Automatic Resource Selection Report

Scheduling Working Group

2/2/2009

1. Introduction

This report describes the current status and future plans for automatic resource selection on TeraGrid. We define automatic resource selection as programmatically selecting resources for a job based on user-specified requirements and preferences. Users invoke this capability when they have a job that can run on any of a number of resources and they want a tool to help them meet a scheduling goal such as minimizing turn around time. The Metascheduling Requirements Analysis Team (RAT) identified automatic resource selection as an important capability for TeraGrid users and providing this capability is one of the goals of the current Scheduling Working Group.

The Metascheduling RAT identified and described a number of tools that could potentially be deployed on TeraGrid. The Scheduling Working Group deployed four of these tools for hands-on evaluation: Condor-G with matchmaking, Gridway, Master Control Panel (MCP), and Moab. The purpose of this hands-on evaluation was to select which tools should be moved into production on TeraGrid. The rest of this report will describe these tools, our deployments, our experiences, and our future plans.

2. Summary of Tools

The four tools deployed by the Scheduling Working Group demonstrate a variety of philosophies to providing automatic resource selection. Moab is a commercial tool that supports a variety of different types of scheduling. Condor is a long-term academic project that also supports a range of scheduling functionality. Gridway is an open academic project that is targeted specifically to scheduling jobs in grids of clusters accessed via Globus or similar infrastructure. MCP is an effort aimed at end users - it supports metascheduling using simple mechanisms, with few dependencies on other tools, and can be installed without administrative privileges.

The first thing to note about Condor-G, Gridway, MCP, and Moab is that they perform job management in addition to resource selection. A user of one of these tools submits their job to the tool and the tool will select system(s) for the job, submit the job to that system(s), and then manage the job. The user interacts with the tool to observe and manage their job.

There are some variations in the types of jobs that these tools support. All four metascheduling tools support a job that is the execution of a serial or parallel program. Condor-G, Gridway, and Moab also support job arrays or ensembles – the specification of multiple independent programs to execute. These programs may or may not run at the same time so the purposes of a job array are to allow the user to organize their job submissions and to allow a scheduling tool to optimize

the completion time of an entire array of jobs instead of just the individual jobs. Condor-G, Gridway, and Moab also support workflows where there are dependencies between sub-jobs. The types of dependencies supported by these three tools varies slightly, but they typically take the form of statements such as "execute sub-job C when sub-jobs A and B complete successfully".

Condor-G, Gridway, and Moab operate by selecting a single cluster for a job and then submitting the job to that cluster. This selection is performed using the requirements and preferences provided by the job as well as any requirements or preferences of the cluster. For example, a job may only be able to execute on systems with a specific CPU architecture and prefer systems with faster CPU clock speeds. A cluster may only allow certain users to submit jobs to it. If a job fails to execute on a cluster these tools will move the job to a different cluster. If a job is taking longer than expected to start, Condor-G and Gridway may move the job to a different cluster. This situation doesn't occur with Moab as it is typically also scheduling the cluster, but Moab can move jobs if necessary.

The fact that Moab is also scheduling the cluster offers advantages and disadvantages. An important advantage is that scheduling can be done more efficiently and predictably at the grid (multiple clusters) level. Condor-G and Gridway interface to a cluster scheduling using Globus and, in contrast to Moab, do not have access to information internal to the cluster scheduler or influence over scheduling decisions made by the cluster scheduler. Instead, Condor-G and Gridway rely on information that cluster schedulers make available to users and tools to predict how a cluster scheduler will schedule jobs.

An important disadvantage of Moab is that many TeraGrid systems do not use Moab to schedule their clusters. Even if TeraGrid resource providers were interested in using Moab to schedule their clusters, only a subset of TeraGrid sites are licensed to use Moab at the current time. There is an option to run a Moab daemon that interfaces directly to a (non-Moab) cluster scheduler, but we have not successfully deployed this option. We expect that this approach would have the same advantages and disadvantages as Condor-G or Gridway.

MCP operates in a different manner from the other three tools. A job submitted to MCP is submitted to multiple clusters via either Globus WS-GRAM or using SSH to interact directly with the cluster scheduler. The clusters to submit a job to are either specified by the user or selected based on user-specified constraints and preferences. When the first copy of the job starts to execute, the other copies are terminated.

3. Current TeraGrid Deployments

Condor-G without matchmaking is available to users on 14 of the 15 systems in TeraGrid (it is not installed on the NICS Kraken system). These deployments allow users to submit jobs from these systems to any specific system that has Globus services installed on it. The Scheduling Working Group has deployed Condor-G with matchmaking and Gridway so that these tools can submit jobs to the following systems:

- IU BigRed

- NCSA Abe
- NCSA Mercury (DTF)
- LSU QueenBee
- Purdue Steele
- SDSC DTF
- TACC Lonestar
- TACC Ranger
- UC/ANL DTF

Users can access Gridway from the system tg-t004.uc.teragrid.org if they have an account on the UC/ANL Itanium cluster. Users can currently only access Condor-G with matchmaking from the TACC Lonestar system. However, a small modification to the Condor-G configuration on TeraGrid systems allows users on those systems to access Condor-G matchmaking.

MCP is available on the SDSC and NCSA DTF systems. A test deployment of Moab as a metascheduler was done at NCSA where a submit system that could route jobs to Mercury or Abe. Moab is used as a cluster scheduler on the following TeraGrid systems (they are therefore the natural candidates for inclusion in a Moab metascheduling testbed):

- IU BigRed
- NCSA Cobalt
- NCSA Mercury
- NCSA Abe
- NICS Kraken
- ORNL NTSG
- Purdue Lear
- UC/ANL DTF

In addition to these deployments, the Scheduling Working Group has provided documentation about the installations of Condor-G and Gridway on TeraGrid systems and user guides that describe how to use these software packages on the TeraGrid. The user guides are on the TeraGrid wiki and can be found from the testbeds pages that are linked from the Scheduling Working Group home page at http://www.teragridforum.org/mediawiki/index.php?title=Scheduling_WG.

4. Future Plans

Based on our experiences, available resources, and interests of participants, we have decided to support the production deployment of Condor-G, MCP, and Moab on TeraGrid. Both Condor-G with matchmaking and Gridway are good tools that can be used to perform automatic resource selection and metascheduling on any TeraGrid system. Since there is only one person available to deploy and support these tools, we have decided to initially focus on Condor-G. This choice was made because Condor-G (without matchmaking) is already deployed on many TeraGrid systems, several science gateways are already using Condor-G, and Condor-G provides more flexibility for integration with TeraGrid services.

MCP is developed by SDSC and there is support for continued development and deployment from that site. Moab is of interest to the TeraGrid sites that are already using Moab to schedule their clusters so those sites will continue on the path of using Moab for automatic resource selection.

As we move these tools into production, there are a number of tasks the Scheduling Working Group needs to perform including:

- Creating new or modifying existing CTSS capability kits to support these tools. This includes packaging software for deployment by RPs.
- Developing INCA tests for these tools.
- Creating documentation for the TeraGrid web site.
- Providing support for these tools to TeraGrid users and administrators.

5. Summary

The previous Metascheduling RAT found that there are TeraGrid users that would like to automatically select a system to run their jobs on. This RAT also identified a number of software packages that provide this functionality. The Scheduling Working Group deployed a subset of these software packages (Condor-G with matchmaking, Gridway, MCP, and Moab) to gain hands-on experience with them.

Based on our experience and the interests of the participants, the working group plans to move Condor-G, MCP, and Moab into production on TeraGrid. Condor-G and MCP will be in production in early 2009 and Moab will follow as available effort allows. This effort will include creating any CTSS kits needed, writing test programs, creating documentation, and providing support.

A. Automatic Resource Selection Tools

This section provides detailed information about the four tools for automatic resource selection that we examined.

A.1. Condor-G

1. General information
    a. What is the name of the tool? **Condor-G**
    b. Where is the web site for the tool? **http://www.cs.wisc.edu/condor/**
    c. Cost/Licensing
        i. Is the tool free to use? **Yes.** Is support free? **There is free support (via email lists) and paid support.**
        ii. How is the tool licensed? (GPL, Commercial, etc.)**) Apache version 2**
    d. Code Availability

i. Is the code open-source? **Yes**
ii. Is there a mechanism for the developer to accept code changes from TeraGrid? **Unknown, but TeraGrid has an existing relationship with Condor**

e. Support/Documentation
i. What support does the scheduler developer provide? (24/7, user forums, faqs) **Free support is best-effort email lists. Paid support includes phone and high-priority email access during business hours.**
ii. What is the quality of the documentation? **Very good**
iii. Is the web site for the tool helpful and informative? **Yes**

f. Product Maturity
i. How long has the product been available? **Condor has been available for almost 20 years and Condor-G has been available for about 7 or 8 years.**
ii. What is the production status of the code (prototype/alpha/beta/production)? **Production.**
iii. How many other production grids use this software today? **Many use Condor, a number use Condor-G. TeraGrid uses Condor-G already, Open Science Grid uses Condor-G with matchmaking.**
iv. Approximately how many users of this software are there? **Hundreds or thousands.**
v. Approximately how many developers support this product? **5-15**

2. Functionality. Does the tool support the following functionality at this time? (also indicate if the functionality is planned in the future and a timeline, if known)
a. Automatic Resource Selection.
i. Specifying requirements for a single job? **Yes**
ii. Specifying preferences for a single job? **Yes**
iii. Requirements or preferences for each job that include:
1. System name(s). **Yes**
2. Queue(s) to use on a system. **Yes**
3. Number of CPUs/cores. **Yes**
4. Amount of physical memory. **Yes**
5. CPU architecture. **Yes**
6. Operating system. **Yes**
7. CPU clock speed. **Potentially**
8. System load (fraction of system in use). **Yes**
9. Queue properties (wall time, nodes/CPUs/cores, priority). **Yes**
10. Cost (in terms of allocation). **Potentially**
11. Deadline. **Potentially**
12. List any other properties that can be specified: **TeraGrid can define the information about each system that is available to Condor. A job can use any of this available information when specifying its requirements and preferences. In addition, TeraGrid can define functions that can invoke services such as the batch queue prediction service or the TGCDB and the**

results of these functions can be used by jobs when specifying requirements and preferences.

iv. Specifying requirements for an ensemble of jobs? **Limited – see question vi below.**
v. Specifying preferences for an ensemble of jobs? **No**
vi. Requirements or preferences for an ensemble that include:
    1. Total cost. **No**
    2. Deadline. **No**
    3. Others: **Relative priority of nodes. Maximum number of tasks submitted per user-defined task class. No high-level requirements or preferences such as deadline, completion time, or cost can be specified.**
vii. Specifying requirements for a workflow of jobs? **Limited**
viii. Specifying preferences for a workflow of jobs? **No**
ix. Requirements or preferences for an ensemble that include:
    1. Total cost. **No**
    2. Deadline. **No**
    3. Others: **See vi above.**
x. Using information from TeraGrid information sources? **These sources can operate like databases and be "dumped" into the resource selection tool**.
    1. MDS4 (static and dynamic system characteristics). **Yes, with work**
    2. TGCDB (e.g. user access & allocations). **Yes, with work**
    3. Inca (system and service status). **Yes, with work**
    4. Arbitrary information sources. **Yes. Condor doesn't actually provide information gathering services for use with Condor-G. Condor also doesn't define the information that should be provided about systems used via Globus. TeraGrid can therefore include whatever information it needs, but we must gather the information ourselves.**
xi. Query information from TeraGrid services? These services require input to dynamically generate information and can't be "dumped" into the resource selection tool. **Yes, functions to query services can be written and used by Condor.**
    1. Batch Queue Prediction Service (queue wait times). **Yes**
xii. Mechanisms for not selecting a single system for too many jobs at once? **Yes**

b. Job management (individual jobs).
i. Job submission. **Yes**
ii. Job monitoring. **Yes**
iii. Pre-staging of files. **Yes**
iv. Post-staging of files. **Yes**
v. Capturing stdout and stderr. **Yes**
vi. Fault tolerance.
    1. Re-submission of job to same system. **Yes**
    2. Submission of job to different system. **Yes**

vii. Performance optimization
    1. Submission of job to different system if it is slow to execute. **Yes**
    2. Submission of job to multiple systems. **No**

c. Job management (ensembles). Does the tool support the management of ensembles (sets) of jobs? **Yes**
    i. Performance optimization
        1. Submission of jobs to systems that are executing jobs more quickly. **Yes**
        2. Other: **If jobs are queued too long, Condor will move them to different systems.**
    ii. Describe differences in functionality from the management of individual jobs. **Condor does not seem to manage the jobs in an ensemble differently than if they were all submitted separately.**

d. Job management (workflow). Does the tool support the management of workflows (sets of jobs with dependencies)? **Yes**
    i. Performance optimization
        1. Long-term planning of the workflow. **No**
        2. Adapting to dynamic conditions during workflow execution. **Yes**
        3. Use file locations and sizes when placing jobs on systems. **User may be able to specify this.**
        4. Other:
    ii. Describe differences in functionality from the management of individual jobs. **Aside from managing the dependencies between jobs, none known.**

3. Installation
    a. Were the installation instructions clear? **Yes**
    b. If the tool supports job management, what tools does it use for this (e.g. services such as GRAM, ws-GRAM and/or workload managers such as LSF, PBS, LoadLeveler, SGE). **GRAM and WS-GRAM are applicable to us.**
    c. Does the tool require any modifications to local resource manager? **No.** Are these modifications straightforward?
    d. How long did installation take in hours of work? **A few hours for Condor-G. A few weeks to write and test the software to gather information about TeraGrid systems and insert it into Condor-G.**
    e. What additional software is required in order to support the tool and where must it be installed? For each software dependency, is that software already in CTSS?
        i. Each Teragrid resource (for example, GRAM, MDS). **GRAM or WS-GRAM and information gathering. Globus is already in CTSS.**
        ii. Somewhere on the TeraGrid (for example, MDS, MyProxy) **None.**
        On the same machine as the metascheduler (for example, OS, MySQL) **None on the central Condor-G system, Globus on the systems where users are submitting jobs to Condor-G. Globus is already in CTSS.**
    f. Did you ask any questions of the developers? **Yes.** If so, were the developers responsive? **Response on the condor-users email list within a day.**
    g. What customization was necessary to get the software to work? **Custom information gatherers and also wrote a function to query BQPS.** Was this

customization easy or difficult? **It was relatively straightforward, but took some time.**
Are there installation problems that you expect would occur on many installations? **Some customization of the information gatherers may be required.**

h. For the software components that would be installed by RPs, are there any barriers to installing these components automatically as part of a CTSS kit? **No**

4. Operation
    a. How reliable is the software (failures/week)? **Very reliable**
    b. What failures were encountered? **The failures that have been encountered are related to the information gatherers, rather than Condor-G.**
    c. Does the software provide logging? Yes. Can the amount of logging be adjusted? **Unknown**
    d. What amount of resources are typically used by the software? On what systems? (e.g. central server, login node) (e.g. disk space, physical/virtual memory, CPU time) **Not measured, but qualitatively, the resources used by Condor-G do not seem significant.**
5. User Experience
    a. What is the quality of the user documentation? **Very good**
    b. What client interfaces are provided (GUI, command line, web interface, etc.). **Command line**
    c. For each user interface evaluated (e.g. GUI, API, command line):
        i. Provide the interface name: **Command line**
        ii. Is it well documented? **Yes**
        iii. Is it easy to understand and use? **Yes**
        iv. Are there any changes to the interface that would improve it? **None that I can think of.**
    d. Where any problems encountered (e.g. documentation not matching interface, unimplemented features)? **No**
    e. Are the error messages clear and helpful for debugging problems? **Generally good, but not in all situations. In particular, it can be hard to debug problems with Globus using the information provided by Condor-G.**
    f. What is the average response time of the software? **Less than a second.**
    g. How does the software perform under load? At what amount of load does the software begin to respond slowly? (e.g. twice as slow as unloaded response time) **Untested, but Condor-G has been used at high load by many users.**
    h. For each TeraGrid user helping evaluate: **No user feedback, yet.**
        i. Does this software meet your needs?
        ii. Is this your preferred software for performing advance reservation and/or co-scheduling?
6. Other evaluator comments

I found Condor-G to be a solid tool with a lot of capabilities for performing automatic resource selection and job management. It does require work on the back end to get information about systems into Condor-G, but it also allows us to customize this information so that we can provide what our users need.

A.2. Gridway

1. General information
    a. What is the name of the tool? **Gridway**
    b. Where is the web site for the tool? **http://www.gridway.org**
    c. Cost/Licensing
        i. Is the tool free to use? **Yes.** Is support free? **There is free support (via email lists) and there appears to be paid support.**
        ii. How is the tool licensed? (GPL, Commercial, etc.)**) Apache version 2**
    d. Code Availability
        i. Is the code open-source? **Yes**
        ii. Is there a mechanism for the developer to accept code changes from TeraGrid? **Yes, as a Globus project, there are mechanisms to accept bug fixes and enhancements.**
    e. Support/Documentation
        i. What support does the scheduler developer provide? (24/7, user forums, faqs) **Free support is best-effort email lists. Paid support is unknown.**
        ii. What is the quality of the documentation? **Very good**
        iii. Is the web site for the tool helpful and informative? **Yes**
    f. Product Maturity
        i. How long has the product been available? **2-3 years.**
        ii. What is the production status of the code (prototype/alpha/beta/production)? **Production.**
        iii. How many other production grids use this software today? **There are at least 20-30 different grids that are reported to be using Gridway. No large-scale grids known to the reviewer use Gridway.**
        iv. Approximately how many users of this software are there? **Unknown.**
        v. Approximately how many developers support this product? **3-10**
2. Functionality. Does the tool support the following functionality at this time? (also indicate if the functionality is planned in the future and a timeline, if known)
    a. Automatic Resource Selection.
        i. Specifying requirements for a single job? **Yes**
        ii. Specifying preferences for a single job? **Yes**
        iii. Requirements or preferences for each job that include:
            1. System name(s). **Yes**
            2. Queue(s) to use on a system. **Yes**
            3. Number of CPUs/cores. **Yes**
            4. Amount of physical memory. **Yes**
            5. CPU architecture. **Yes**
            6. Operating system. **Yes**
            7. CPU clock speed. **Yes**
            8. System load (fraction of system in use). **Yes**

9. Queue properties (wall time, nodes/CPUs/cores, priority). **Yes**
10. Cost (in terms of allocation). **No**
11. Deadline. **Yes**
12. List any other properties that can be specified: **See http://www.gridway.org/documentation/stable5.4/user/gridway-user-functionality.html#ResSelectExpr for the full list. Important properties to note are queue characteristics.**

iv. Specifying requirements for an ensemble of jobs? **No**
v. Specifying preferences for an ensemble of jobs? **No**
vi. Requirements or preferences for an ensemble that include:
    1. Total cost. **No**
    2. Deadline. **No**
    3. Others:
vii. Specifying requirements for a workflow of jobs? **No**
viii. Specifying preferences for a workflow of jobs? **No**
ix. Requirements or preferences for an ensemble that include:
    1. Total cost. **No**
    2. Deadline. **No**
    3. Others:
x. Using information from TeraGrid information sources? These sources can operate like databases and be "dumped" into the resource selection tool.
    1. MDS4 (static and dynamic system characteristics). **Yes, with work**
    2. TGCDB (e.g. user access & allocations). **Yes, with work**
    3. Inca (system and service status). **Yes, with work**
    4. Arbitrary information sources. **Yes. Gridway defines an Information Driver interface. We can write a program that implements this interface and the program can interact with an arbitrary information source. An important note is that this interface limits the information that.**
xi. Query information from TeraGrid services? These services require input to dynamically generate information and can't be "dumped" into the resource selection tool. **No, in general. The Information Driver can invoke TeraGrid services, but it only has limited information from Gridway to do so.**
    1. Batch Queue Prediction Service (queue wait times). **Yes.**
xii. Mechanisms for not selecting a single system for too many jobs at once? **Yes**

b. Job management (individual jobs).
i. Job submission. **Yes**
ii. Job monitoring. **Yes**
iii. Pre-staging of files. **Yes**
iv. Post-staging of files. **Yes**
v. Capturing stdout and stderr. **Yes**
vi. Fault tolerance.
    1. Re-submission of job to same system. **Yes**
    2. Submission of job to different system. **Yes**

vii. Performance optimization
    1. Submission of job to different system if it is slow to execute. **Yes**
    2. Submission of job to multiple systems. **No**

c. Job management (ensembles). Does the tool support the management of ensembles (sets) of jobs? **Yes**
    i. Performance optimization
        1. Submission of jobs to systems that are executing jobs more quickly. **Yes**
        2. Other:
    ii. Describe differences in functionality from the management of individual jobs. **None known.**

d. Job management (workflow). Does the tool support the management of workflows (sets of jobs with dependencies)? **Yes**
    i. Performance optimization
        1. Long-term planning of the workflow. **No**
        2. Adapting to dynamic conditions during workflow execution. **Yes**
        3. Use file locations and sizes when placing jobs on systems. **Unknown.**
        4. Other:
    ii. Describe differences in functionality from the management of individual jobs. **Aside from managing the dependencies between jobs, none known.**

3. Installation
    a. Were the installation instructions clear? **Yes**
    b. If the tool supports job management, what tools does it use for this (e.g. services such as GRAM, ws-GRAM and/or workload managers such as LSF, PBS, LoadLeveler, SGE). **GRAM and WS-GRAM are applicable to us.**
    c. Does the tool require any modifications to local resource manager? **No.** Are these modifications straightforward?
    d. How long did installation take in hours of work? **A few hours for Gridway. A few weeks to write and test the software to gather information about TeraGrid systems and create a Gridway Information Driver.**
    e. What additional software is required in order to support the tool and where must it be installed? For each software dependency, is that software already in CTSS?
        i. Each Teragrid resource (for example, GRAM, MDS). **GRAM or WS-GRAM and information gathering. Globus is already in CTSS.**
        ii. Somewhere on the TeraGrid (for example, MDS, MyProxy) **None.**
        On the same machine as the metascheduler (for example, OS, MySQL) **Globus, Java, Berkeley Database. Globus is already in CTSS and requires Java.**
    f. Did you ask any questions of the developers? **Yes.** If so, were the developers responsive? **Response on the gridway-users email list within a day.**
    g. What customization was necessary to get the software to work? **Custom information drivers.** Was this customization easy or difficult? **It was relatively straightforward, but took some time.**
    Are there installation problems that you expect would occur on many

installations? **Some customization of the information gatherers may be required.**

h. For the software components that would be installed by RPs, are there any barriers to installing these components automatically as part of a CTSS kit? **No**

4. Operation
    a. How reliable is the software (failures/week)? **Very reliable**
    b. What failures were encountered? **The failures that have been encountered are related to the information drivers, rather than Gridway.**
    c. Does the software provide logging? **Yes**. Can the amount of logging be adjusted? **Yes.**
    d. What amount of resources are typically used by the software? On what systems? (e.g. central server, login node) (e.g. disk space, physical/virtual memory, CPU time) **Not measured, but qualitatively, the resources used by Gridway do not seem significant.**
5. User Experience
    a. What is the quality of the user documentation? **Good**
    b. What client interfaces are provided (GUI, command line, web interface, etc.). **Command line, API (DRMAA)**
    c. For each user interface evaluated (e.g. GUI, API, command line):
        i. Provide the interface name: **Command line**
        ii. Is it well documented? **Yes**
        iii. Is it easy to understand and use? **Yes**
        iv. Are there any changes to the interface that would improve it? **None that I can think of.**
    d. Where any problems encountered (e.g. documentation not matching interface, unimplemented features)? **No**
    e. Are the error messages clear and helpful for debugging problems? **Generally good.**
    f. What is the average response time of the software? **Less than a second.**
    g. How does the software perform under load? At what amount of load does the software begin to respond slowly? (e.g. twice as slow as unloaded response time) **Untested.**
    h. For each TeraGrid user helping evaluate: **No user feedback, yet.**
        i. Does this software meet your needs?
        ii. Is this your preferred software for performing advance reservation and/or co-scheduling?
6. Other evaluator comments

I found Gridway to be a solid tool with a fair number of capabilities for performing automatic resource selection and job management. One potential problem that I saw is the limited number of variables (six) that can be used in job scripts. This caused problems when I wanted to, for example, specify a system-specific executable, but I could not. A nice feature of Gridway is its modular architecture. This made it relatively straightforward to create a new Information Driver (although that interface limits what we can do). A different scheduler could also be plugged in, even though the default scheduler has a number of different algorithms you can select and configure.

An annoyance that was encountered is that Gridway expects the user proxy to be in /tmp/x509up_u<uid>. This is where the gridway server looks for the proxy. The location of the user proxy in the environment where the user submits (often ~/.globus/userproxy.pem) is not used at all.

A.3. Master Control Program

1. General information
    a. What is the name of the tool? **MCP (Master Control Program)**
    b. Where is the web site for the tool? **http://www.sdsc.edu/scheduler/mcp/mcp.html**
    c. Cost/Licensing
    d. Is the tool free to use? **Yes.** Is support free? **There is free support (via email).**
        i. How is the tool licensed? (GPL, Commercial, etc.)**) University of California**
        ii. **non-commercial licence.**
    e. Code Availability
        i. Is the code open-source? **Yes, for educational, research and non-profit use**
        ii. Is there a mechanism to submit changes? **Yes, email to developer.**
    f. Support/Documentation
        i. What support does the scheduler developer provide? (24/7, user forums, faqs) **Free support is best-effort email.**
        ii. What is the quality of the documentation? **Very basic**
        iii. Is the web site for the tool helpful and informative? **Sufficient for download**
    g. Product Maturity
        i. How long has the product been available? **MCP has been in development for 5 years.**
        ii. What is the production status of the code (prototype/alpha/beta/production)? **Production.**
        iii. How many other production grids use this software today? **Teragrid, at SDSC and NCSA**
        iv. Approximately how many users of this software are there? 1
        v. Approximately how many developers support this product? **1**
2. Functionality. Does the tool support the following functionality at this time? (also indicate if the functionality is planned in the future and a timeline, if known)
    a. Automatic Resource Selection.
        i. Specifying requirements for a single job? **Yes**
        ii. Specifying preferences for a single job? No
        iii. Requirements or preferences for each job that include:

1. System name(s). **Yes**
2. Queue(s) to use on a system. **Yes**
3. Number of CPUs/cores. Potentially, nodes are counted now
4. Amount of physical memory. **Yes**
5. CPU architecture. **Yes**
6. Operating system. Potentially
7. CPU clock speed. Yes
8. System load (fraction of system in use). No
9. Queue properties (wall time, nodes/CPUs/cores, priority). **Yes**
10. Cost (in terms of allocation). **Potentially**
11. Deadline. **No**
12. List any other properties that can be specified: **Any static property can be added to the attribute lists for machines. Numeric properties can be specified with comparators: ==, <=, >=, etc.**

iv. Specifying requirements for an ensemble of jobs? No
v. Specifying preferences for an ensemble of jobs? **No**
vi. Requirements or preferences for an ensemble that include:
1. Total cost. **No**
2. Deadline. **No**
3. Others: Relative priority of nodes.

vii. Specifying requirements for a workflow of jobs? No
viii. Specifying preferences for a workflow of jobs? **No**
ix. Requirements or preferences for an ensemble that include:
1. Total cost. **No**
2. Deadline. **No**
3. Others:

x. Using information from TeraGrid information sources? Not currently
1. MDS4 (static and dynamic system characteristics). No
2. TGCDB (e.g. user access & allocations). No
3. Inca (system and service status). No
4. Arbitrary information sources. No

xi. Query information from TeraGrid services? These services require input to dynamically generate information and can't be "dumped" into the resource selection tool. Not currently
1. Batch Queue Prediction Service (queue wait times). No

xii. Mechanisms for not selecting a single system for too many jobs at once? No

b. Job management (individual jobs).
i. Job submission. **Yes**
ii. Job monitoring. **Yes**
iii. Pre-staging of files. No
iv. Post-staging of files. No
v. Capturing stdout and stderr. No
vi. Fault tolerance.
1. Re-submission of job to same system. No

2. Submission of job to different system. **Yes**

vii. Performance optimization

1. Submission of job to different system if it is slow to execute. **Yes**
2. Submission of job to multiple systems. Yes

c. Job management (ensembles). Does the tool support the management of ensembles (sets) of jobs? No

i. Performance optimization

1. Submission of jobs to systems that are executing jobs more quickly.
2. Other:

ii. Describe differences in functionality from the management of individual jobs.

d. Job management (workflow). Does the tool support the management of workflows (sets of jobs with dependencies)? No

i. Performance optimization

1. Long-term planning of the workflow.
2. Adapting to dynamic conditions during workflow execution.
3. Use file locations and sizes when placing jobs on systems.
4. Other:

ii. Describe differences in functionality from the management of individual jobs. **.**

3. Installation

a. Were the installation instructions clear? **Yes**

b. If the tool supports job management, what tools does it use for this (e.g. services such as GRAM, ws-GRAM and/or workload managers such as LSF, PBS, LoadLeveler, SGE). **MCP uses local job submission or WS-GRAM job submission**

c. Does the tool require any modifications to local resource manager? **No.** Are these modifications straightforward?

d. How long did installation take in hours of work? **About an hour to install and test.**

e. What additional software is required in order to support the tool and where must it be installed? For each software dependency, is that software already in CTSS?

i. Each Teragrid resource (for example, GRAM, MDS). **Optionally, WSGRAM for job submission, if local job submit is not desired.**

ii. Somewhere on the TeraGrid (for example, MDS, MyProxy) **None.**

iii. On the same machine as the metascheduler (for example, OS, MySQL**) If WS-GRAM is to be used, then globus client software needs to be used on the metascheduler machine. SSH, Python, Expect.**

f. Did you ask any questions of the developers? **No.** If so, were the developers responsive?

g. What customization was necessary to get the software to work? Stanzas and machine attributes needed to be added to the authomachine.py file. Was this customization easy or difficult? **It was relatively straightforward, but took**

some time.

h. Are there installation problems that you expect would occur on many installations? **Permissions on the installed files and directories need to be verified.**
i. For the software components that would be installed by RPs, are there any barriers to installing these components automatically as part of a CTSS kit? **No**

4. Operation
    a. How reliable is the software (failures/week)? **Very reliable**
    b. What failures were encountered? **Failures were due to inappropriate file permissions.**
    c. Does the software provide logging? **Yes.** Can the amount of logging be adjusted? **Yes, A log command can be specified to record job submits and cancels. By default, this is /bin/logger -p user.notice -t MCP, but can be overridden by setting the MCPLOGCOMMAND env variable**
    d. What amount of resources are typically used by the software? On what systems? (e.g. central server, login node) (e.g. disk space, physical/virtual memory, CPU time) **Not measured, but qualitatively, the resources used by MCP do not seem significant.**
5. User Experience
    a. What is the quality of the user documentation? **Basic**
    b. What client interfaces are provided (GUI, command line, web interface, etc.). **Command line**
    c. For each user interface evaluated (e.g. GUI, API, command line):
        i. Provide the interface name: **Command line**
        ii. Is it well documented? **Yes**
        iii. Is it easy to understand and use? **No**
        iv. Are there any changes to the interface that would improve it? **None that I can think of.**
    d. Where any problems encountered (e.g. documentation not matching interface, unimplemented features)? **No**
    e. Are the error messages clear and helpful for debugging problems? **Yes, with the –debug flag. This prints each command run and output received.**
    f. What is the average response time of the software? **Less than a second.**
    g. How does the software perform under load? At what amount of load does the software begin to respond slowly? (e.g. twice as slow as unloaded response time) **Untested**
    h. For each TeraGrid user helping evaluate: **No user feedback, yet.**
        i. Does this software meet your needs?
        ii. Is this your preferred software for performing advance reservation and/or co-scheduling?
6. Other evaluator comments

MCP is simple and robust, but does not incorporate dynamic information from the remote machines. It does have a unique strategy (Disneyland) for achieving earliest possible job start times.

- **Best effort support is probably not acceptable for TG production deployment**
- **Documentation is VERY basic and needs improvement**
- **Small number of sites with production deployments (SDSC and NCSA) may be indication little value or poor implementation**
- **Small number of users (1) may be indication little value or poor implementation**
- **Installation instructions could be improved**
- **Length of time to configured and install is much longer than 1 hour because someone not familiar with the product needs to learn about the details before installing and configuring. I would expect that a day or 2 is a more reasonable estimate of the effort required to make the product operational**
- **The developer was very responsive to questions asked and support provided.**
- **One of the configuration files used needs to be synchronized across all sites to reflect changes and there is no automatic mechanism to perform this task currently**
- **MCP could cause excess load on systems if the MCP service is heavily used due to the polling of the status of jobs that are submitted.**
- **MCP could cause the system scheduler to make decisions that are based on potentially inaccurate presumptions (i.e. many MCP jobs could be in the queue influencing the decisions of the system scheduler when those jobs could all be removed at some later time)**
- **Not being able to site any user feedback or experience is not a good sign**

A.4. Moab

1. General information
    a. What is the name of the tool? **Moab**
    b. Where is the web site for the tool? **http://www.clusterresources.com/pages/products/moab-cluster-suite.php**
    c. Cost/Licensing
        i. Is the tool free to use? **No, cost could be shared via a multi site license for all TG resources but that cost is not known at this time.**
        ii. Is support free? **Not with our licensing.**
        iii. How is the tool licensed? (GPL, Commercial, etc.)**) Commercial**
    d. Code Availability
        i. Is the code open-source? **No**
        ii. Is there a mechanism for the developer to accept code changes from TeraGrid? **The existing process is not formal, but TeraGrid does have an existing relationship with (Cluster Resources Inc) CRI**
    e. Support/Documentation
        i. What support does the scheduler developer provide? (24/7, user forums, faqs) **Various levels of support are provided including standard, premier, and 24/7. Support options are described at http://www.clusterresources.com/pages/services/technical-support.php**
        ii. What is the quality of the documentation? **Very good**

iii. Is the web site for the tool helpful and informative? **Yes**

f. Product Maturity
   i. How long has the product been available? **Cluster Resources, Inc was founded in 1995 and incorporated in 2001**
   ii. What is the production status of the code (prototype/alpha/beta/production)? **Production.**
   iii. How many other production grids use this software today? **Many enterprise deployments exist as well as research environment deployments including Teragrid**.
   iv. Approximately how many users of this software are there? **In terms of deployments/sites, on the order of hundreds.**
   v. Approximately how many developers support this product? **Unknown, but presumed to be significant and adequate.**

2. Functionality. Does the tool support the following functionality at this time? (also indicate if the functionality is planned in the future and a timeline, if known)
   a. Automatic Resource Selection.
      i. Specifying requirements for a single job? **Yes**
      ii. Specifying preferences for a single job? **Yes**
      iii. Requirements or preferences for each job that include:
         1. System name(s). **Yes**
         2. Queue(s) to use on a system. **Yes**
         3. Number of CPUs/cores. **Yes**
         4. Amount of physical memory. **Yes**
         5. CPU architecture. **Yes**
         6. Operating system. **Yes**
         7. CPU clock speed. **Yes**
         8. System load (fraction of system in use). **Yes**
         9. Queue properties (wall time, nodes/CPUs/cores, and priority). **Yes**
         10. Cost (in terms of allocation). **No**
         11. Deadline. **Yes**
         12. List any other properties that can be specified: **Moab supports custom properties. That is, any property:value pair can be defined and associated with resources**
      iv. Specifying requirements for an ensemble of jobs? **Requirements can be set up for job templates.**
      v. Specifying preferences for an ensemble of jobs? **No**
      vi. Requirements or preferences for an ensemble that include:
         1. Total cost. **No**
         2. Deadline. **No**
         3. Others: Relative priority of nodes. **No**
      vii. Specifying requirements for a workflow of jobs? **Requirements can be set up for job templates.**
      viii. Specifying preferences for a workflow of jobs? **No**
      ix. Requirements or preferences for an ensemble that include:
         1. Total cost. **No**

2. Deadline. **No**
3. Others: **No**

x. Using information from TeraGrid information sources? **Native interface allows Moab to monitor and manage any resource**.
   1. MDS4 (static and dynamic system characteristics). **Yes, with work**
   2. TGCDB (e.g. user access & allocations). **Yes, with work**
   3. Inca (system and service status). **Yes, with work**
   4. Arbitrary information sources. **Yes. Simple scripts are needed to convert the information into Moab specific job, node, user or queue information or to turn it into generic resources inside Moab.**

xi. Query information from TeraGrid services? These services require input to dynamically generate information and can't be "dumped" into the resource selection tool. **Yes, functions to query services can be written and used by Moab.**
   1. Batch Queue Prediction Service (queue wait times). **Yes**

xii. Mechanisms for not selecting a single system for too many jobs at once? **Yes**

b. Job management (individual jobs).
   i. Job submission. **Yes**
   ii. Job monitoring. **Yes**
   iii. Pre-staging of files. **Yes**
   iv. Post-staging of files. **Yes**
   v. Capturing stdout and stderr. **Yes**
   vi. Fault tolerance.
      1. Re-submission of job to same system. **Yes**
      2. Submission of job to different system. **Yes**
   vii. Performance optimization
      1. Submission of job to different system if it is slow to execute. **Yes**
      2. Submission of job to multiple systems. **Yes**

c. Job management (ensembles). Does the tool support the management of ensembles (sets) of jobs? **Yes**
   i. Performance optimization
      1. Submission of jobs to systems that are executing jobs more quickly. **Yes**
      2. Other: **Unknown**
   ii. Describe differences in functionality from the management of individual jobs. **None**

d. Job management (workflow). Does the tool support the management of workflows (sets of jobs with dependencies)? **Yes**
   i. Performance optimization
      1. Long-term planning of the workflow. **Yes**
      2. Adapting to dynamic conditions during workflow execution. **Yes**
      3. Use file locations and sizes when placing jobs on systems. **Yes**
      4. Other: **Unknown**

ii. Describe differences in functionality from the management of individual jobs. **None**

3. Installation
    a. Were the installation instructions clear? **Yes**
    b. If the tool supports job management, what tools does it use for this (e.g. services such as GRAM, ws-GRAM and/or workload managers such as LSF, PBS, LoadLeveler, SGE). **LSF, Torque, PBSPro, SGE, LoadLeveler, SLURM**
    c. Does the tool require any modifications to local resource manager? **Yes** Are these modifications straightforward? **Yes, the local schedulers need to be disabled.**
    d. How long did installation take in hours of work? **Basic installation is less than an hour. Configuring the site and setting up scripts to pull data in from other sources would take a few days.**
    e. What additional software is required in order to support the tool and where must it be installed? For each software dependency, is that software already in CTSS?
        i. Each Teragrid resource (for example, GRAM, MDS). **None.**
        ii. Somewhere on the TeraGrid (for example, MDS, MyProxy) **None.**
        On the same machine as the metascheduler (for example, OS, MySQL) **Scripts written to pull in external data would need to be accessible to Moab.**
    f. Did you ask any questions of the developers? **Yes.** If so, were the developers responsive? **Yes, they are responsive to tickets, e-mail and direct contact.**
    g. What customization was necessary to get the software to work? **Custom information gatherers for additional information. No customization necessary for most needs.** Was this customization easy or difficult? **Easy**
    Are there installation problems that you expect would occur on many installations? **No**
    h. For the software components that would be installed by RPs, are there any barriers to installing these components automatically as part of a CTSS kit? **No**

4. Operation
    a. How reliable is the software (failures/week)? **Very reliable**
    b. What failures were encountered? **Most of the failures we have encountered have been related to the development of new features in beta software.**
    c. Does the software provide logging? **Yes.** Can the amount of logging be adjusted? **Yes.**
    d. What amount of resources are typically used by the software? On what systems? (e.g. central server, login node) (e.g. disk space, physical/virtual memory, CPU time) **Disk space for logging depending on how the logs are configured, otherwise, negligible demands on the system.**

5. User Experience
    a. What is the quality of the user documentation? **Very good**
    b. What client interfaces are provided (GUI, command line, web interface, etc.). **GUI, Command line, and web interface**
    c. For each user interface evaluated (e.g. GUI, API, command line):

i. Provide the interface name: **GUI**
ii. Is it well documented? **Yes**
iii. Is it easy to understand and use? **Yes**
iv. Are there any changes to the interface that would improve it? **None that I can think of.**

d. For each user interface evaluated (e.g. GUI, API, command line):
i. Provide the interface name: **API**
ii. Is it well documented? **Yes**
iii. Is it easy to understand and use? **Yes**
iv. Are there any changes to the interface that would improve it? **None that I can think of.**

e. For each user interface evaluated (e.g. GUI, API, command line):
i. Provide the interface name: **Command line**
ii. Is it well documented? **Yes**
iii. Is it easy to understand and use? **Yes**
iv. Are there any changes to the interface that would improve it? **None that I can think of.**

f. Where any problems encountered (e.g. documentation not matching interface, unimplemented features)? **No**

g. Are the error messages clear and helpful for debugging problems? **Usually very good**

h. What is the average response time of the software? **Less than a second.**

i. How does the software perform under load? At what amount of load does the software begin to respond slowly? (e.g. twice as slow as unloaded response time) **As per conversation with vendor technical staff the software performs very well under load and is deployed and used at other large sites. For example, other deployments handle submission of 100s of jobs per second.**

j. For each TeraGrid user helping evaluate: **No user feedback**
i. Does this software meet your needs? **N/A**
ii. Is this your preferred software for performing advance reservation and/or co-scheduling? **N/A**

6. Other evaluator comments

Moab probably offers more capability and flexibility than any other computer batch system scheduling software on the market. It has been deployed and used successfully at several large enterprise and governmental sites to manage individual systems as multiple systems in the mode of auto resource selection configuration. It is the evaluator's opinion that with a well coordinated effort among RP sites and the development of, and agreement to, the appropriate usage policies Moab would be an excellent solution to TeraGrid's needs for an auto resource selection capability. In addition to the capability provided to users, Moab offers excellent administrative capabilities to track, manage, and report on job scheduling and execution tasks. A potential obstacle of a wide spread deployment of Moab for TeraGrid resources is the cost of licensing. However, the commercial quality support associated with the licensing cost could easily justify the cost of the software licensing. Regardless of licensing cost, which could probably be negotiated to be a reasonable given the exposure to CRI that would result in the deployment of Moab across all TeraGrid

resources, Moab should be strongly considered as a potential implementation that will meet TeraGrid's current and future needs.