GWD-I (ggf-draft-sched-graap-2.0)
Advance Reservations: State of the Art
http://www.fz-juelich.de/zam/RD/coop/ggf/graap/graap-wg.html

Editor:
Jon MacLaren

6th June 2003

Advance Reservations: State of the Art

(draft)

Status of this Memo:

This draft provides information to the community regarding the status of support for advance reservation within existing scheduling systems. It does not define any standards or technical recommendations. Distribution of this document is unlimited.

Copyright Notice:

Copyright © Global Grid Forum (2003). All Rights Reserved.

Abstract

This document surveys advance reservation functionality in batch scheduling systems. It considers this functionality from the client's perspective, where the client may be a user or a super-scheduler. This document will be frequently updated, and will track the development of advance reservation support in the scheduling community.

GLOBAL GRID FORUM

office@gridforum.org
www.ggf.org

Full Copyright Notice

Copyright © Global Grid Forum (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works

that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the GGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the GGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the GGF or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE GLOBAL GRID FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

Contents

Advance Reservations: State of the Art

1 Introduction

This is a document which is designed to catalog the advance reservation functionality which is available in current scheduling systems. There are possibly some schedulers still not covered - if you know of one, please contact us and we can look at updating and re-releasing this document. We're not interested in adding dead schedulers like NQS and NQE though.

The structure of this document is as follows. Section 2 makes some definitions which are used in the rest of the document. Section 3 elaborates the properties that an advance reservation might have, divided into different categories. Section 4 lists, and briefly describes, the schedulers which are covered in this document. Finally Section 5 provides a capability table showing which schedulers can provide reservations with which properties. This document has had several contributors; Author Information can be found at the end of the document.

2 Definitions and Terminology

User

A user is simply a requestor of the reservation. This could be a human (via any kind of interface: command line, GUI, etc.) or a piece of software (e.g. a superscheduler).

Advance Reservation

This definition has being discussed at some length at meetings of GRAAP-WG, as well as on the mailing list. This reflects the (possibly final) agreed definition.

"An advance reservation is a possibly limited or restricted delegation of a particular resource capability over a defined time interval, obtained by the requester from the resource owner through a negotiation process."

Example resource capabilities: number of processors, amount of memory, disk space, software licences, network bandwidth, etc.

It is worth noting that this definition is more detailed than the one used in the scheduling dictionary WG:

"The process of requesting various resources for use at a later time."

States of Advance Reservation

The states given below are used later to help define the properties of advance reservation handling.

Requested or In Negotiation:	A user has requested, or is negotiating, a set of resources for a reservation. If the reservation is accepted/agreed, it goes to being booked. Otherwise, it becomes declined.
Declined:	The reservation is not successfully allocated for some reason. (Not sure if this state is useful.)
Booked:	A reservation has been made, and will be honoured by the scheduler. From here, the reservation can become active, or be cancelled by the user or system, or be altered.
Booked, change requested/in renegotiaion:	A user is trying to alter the resources for the reservation prior to its starting. Goes back to booked state on success or failure.
Cancelled:	A user cancels the reservation prior to beginning. Or, the scheduler cancels the reservation prior to it's beginning (this may be due to maintenance downtime being scheduled).
Active:	The reservation has started, but not ended.
Terminated:	A user, or possibly the system, terminates an active reservation before the end-time. (The system may want to do this if the reservation becomes active, but is then idle for a certain length of time.)
Completed:	The reservation continued until its end-point.
Active, change requested/in renegotiaion:	A user is trying to alter the resources for the reservation after the reservation has become active. Goes back to active state on success or failure.

3 Properties of Reservations

A list of properties of reservations is given below. They're grouped into Basic, Advanced and Two-Phase Commit sections.

There are other issues with using advance reservations, like "can/does the scheduler pre-empt to honour a reservation". These are local site issues which may change under time and which are in most cases not visible to users or agents negotiating time slots.

3.1 Basic

The scheduler supports Advance Reservations

Where the answer to this question is no, the scheduler should not be documented in any detail (i.e. description or overview only). Where the answer is "yes", I'm assuming that it is possible to cancel a booked reservation, or to abort an active reservation.

Note that at the moment, this document does not distinguish the ability to make a reservation for a precise time from the ability to establish a more general Service Level Agreement for the job, which might state ealiest acceptable start time and latest acceptable end time. Currently, we know of no scheduler which permits this, so there is no distinction to make.

Reservations can be made by "normal" users

For some schedulers, reservations can only be made by an administrator. This is not general enough for everyday use of the Grid.

Reservations (can) have their own queue(s)

This is about how work is scheduled within the reservation. Creating a queue to which the users of the reservation can submit jobs seems to be a common approach. If the reservation does have a queue, then jobs can probably be queued to a reservation as soon as the reservation becomes booked.

Interactive jobs can be run in the reservation

If the reservation has been set aside for users, can they do interactive work in this space, i.e. debugging, command line interaction. I'm thinking of work that is submitted immediately to the resource during the time of the reservation. (This allows the users a way of booking time for the debugging large parallel jobs. Even for small parallel jobs, it's much cleaner than competing in a limited set of resources set aside for interactive work.)

So, for example, with EASY/COSY, upon submitting a request for resources, the user decides whether his job should be run interactive or as a batch job without further intervention. In the later case the user has to submit a script as usual, in the first case he may login to the granted nodes interactively.

ACL provided for controlling the use of the resources in the reservation

When the reservation is set up, is it possible to allow other users (or groups of users) permission to submit jobs to (or interactively use) the reservation?

ACL(s) provided for the modifying/cancelling/terminating of the reservation

Can the reservation be set up so that users other than the reservation's owner can modify, cancel or terminate the reservation?

3.2 Advanced

A booked reservation can be altered

Can the start-time, end-time, number of processors, etc. of a booked reservation be changed after the booking is made?

An active reservation can be altered

Can the resources of the reservation still be altered after a reservation has become active? A particularly interesting example would be a calculation which noted that it was running out of time, and decided to try and extended the reservation by requesting either more processing elements, or more time. (Sounds like something the Cactus team would try to use.)

3.3 Two-Phase Commit

For meta-scheduling on multiple machines to work well, it must be possible for an advance reservation system to support two-phase commit, where there is negotiation before the reservations are fixed. So the following sort of dialog occurs:

> **User:** "Can I have 32 processors at 1:00pm?"
> **System:** "No, there are only 20 free"
> **User:** "Can I have 16 processors at 1:00pm?"
> **System:** "Yes, but confirm within the next 60 seconds if you want them"
> **User:** "I'll take them"

Obviously this is pretty sophisticated. It would still be negotiation if the system only answered "no" in response to the first question. But we do need the last part, where an offer is made with a time limit. This is required so that a super-scheduler can negotiate at several resources in order to co-schedule them.

A reservation can be negotiated

Does the scheduler permit a user (human or agent) to negotiate the reservation in the way described above?

Negotiation can be used to alter booked reservations

If negotiation were possible when booking, I'd expect the same style of negotiation to be possible when altering booked reservations (if permitted at all).

Negotiation can be used to alter active reservations

If negotiation were possible when booking, I'd expect the same style of negotiation to be possible when altering active reservations (if permitted at all).

4 Scheduling Systems

4.1 LSF

Load Sharing Facility from Platform. Not open source or free.

4.2 PBSPro

Portable Batch System (Professional edition) from Veridian. Not open source or free.

4.3 OpenPBS

OpenPBS is an open source and free version of PBSPro. It lacks some of the more advanced features of PBSPro, including advance reservation. Still has an interface for an external scheduler, such as Maui/Catalina.

4.4 Paderborn CCS

Paderborn have asked us to remove CCS from this document. Information will be added again once CCS becomes publicly released.

4.5 LoadLeveler

Proprietary scheduler from IBM. Does not currently provide advance reservations without an external scheduler.

4.6 Maui (Plugs into OpenPBS, PBSPro, LoadLeveler, LSF or Sun GridEngine)

Open Source scheduler that plugs into existing batch system to enhance functionality. Provides backfilling and advance reservation.

4.7 Sun GridEngine

Sun[tm] ONE Grid Engine (SGE) is a distributed resource management and job queuing system. Version 5.3 is issued under a standard Sun Binary Code License, and can be downloaded freely at http://wwws.sun.com/software/gridware/

SGE does not provide directly advanced reservation. For that, the MAUI-SGE integration has to be used. In that configuration, Maui is used as an external scheduler for SGE. SGE manages the job queue and the compute resources while Maui directs SGE to manage jobs in accordance with specified Maui policies, priorities, and reservations. More information can be found at http://supercluster.org/documentation/maui/sgeintegration.html

4.8 EASY (SCAI flavour)

Research prototype scheduler derived from the Argonne EASY scheduler for the SP2. Ongoing work at the Institute for Algorithms and Scientific Computing (SCAI). Supports both batch and interactive jobs. Available and used for several parallel machines and Unix-Clusters. Public domain.

4.9 COSY (NEC)

Research prototype scheduler derived from the SCAI EASY scheduler. Ongoing work at NEC's C&C Research Laboratory at Sankt Augustin. Similar features to EASY, but based on C++ and Corba. In addition COSY is able to interact with other COSY-like schedulers for co-allocation of resources. Supports both batch and interactive jobs. In use at the lab for different PC-Clusters. Binaries available for academic institutions without costs (some kind of contract necessary), source code not available.

4.10 Catalina (Plugs into OpenPBS or LoadLeveler)

Freely available for educational, research and non-profit purposes (See accompanying COPYRIGHT notice). Very similar to Maui in design, but written in Python. Lacks Maui's fairshare, workload profiling, multiple jobs/node. Has short pool, more flexible job filtering by reservations. Designed to support parts of the GridForum Advance Reservation API. See http://www.sdsc.edu/catalina for more information.

4.11 Condor

Condor (http://www.condorproject.org) is a distributed resource management system developed at the University of Wisconsin. It was designed to execute compute intensive applications on desktop or dedicated resources. Because of its support for non-dedicated resources, Condor does not presently support Advance Reservation because the resources cannot be relied on to be present at the time the reservation becomes active.

5 Capability Table

Taking the properties from Section 3, and the schedulers from Section 4, the following table has been produced. I've only used "Y" and "N". Blank means "don't know" rather than "No" (using question marks for "don't knows" makes things too cluttered). Only schedulers which permit advance reservations appear in this table.

To save space, I've used AR for Advance Reservation.

Just to clarify things, this table refers to what is possible for normal users, rather than administrators, as that is the perspective we're interested in. When I say "what is possible", I mean what the system can be configured to allow rather than what is there by default - so if it's possible to configure the scheduler to allow advance reservation by normal users, it's a "Y".

Scheduler -> Property	**LSF**	**PBS Pro**	**Maui**	**Catalina**	**EASY**	**COSY**
Supports AR	Y	Y	Y	Y	Y	Y
AR by normal users	Y	Y	Y	Y	Y	Y
AR has queue	Y	Y	Y	N[1]	Y	Y
AR supports interactive jobs	Y	Y[2]	Y	Y	Y	Y
ACL for use of AR	Y	Y	Y	N		
ACL(s) for mod./ canc. of AR		Y	Y	N		
Can alter booked AR	N	N	Y	N		
Can alter active AR	N	N	Y	N		
AR can be negotiated (2-phase commit)	N	N	Y	N	Y	Y
Booked AR can be re-negotiated (2-phase commit)	N	N	Y	N	Y	Y
Active AR can be						

re-negotiated (2-phase commit)	N	N	Y	N		

Fig. 1: Advance Reservation Capability Table

Notes:

1. In answer to this question, KY stated "No. Jobs can be explicitly bound to a set of reservations. Reservations can be explicitly bound to a set of jobs. Or reservations can allow any jobs by a given user."
2. CL pointed out that "I don't think one would really want to submit an inteactive job until the reservation transitions to ACTIVE, since qsub would be consuming a TCP connection for work that has no chance of starting immediately."

The rows in this table are not definitive. It may be that they need to be expanded upon, and re-categorised as time goes on, resulting in a new version of this document.

6. Security Considerations

No security considerations have been identified yet.

Author Information

Authors and Contributors:

Initials	Name	Institution	Area of Contribution
NF	Nathalie Furmento	Imperial College	Sun Grid Engine content
DJ	Dave Jackson	Supercluster Development Group	Maui content
CL	Casimir Lesiak	Veridian	PBSPro and OpenPBS content
BL	Bingfeng Lu	Platform Computing	LSF content
JM	Jon MacLaren	Manchester University	Struucture, editing, general content
JP	Jim Pruyne	Hewlett Packard Laboratories	Condor content
VS	Volker Sander	Forschungszentrum Jülich	Structure, editing, general content
KY	Kenneth Yoshimoto	San Diego Supercomputing Centre	Catalina content
WZ	Wolfgang Ziegler	Fraunhofer Institute for Algorithms and Scientific Computing	Structure, editing, general content

Contacts for document:

To contribute to this document, send an e-mail to Jon MacLaren.
To comment on this document, you can mail the GRAAP-WG list.
The GRAAP-WG website is at http://www.fz-juelich.de/zam/RD/coop/ggf/graap/graap-wg.html

Document Change Control

Version 1 - 28th July 2002

- Source was Word documented, circulated as PDF.
- Released to GRAAP-WG before GGF5.

Version 2 - 3rd October 2002

- Converted document to HTML.
- Used XML-style markup on any comments not fitting with narrative of the document.
- Basic clarifications, incl. replacing "Advanced Reservation" with "Advance Reservation".
- Table uses Y/N, not ticks and crosses.
- Added contributions on Maui and Catalina.
- Added list of contributers/editors.
- Clarified what negotiation means in the table.
- Added document change control section.

Version 2.1 - 4th October 2002

- Corrected Maui table content.
- Updated document into correct GGF format.

Version 3 - 12th February 2003

- Removed Paderborn CCS Information as requested (19th December).
- Added descriptions of Sun GridEngine and Condor.
- Added PBS and OpenPBS information.

Version 4 - 6th June 2003

- Rearranged document format to match recent GGF draft style.
- Tidied up document, removed unresolved comments, etc.
- Corrected document number, in line with new standard.