

Metascheduler Query and Reservation Interface

Status of this Draft

This document specifies a best current practices for the grid scheduling community. Discussion and suggestions are requested. Distribution of this memo is unlimited.

Copyright Notice

Copyright © The Grid Forum (2000). All rights reserved.

1. Introduction

Information discovery is vital to metascheduling and grid computing. A metascheduler must be able to gather information from local schedulers to determine where to submit a job and/or make an advance reservation. This document is intended to specify a standard format for querying and requesting an advance reservation. It does not describe the format of the arguments. Two documents must be written before defining the argument and protocol format. The first is a document describing the tokens and the values they can take. Lastly, the Scheduling Working Group needs to come to consensus on the format of the queries and commands. The Maui scheduler currently uses a semicolon-separated list of token equals value pairs. Other systems use an LDAP form. It has also been suggested that an XML format would be appropriate.

There are two types of queries as specified by the Grid Forum Scheduling working group. The queries answer the questions: "When will my job start?", and "When is the earliest time that I can make a reservation for my job?". The first query returns the earliest time that the job would start if it were submitted immediately. There is no reservation associated with the query and the job start time is not guaranteed.

Advance reservations guarantee the start time of a job. This guarantee applies as long as a higher priority job does not preempt the job's reserved resources. The local scheduler's response to this query is a range of times in which the reservation could start. To assist metascheduling decisions for user interaction and co-allocation more advanced local schedulers return a list of reservation start time ranges. The range list allows a metascheduler to gather information from multiple local schedulers and compute an intersecting set of ranges to start the job at coinciding times on multiple machines.

This document specifies the general API format for the queries and reservation request. It does not specify the exact format of the arguments. To define the argument format, a list of identifiers must be defined and the format for specifying the identifiers and their values must also be defined.

In all cases:

- Job and reservation start times are specified in Unix epoch time and are based on GMT.
- OtherInfo is a local scheduler specific string that may be used for advanced and/or local system specific features.
- Arguments surrounded in {} are optional.
- The UserID is always provided so that the local scheduler can enforce per-user quotas and policies
- ResKey is returned by the local scheduler so that a metascheduler can securely claim a local reservation when the job is ready to be staged.

2. Job-Start Query

This query is meant to ask the question, "When will my job start?". This query will be used by metaschedulers to find the machine that can run the job the soonest. In this situation, the job will be submitted to and run wholly on a single machine.

QUERY UserID TimeNeeded ResourceList

UserID

Specifies the local user for whom the query is being made.

TimeNeeded

Specifies the amount of time requested.

May be specified in wallclock time or CPU time depending on local scheduler requirements. The units of time used must be indicated in the request.

ResourceList

Specifies the resources required for the job.

Local Scheduler Returns

StartTime {OtherInfo}

StartTime

Specifies the time the job would start. Time is specified in Unix epoch time GMT.

OtherInfo

Local scheduler specific character string. Used to specify local system specific information that may be used by metaschedulers to enhance metascheduling decisions.

3. Advance Reservation Query

In this case, the query is intended to determine the times a reservation could be guaranteed by a local scheduler. This query could be used in single system scheduling as well as co-allocation of multiple systems. Returning a list of start time ranges assists in co-allocation to determine a coinciding time in which a reservation could be made across multiple systems.

RESQUERY UserID TimeNeeded {TimeSpec} {ResAccess} ResourceList

UserID

Specifies the local user for whom the query is being made.

TimeNeeded

Specifies the amount of time requested.

May be specified in wallclock time or CPU time depending on local scheduler requirements. The units of time used must be indicated in the request.

TimeSpec

Specifies time constraints that must be met.

ResAccess

List of existing local system reservations this request has access to. The resulting reservation may intersect with any of the reservations listed.

ResourceList

List of required resource types and amounts.

Local Scheduler Returns

STime EndSTime {OtherInfo} {{STime EndSTime {OtherInfo}} ...}

STime

Specifies the beginning of a range of time in which the job could start. STime is specified in Unix epoch time GMT.

EndSTime

Specifies the beginning of a range of time in which the job could start. EndSTime is specified in Unix epoch time GMT.

OtherInfo

Local scheduler specific character string. Used to specify local system specific information that may be used by metaschedulers to enhance metascheduling decisions.

4. Advance Reservation Query coupled with Reservation Request

Often, a user simply wants to make a reservation on a system at the earliest possible time. This query is identical to the reservation query except that a reservation will be made on the local system at the earliest possible time.

RESQANDSET UserID TimeNeeded {TimeSpec} {ResAccess} ResourceList

UserID

Specifies the local user for whom the query is being made.

TimeNeeded

Specifies the amount of time requested.

May be specified in wallclock time or CPU time depending on local scheduler requirements. The units of time used must be indicated in the request.

TimeSpec

Specifies time constraints that must be met.

ResAccess

List of existing local system reservations this request has access to. The resulting reservation may intersect with any of the reservations listed.

ResourceList

List of required resource types and amounts.

Local Scheduler Returns

SETRES SuccessOrNot {StartTime} {ResKey} {OtherInfo}

SuccessOrNot

Boolean value indicating whether or not the requested reservation was successfully created.

StartTime

Specifies the time the reservation will start. Time is specified in Unix epoch time GMT.

ResKey

A data block that identifies the reservation and is sent, along with the full job description, when submitting the job. The ResKey should be unique and secure. It should be immune to sniffing and replay attempts.

OtherInfo

Local scheduler specific character string. Used to specify local system specific information that may be used by metaschedulers to enhance metascheduling decisions.

5. Requesting a Reservation

Once reservation information has been gathered, a start time decision can be made. The following command is then used to request a particular reservation. The local scheduler returns a boolean indicating success or failure and a reservation key. The reservation key identifies the reservation and allows access to the reserved resources.

```
SETRES UserID TimeNeeded StartTime {ResAccess} ResourceList
```

UserID

Specifies the local user for whom the query is being made.

TimeNeeded

Specifies the amount of time requested.

May be specified in wallclock time or CPU time depending on local scheduler requirements. The units of time used must be indicated in the request.

StartTime

Specifies the time the job should start. Time is specified in Unix epoch time GMT.

ResAccess

List of existing local system reservations this request has access to. The resulting reservation may intersect with any of the reservations listed.

ResourceList

List of required resource types and amounts.

Local Scheduler Returns

```
SETRES SuccessOrNot {ResKey} {OtherInfo}
```

SuccessOrNot

Boolean value indicating whether or not the requested reservation was successfully created.

ResKey

A data block that identifies the reservation and is sent, along with the full job description, when submitting the job. The ResKey should be unique and secure. It should be immune to sniffing and replay attempts.

OtherInfo

Local scheduler specific character string. Used to specify local system specific information that may be used by metaschedulers to enhance metascheduling decisions.

6. Author's Addresses

Quinn O. Snell
Computer Science Department
2218 TMCB
Brigham Young University
Provo, UT 84602
Tel: (801)378-5098
Fax: (801)378-7775
snell@cs.byu.edu

Mark Clement
Computer Science Department
2216 TMCB
Brigham Young University
Provo, UT 84602
Tel: (801)378-7608
Fax: (801)378-7775
Clement@cs.byu.edu