - Crash of Globus JobManager
  - Crash of the machine that manages the remote resource
  - Crash of the machine on which the GridManager is executing
  - Network failure between two machines
- Credential management
  - GSI proxy given finite lifetime
  - Condor-G agaent periodically analyzes credentials for users with queued jobs
  - Credentials may have to be forwarded to a remote location
- Resource discovery and scheduling
  - Different strategies
    - User supplied list of GRAM servers
    - Personal resource broker that runs as part of Condor-G agent

Condor-G FAQ From (3)

- **Do I need to have a Condor pool installed to use Condor-G?**

  No, you do not. Condor-G is only the job management part of Condor. It makes perfect sense to install Condor-G on just one machine within an organization. Note that access to remote resources using a Globus interface will be done through that one machine using Condor-G.2.
- **I am an existing Condor pool administrator, and I want to be able to let my users access resources managed by Globus, as well as access the Condor pool. Should I install Condor-G?**

  Yes
- **I want to install Globus on my Condor pool, and have external users submit jobs into the pool. Do I need to install Condor-G?**

  No, you do not need to install Condor-G. You need to install Globus, to get the "Condor jobmanager setup package."

More from the Condor-G FAQ

- **I am the administrator at Physics, and I have a 64-node cluster running Condor. The administrator at Chemistry is also running Condor on her 64-node cluster. We would like to be able to share resources. Do I need to install Globus and Condor-G?**

  You may, but you do not have a need to. Condor has a feature called flocking, which lets multiple Condor pools share resources. By setting configuration variables, jobs may be executed on either cluster. Flocking is good (better than Condor-G) because all the Condor features continue to work. Examples of features are checkpointing remote system calls. Unfortunately, flocking only works between Condor pools. So, access to a resource managed by PBS, for example, will still require Globus and Condor-G to submit jobs into the PBS queue.

- **Why would I want to use Condor-G?**

  If you have more than a trivial set of work to get done, you will find that you spend a great deal of time managing your jobs without Condor. Your time would be better spent working with your results. For example, imagine you want to run 1000 jobs on the NCSA Itanium cluster without Condor-G. To use the Globus interface directly, you will type 'globusrun' 1000 times. Now imagine that you want to check on the status of your jobs. Using 'globus-job-status' 1000 times will not be much fun. And, heaven help you if you find a bug and need to cancel your thousand jobs. This will be followed by resubmitting all 1000 jobs. Under Condor-G, job submission and management is simplified. Condor-G will also ease submission and management of jobs when job flow is complex. For example, if the first 100 jobs must run before any of the next 100 jobs, and once those 200 jobs are done, the next 700 may start. The last 100 may run only after the first 900 finish. Condor DAGMan manages all of the dependencies for you. You can avoid writing a rather complex Perl script.