A WS-Agreement based negotiation protocol

Oliver Waeldrich, Fraunhofer Institute SCAI
Wolfgang Ziegler, Fraunhofer Institute SCAI

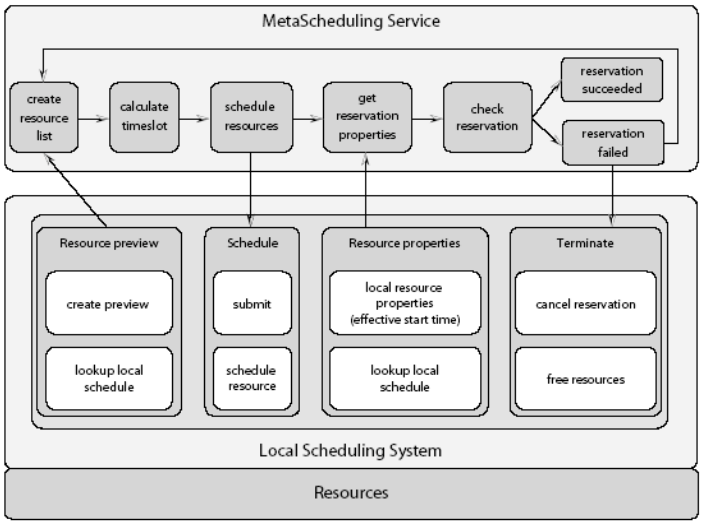
Introduction

To successfully execute distributed applications or workflows, usually different resources like compute nodes, visualization devices, storage devices, or network connectivity with a defined QoS are required at the same time or in a certain sequence. Orchestrating such resources locally within one organization represents only a minor task, whereas the orchestration of resources on a Grid level requires a service that is able to solve the same problems in an environment that may stretch across several administrative domains. Additional conditions have to be taken into account, like the compliance with site-specific policies or the protection of a site's autonomy. In the VIOLA project [4] we have developed such a meta-scheduling service (MSS), which is capable to co-allocate different types of resources (currently compute resources and network resources) in multiple administrative domains.

The VIOLA approach

To achieve the co-allocation of different resources the MetaScheduling Service (MSS) communicates with the selected resource management systems (RMS) through a set of adapters. These adapters provide a uniform interface to the MSS and may implement missing functionalities of the RMS. At the first step of the co-allocation process the MSS queries the adapters of the selected RMS to get the earliest time the requested resources will be available. This time possibly has to be after an offset specified by the user. The adapters acquire a preview of the local resource availability from the individual scheduling systems. Such a preview comprises a list of time frames during which the requested resources (e.g. a fixed number of nodes) can be provided. It is possible that the preview contains only one entry or even zero entries if the resource is fully booked within the preview's time frame. Based on the preview the adapter calculates the next possible start-time.

These start times are sent back to the MetaScheduling Service. If the individual start times match, the MSS will try to reserve the resources at the computed start time via the adapters, making use of the advance reservation capability of the local RMS. If the individual start times do not match, the MetaScheduling Service uses the latest possible start time indicated by the RMS as start time for the next scheduling iteration. The process is repeated until a common time frame is found or the end of the preview period for at least one of the RMS is reached. The latter case generates an error condition.



In case the MSS was able to find a common timeslot and reserve the resources, it afterwards checks the scheduled start times of each reservation. This step is necessary because after negotiating the common start time, other reservations may be submitted by other users or processes to the local RMS, preventing the scheduling of the reservation at the requested time. If the MetaScheduling Service detects one or more reservations that are not scheduled at the requested time, all reservations will be cancelled. The latest effective start time of all reservations

will be used as the earliest start time for the co-allocation try, and the MSS will try again to negotiate of a common timeslot as described.

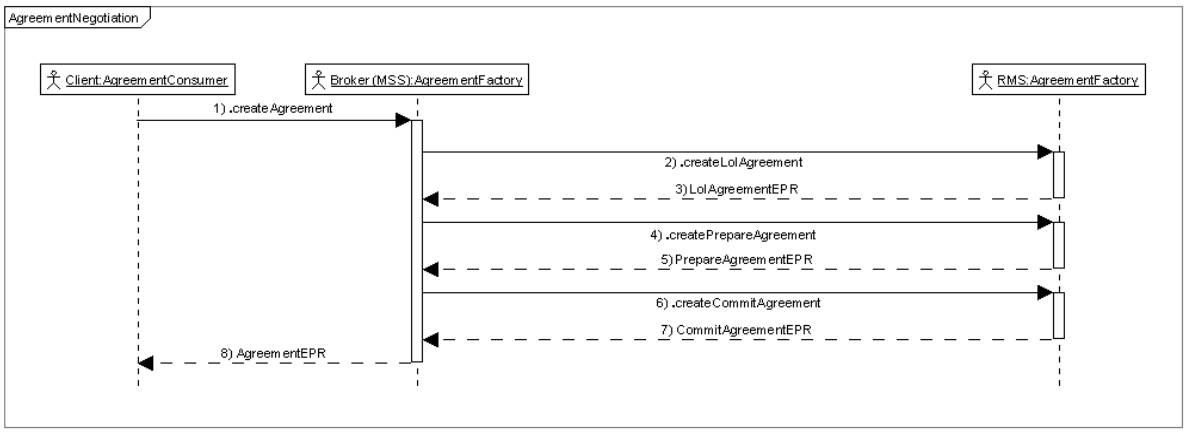
A WS-Agreement based negotiation protocol

Due to the lack of native support for negotiation in WS-Agreement, we have developed a simple negotiation protocol in the VIOLA, as described in the previous section. But since the advance reservation of resources already represents a kind of *Service Level Agreement* (SLA) creation, we propose a new approach to overcome the limitations of WS-Agreement, by using a three-phase-commit protocol. Three-phase-commit-protocols (3PCP) have been already discussed in the past [1], and were said to have restrictions in performance and fault-tolerance [TODO: Referenc???]. However, the coordination of resource reservations, in particular the co-allocation of resources using advance reservation, implies difficult problems, and 3PCP's promise to solve at least some of the problems.

WS-Agreement 3PCP approach

In order to negotiate co-allocation of resources effectively and efficiently, a brokering service must be able to (i) identify a comment timeslot, and (ii) to allocate resources at a specified time. The identification of a common timeslot can already be done by using WS-Agreement templates for publishing the availability of local resources. Furthermore, it seems to be feasible that agreements can be created while specifying a start time for a resource usage within an agreement offer. However, since WS-Agreement only allows creating agreements within on step (createAgreement), one can not be sure whether the creation of an agreement on multiple sites succeeds or not. Additionally, penalties may be associated with an agreement, preventing a broker to simply terminate all agreements and re-negotiate, if one site fails to create an agreement at the proper time.

To overcome these problems we propose the creation of different types of agreements within a negotiation process. These types are a *Declaration of Intention Agreement*, a *Preparation Agreement*, and a *Commitment Agreement*. All of these agreements are normal WS-Agreements as specified in [5], following a certain naming convention. Subsequent agreements (Declaration of Intention Agreement → Preparation Agreement; Preparation Agreement → Commitment Agreement) reference the prior created agreement using wsag:Service Reference, which contains an EndpointReference of the related agreement. The negotiation process is described in the below picture.



Declaration of Intention (DoI Agreement)

A Declaration of Intention (DoI) indicates whether a service provider is able to accept the agreement offer or not. This declares at least that an agreement responder is willing to provide resources according to an agreement offer. The decision, whether to accept a DoI offer or not, depends e.g. on the requested resources (can the requested resources be provided), the requested QoS levels (can the requested guarantees be fulfilled), and the requested start time (are the resource available at the requested time). If an agreement responder is able to deliver the specified service with the appropriate QoS, but not at the specified time, the agreement responder may accept the DoI offer with a changed start time (normally the next time after the requested the service could be provided). An agreement responder can query the DoI agreement and check, whether it fulfills his requirements or not. This enables an agreement initiator to negotiate a common timeslot for a resource usage as described in [3]. Since a DoI Agreement only declares the general intention to provide resources at a specified time, normally no costs and penalties will be associated with a DoI-Agreement. Therefore an agreement initiator may terminate a DoI agreement without any penalties.

Preparation Agreement

The main purpose of a Preparation Agreement (PA) is to provide a pre-reservation of resources (with defined QoS) for a specified time. This pre-reservation has a defined lifetime (pre-reservation duration), in which an agreement initiator can choose whether the specified resources will be used or not. If an agreement initiator chooses to use the pre-reserved resources, a subsequent Commit Agreement has to be created that references the PA using the wsag:ServiceReference. If no subsequent Commit Agreement was created within the pre-reservation duration, the PA expires and the reservation is deleted, in order to have the resources available again for other usage. After a PA has expired, the agreement initiator can not reference it anymore to create a subsequent Commit Agreement. Since resources will be reserved when establishing a PA (at least for a defined time), this agreement may or may not be associated with any costs or penalties, in case of canceling the PA. This depends on the local charging model, which should be published in the Agreement Template.

Commit Agreement
A Commit Agreement (CA) is a final outcome of a negotiation process. A CA references a PA in order to indicate that an agreement initiator definitely wants to use the resources reserved with the PA. This is the final contract between agreement initiator and agreement responder. On acceptance of a CA, all of the defined costs, penalties and reward defined in the agreement become effective.

Conclusion
[TODO]

[1] Skeen, D., "A Formal Model of Crash Recovery in a Distributed System," *IEEE Transactions on Software Engineering*, vol.9, no.3, May 1983, pp.219-228.
[2] Mark Mckeown, Dean Kuo, „Advance Reservation and Co-Allocation Protocol For Grid Computing“, In Proceedings of the First International Conference on e-Science and Grid Computing (e-Science'05)
[3] O. Waeldrich, Ph. Wieder, and W. Ziegler, A Meta-Scheduling Service for Co-allocating Arbitrary Types of Resources. In Proc. of the Second Grid Resource Management Workshop (GRMWS'05)
[4] VIOLA - Vertically Integrated Optical Testbed for Large Application in DFN. Web site, 22 November 2005, <http://www.viola-testbed.de/>.
[5] Andrieux, A., Czajkowski, K., and Dan, A. et al. 2004. Web Services Agreement Specification (WS-Agreement).