Evaluation of the Usage of Condor as Batch Queue System for ESO/DMD

DFS/BET – ESO Garching 2005-04-13

Introduction

This document is a follow up of the report "Evaluation of the ESO/DMD Batch Queue System Infrastructure" (March 2005), where it was proposed to investigate the possibility of using Condor (http://www.cs.wisc.edu/condor/) instead of REI as the Batch Queue System for ESO/DMD. The following roadmap had been proposed:

1. use Condor for implementing the Advanced Product Processing infrastructure
2. replace one of the DFO REI installations with Condor
3. participate to the Condor week 14-18 March at the University of Wisconsin (1 person)

For various reasons, among which the unavailability of suitable computing resources, we didn't have the chance to implement point 1; therefore this report is focused on points 2 and 3.

As described in the previous report, the main reasons for evaluating an existing BQS (Condor) as opposed to continuing the development of REI are:

1. Even if REI contains some ESO specific functionalities (integration with CPL), it is evolving towards a generic BQS, of which there are quite a few off-the-shelf solutions on the market, including free packages like Condor.

2. REI works fine, but it still needs quite some development and especially testing before we can consider it an operational system. REI does not provide features like resource management, File Transfer Mechanism, GRID capabilities and others which are available in more mature systems like Condor.

3. The main advantage of Condor is its maturity. Other important features of Condor are:

    a. Resource management.
    b. Support for grid-based computing methodologies and protocols.
    c. Advanced cascade management.
    d. Extensive documentation.
    e. Good support, also at commercial level.

More information about Condor is provided in the next section.

Status of the Condor Project

The Condor project was initiated in 1986 and has since then constantly been improved and enhanced during the last 17 years. There are currently about 30 people involved in the project at its home base, the University of Wisconsin/Madison of which 3-4 are responsible only for testing and integrating the system on about 30 different platforms and architectures, including Enterprise Linux and Scientific Linux (based on Enterprise). Although Condor's core functionality (as batch queue system) is quite stable, there are several areas of new research, like *Stork* (a scheduler for data placement in the grid) or Condor-C (relocation of jobs in the grid). The fact that Condor is

both a production system and a research project is reflected is the version numbering of the Condor releases (the second digit of the release number is odd for development releases, even for stable releases). Major stable releases are scheduled roughly yearly. Minor releases (currently 6.6.x) occur primarily for bug fixes and are unevenly spaced.

The impression of the system is that it has reached maturity and not only research institutes are relying on the system for the batch processing and optimized utilization of computer resources. At the yearly Condor Week in Madison/Wisconsin, several commercial companies were present, which use the system also for large-scale commercial applications. For instance represented at the conference was a company making animations for movie producers like Disney Corporation. To make it possible to carry out the vast amount of renderings needed for this, they have a dedicated cluster of some 1000 nodes all running Condor, which do this processing. Also an American insurance company was present at the conference. This company use Condor in a major pool for their statistical computations and forecasts. They had both used it with great success.

Another important example of the usage of Condor within the scientific world is the CERN GRID project for the LHC:

http://lcg.web.cern.ch/LCG/activities/middleware.html

There are many other commercial and non-commercial users, also probably quite a few, which are not even known to the Condor Team, since it is used for classified projects (for military use and other non-public projects).

Represented at the conference there were also people from many research institutes, which have major 'campus GRID' pools in operation, which harvests unused CPU cycles of many desktop PCs around their campus.

In total there are at present close to 1600 known Condor Pools with some 60000 nodes running Condor world-wide.

Condor provides a wide range of services and features of which many probably are not even interesting for the usage at ESO. However advanced features such as resource management, which makes it possible to assign jobs to the more suitable nodes and to ensure that certain users or types of jobs get the necessary amount of CPU, might be very useful also for ESO.

The File Transferring Mechanism might also be used in certain contexts where it is not possible to have a shared file system for the various nodes in the pool.

Also the GRID capabilities provided by Condor, could be very interesting if an initiative to start a "DMD GRID" or even "ESO GRID" project would be agreed upon, to collect unused computer resources and to be able to do new kinds of processing not possible with a single user desktop PC or with a small cluster of PCs.

There are many other features provided by the Condor system which might be of interest for the usage at ESO. It should also be mentioned, that despite the overall complexity of the system, it is quite easy to set up a Private Condor Pool on a single node and even a pool of nodes with a Central Manager and a number of Submit and/or Execute Nodes.

Condor provides well-defined schemes for installation and for administration. For latter, a palette of tools are provided.

Also the logging facilities of Condor are quite extensive, and Condor insures that the log files produced remotely by a job are transferred back to the Submit Node such that the user can check the status of the job and do troubleshooting if required.

User Support for Condor

The Condor Team provides a free support on 'best effort basis' whereby it is possible to submit questions and problem reports to a dedicated email account. These requests are then assigned within the Condor Team to the most appropriate developer for handling. There is also a number of support schemes offered, where the feedback is provided within a guaranteed time. In this case a fee must be paid, according to the level of support. For instance basic support (Installation and Basic Operation) costs 176 USD per month. For more information, consult:

http://www.cs.wisc.edu/condor/condor-support

So far the free support seems to be satisfactory for our needs. We already submitted a couple of questions, which were answered within a limited time. Apart from that, most of the answers can be found in the extensive documentation provided for Condor, in particular the User's manual (~ 660 pages).

Tests done with Condor

In order to gain experience with Condor, a number of different tests have been carried out. Some of these tests are:

1. Install Condor on a single node (Private Condor) and execute single test jobs and cascades ("DAGs" in the Condor terminology).
2. Install a small Condor Pool with two nodes and execute single test jobs and cascades.
3. Execute single OmegaCam Recipes and cascades in a test pool with two nodes and a shared file system.
4. Execute tests with the file transferring mechanism of Condor.
5. Install Private Condor on a node of the DFO/QC Team and execute recipes on real data.

These tests can't be considered exhaustive (Condor is a complex system), but they cover the main foreseeable use cases for a DMD Batch Queue System. In summary, the test results are positive and motivating for adopting Condor as BQS for ESO/DMD.

Concerning the tests done to use Condor for the DFO/QC environment, it is quite easy to adapt the existing DFO/QC environment to use Condor as BQS as long as a shared file system is used, whereby all nodes have access to a common data area. This is true for any BQS that would be used. The majority of the adaptation could be handled with the environment variables defined for the DFO/QC environment. Only a few changes had to be introduced in the control scripts in use within the DFO/QC.

One of the tests performed was to execute the Sinfoni calibration cascade, consisting of 34 jobs. A graphical representation of the cascade, generated by the system, can be found at the following URLs:

- http://www.eso.org/~jknudstr/BQS/Condor/SINFO-2005-03-14.dot.42-2-sub.jpg
- http://www.eso.org/~jknudstr/BQS/Condor/SINFO-2005-03-14.dot.42-2.jpg

These graphs are generated from the so-called DOT files, which can be created by the Condor DAG Manager. During the execution of the cascade a snapshot is created at each state transition of the DAG. This feature could be used to implement a WEB interface to the system, which can be used by the users to get an overview over the status of their DAGs.

An interesting feature of the cascade handling implemented by Condor is that if a job within a cascade fails, Condor aborts the cascade. However, if the problem with that particular job is subsequently solved, it is possible to request Condor to resume the execution of the cascade, starting from the point where it was interrupted. I.e., all the jobs of the cascade which already executed successfully need not be executed again.

During the tests, most basic features of the system have been used. There does not seem to be any show-stoppers at this point in time, which should prevent us from using Condor.

Conclusion

A close evaluation of the Condor BQS was carried out, according to the roadmap defined in the previous report. A number of tests were carried out to verify various aspects of Condor, e.g. deployment on single node and over a pool of nodes, execution of simple jobs and complex cascades, etc. In particular, the usage of Condor in the DFO/QC environment was successfully demonstrated.

In conclusion Condor seems to provide all features needed now and in the future with the required level of stability. Also the support and documentation provided seems to be sufficient for our needs. We believe that, if we adopt Condor as BQS for ESO/DMD we can meet the present and (foreseeable) future requirements with fewer resources (about half for 2005) compared to continuing the development of REI. In this case, the DFS Back End Team would become the system integrator for Condor, as opposed to developing and maintaining REI.