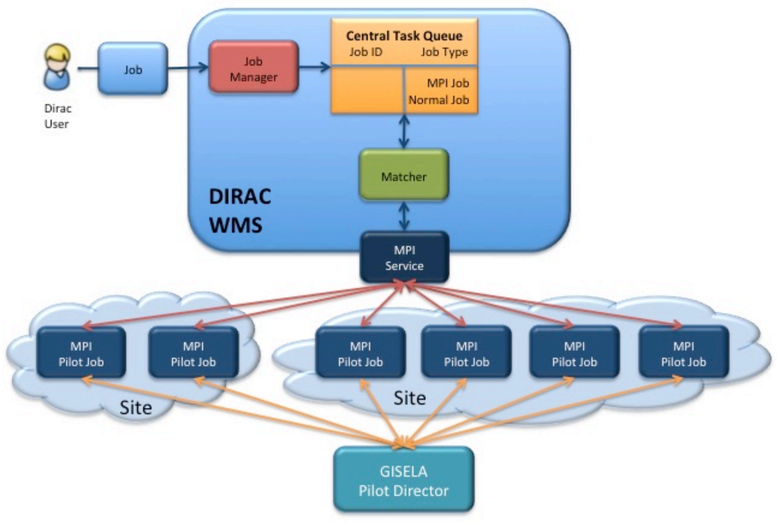
MPI support in the Pilot Job WMS

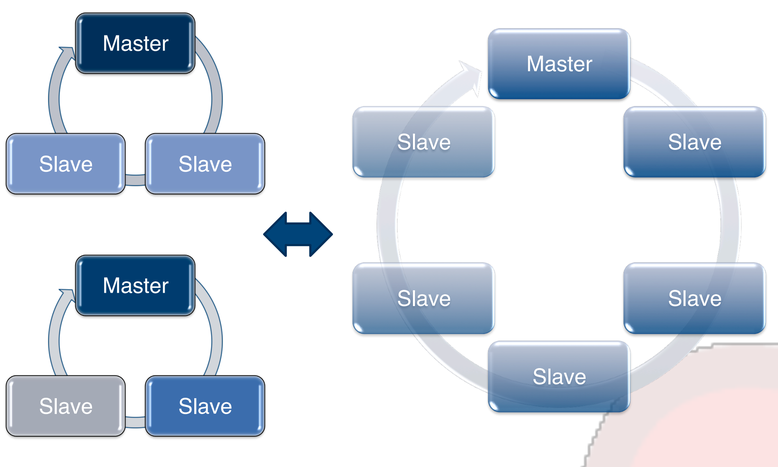
DIRAC COMMUNITY GRID SOLUTION ™

DIRAC Core Services



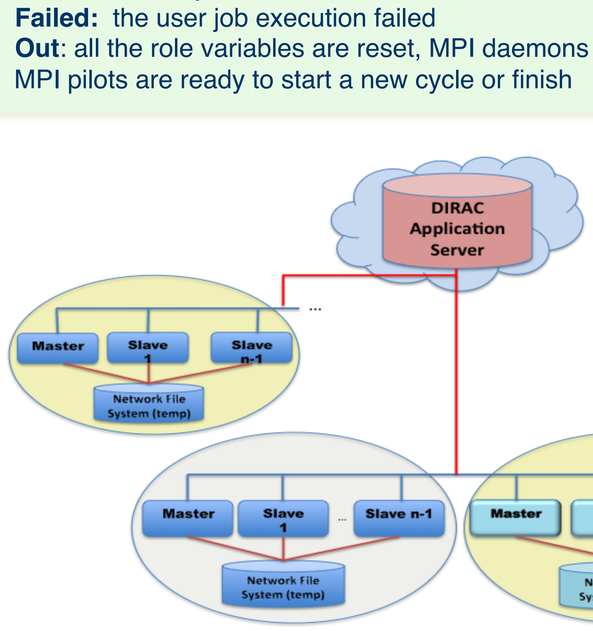
The DIRAC WMS MPI Components:

- **MPI service** orchestrates the MPI Pilot Jobs behavior in connection with the DIRAC Matcher Service to get information about the central Task Queue
- **MPI Pilot Job** is responsible to start MPI Agent in the WN and decide when to free the resource.
- **MPI Agent** connects to the MPI service, sends the information about the WN capacity; creates the environment according to information received from MPI Service; starts, finalizes MPI ring daemons.
- **MPI Job Database** stores the information about the created MPI rings and associated user jobs.



MPI ring life cycle

- **Empty**: first status when a new Ring is started.
- **Accumulating**: the Master Pilot Job is deployed on the site, the Slave Pilot Jobs are joining the ring.
- **Starting**: the Master starts the MPI environment
- **RingInit**: the Slaves start the MPI environment depending of MPI flavor, the ring is completely created and tested.
- **Ready**: the Master starts execution of the matched MPI user job.
- **Running**: the user job is being executed, this state will be valid until the job finishes.
- **Done:** the user job execution successful
- **Failed:** the user job execution failed
- **Out**: all the role variables are reset, MPI daemons are stopped. MPI pilots are ready to start a new cycle or finish



MPI Jobs on the GRID

The DIRAC Project provides a general purpose middleware for multiple user communities. This implies that the middleware should support different kinds of jobs, including MPI and parametric jobs.

In GISELA Latin America Grid approximately 30% of the workload are parallel applications. This revealed certain difficulties:

- Most of the sites were not supporting MPI jobs if even technically apt to do so
- No shared file system between the Worker Nodes (WN) as required by some applications
- Limitations in the number of supported MPI flavors

Even in the sites officially supporting MPI jobs, the execution efficiency stays low.

The DIRAC Project introduced a Workload Management System with Pilot Agents or Jobs which has a high degree of flexibility in managing the user payloads. This concept was extended to support also MPI jobs which opens a lot of interesting opportunities. The Pilot Jobs reserve resources (WNs) for the user parallel jobs without any extra support by the hosting batch systems and prepare the execution environment bringing in the required MPI software.

MPI Agent roles

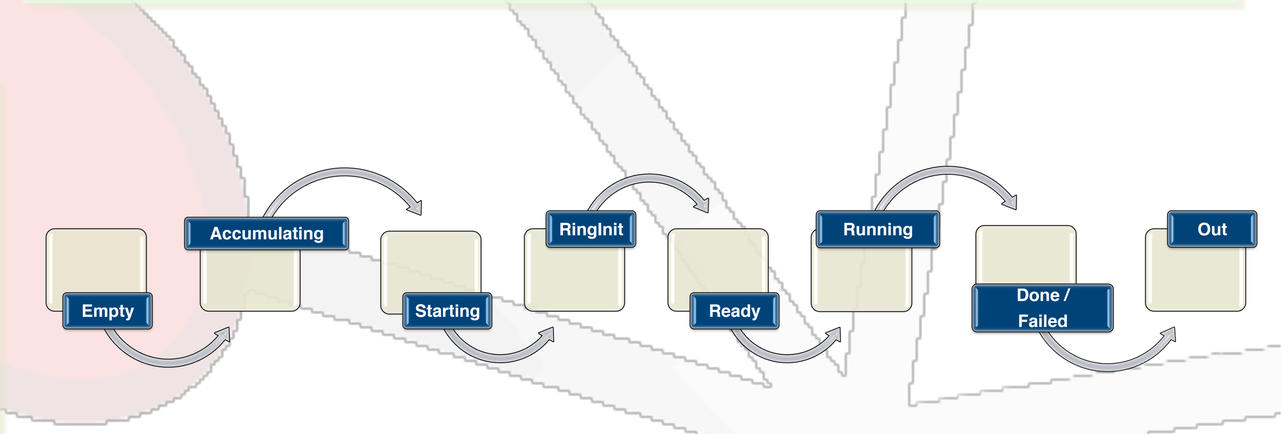
Each MPI Pilot Job can take a role:

- **Master**: the first MPI Pilot Job that starts creation of a new ring on instructions from the MPI Service. At this moment the matching MPI user job status is changed to *Matched*
- **Slave**: the MPI Pilot Job deployed at the same site as the Master adds its WN resource to the MPI ring being accumulated.

Once the MPI ring is complete it will start execution of the user job. The MPI ring is destroyed when:

- The MPI job is executed successfully.
- The MPI job fails.
- The required number of MPI slave nodes is not obtained within a given time period.

After the MPI ring is destroyed, the MPI Pilot Jobs will start the new MPI ring cycle if the remaining time is sufficient, otherwise the WN is freed.



Sharing MPI software and data

The MPI Pilot Jobs are installing the necessary software on the fly. Two mechanism for the software deployment are used:

- **CERNVM-FS** is used by the WLCG Virtual Organizations to deploy their software. It allows access to the remote software repository via HTTP protocol. To use it, the CERNVM-FS should be mounted on the WNs which can not be done from within Pilot Jobs as it requires root priveleges. This limits the number of sites where it can be used.
- **Cooperative Computing Tools (CCTOOLS)** is a package consisting of:
  Chirp – a distributed file system designed to export files for Grid computations**.**
  Parrot is a tool to wrap applications to access remote files with the standard POSIX API using chirp and others protocols like http, grow, ftp, irods, hdfs, cenrvmfs without root privileges.
  CCTOOLS support ACLs in each directory and the use of GSI authentication.

The Chirp server is started by the MPI Master Pilot Job and is mounted by the Slaves providing a dynamically deployed shared file system for the MPI applications.

Usage and outlook

The described mechanism was used successfully in the GISELA Grid with the ABINIT application using 8 processors simultaneously using the MPICH2 MPI flavor.
This work opens new opportunities in using the Grid resources and can be improved in various ways:

- support of various MPI flavors in the MPI Agents;
- adapt more applications in order to find problems and improve the procedures;
- provide support for multi-core WNs improving the times of MPI ring creation especially in the Cloud environments;
- evaluate other ways of distributing the application software. For example, Parrot is supporting actually CERNVM-FS, this combination would allow to use CERNVM-FS as server for DIRAC applications and to use Parrot tools to mount the application repositories dynamically in the WNs.

We would like to acknowledge very valuable help of Dr. Douglas Thain, University of Notre Dame, and also the support by the GISELA project.

Centre de Physique des Particules de Marseille CPPM

V.Hamar, A.Tsaregorodtsev for the DIRAC Project

http://diracgrid.org