EGEE application porting to the EDGeS platform

Characteristics of portable EGEE applications

The EDGeS project receives Community research funding

EGEE vs. DG applications

EGEE applications

- Wide range of applications:
  - Sequential
  - Parallel (MPI, parametric sweeps)
  - Complex workflows
- Shared data storage
- Can be interactive
- High-level services can be used
- Monitoring
- Metadata management
- Portals
- Anyone can submit jobs with a valid certificate (no validation of job executables before submission)

EGEE vs. DG applications

Desktop Grid applications

- Only master/worker or parameter sweep parallelisation
- No shared data storage
- No MPI or internal communication between worker nodes
- Nodes can use the results of other nodes, but only through the server
- Typically long running jobs with small or medium-sized (max. 100 MB per slave) inputs and outputs
- Only trusted applications

EDGeS applications

- EDGeS supports both kind of grids
- Applications must run on both EGEE and DGs
- Characteristics of EDGeS applications are derived from the intersection of the features from the previous slides:
  - Only master/worker or parameter sweep parallelisation
  - No shared data storage
  - No MPI or internal communication between worker nodes
  - Nodes can only use the results of other nodes through the server
  - Typically long running jobs with small or medium-sized (max. 100 MB per slave) inputs and outputs
  - Only trusted applications
- Practically, the application should be able to run on a DG
- If this criterion is met, the application might be an EDGeS application

Applications suitable for EDGeS

- Parameter sweep
  - The size of inputs and outputs should be under 100 MB per slave
  - If they are over 100 MB, multiple data servers and load balancing is necessary
- Master/worker
  - The same size limits apply
- Multi-threaded applications (worker applications that start multiple threads to utilise multi-core CPUs)
- Applications using GPUs (CUDA)

Limitations of job execution times

- To achieve good performance
  - the execution time of individual jobs should be
    - over 10 minutes (otherwise the overhead caused by the DG will reduce the performance)
    - less than 30-60 minutes (if more, application level checkpointing is required to avoid loss of computation caused by user interventions)
  - the execution of different jobs should take around the same amount of time (better scheduling, less load on the server)

Other limitations

- Programming languages
  - APIs available primarily in C and C++ (there is a binding for FORTRAN)
  - Any other languages if the execution environment supports them (either the GenWrapper or BOINC wrapper technology is required in this case)
- Operating systems
  - Depends on the DGs where the application will run
  - It will automatically run on DGs where the Linux platform is supported

Other limitations

- DG resources
  - DG worker nodes have diverse resources (especially public DGs)
  - Applications should minimise their resource usage (primarily RAM and HDD) to maximise the number of nodes involved in the computation

Benefits of using the EDGeS joint platform

- Large computing power is not available or accessible in current scientific e-Infrastructures
- Desktop Grids are easy-to-scale systems and able to collect 1-2 orders of magnitude more compute power
- By interconnecting EGEE and DG systems EDGeS users can transparently execute applications on any arbitrary platform involved in the new infrastructure
- Applications meeting the requirements of the EDGeS platform could utilise much more resources than in the current EGEE infrastructure
- As a consequence we get:
  - reduced turnover time
  - improved fault-tolerance (redundant computing)
  - higher throughput