ARC Broker

Paper Subtitle

M.S. Andersen*

*skou@nbi.dk

Contents

1   Introduction

This document contains a description of the broker used in the Advanced Resource Connector (ARC) middleware as of version 1.0. In section 2 a description of how target attributes are matched and compared to attributes specified in the job description.

2   Attribute matching

In the broker attributes published by the candidate target need to be matched and compared to attributes specified in the job description to be able to select targets which are suitable for accepting the job description. Targets which do not pass a certain comparison will be not be a candidate target for submission. Detailed information about the attributes described below can be found in the ARC job description document. First the broker will check some basics for the target being processed. First, if the job description specifies submission to specific targets and/or queues, the target will be checked and if it is not among these targets and/or queues, it will be dropped. Next the number of free slots on the target will be checked, and if no free slots exist, then it will be dropped. After this, the health state of the target is checked, and if any other value than "ok" is published then it is dropped. Last, if middleware specific submission is specified in the job description, the middleware type of the target is checked, and if it does not satisfy the middleware requirements then it is dropped. After these basic checks, certain attributes specified in the job description are compared to the attributes published by the target being processed. If the target does not publish a attribute which is specified in the job description the target will dropped, with some exceptions.

References

[1] NorduGrid middleware, the Advanced Resource Connector. `http://www.nordugrid.org/middleware/`

[2] The Globus Alliance. `http://www.globus.org`

Job description attributes	Comparator	Target attribute	
`Resources.CandidateTarget[i].EndPointURL`	`==`	`url`	
`Resources.CandidateTarget[i].QueueName`	`==`	`MappingQueue`	
`Application.ProcessingStartTime`	$<$	`DowntimeStarts`	`AND`
`Application.ProcessingStartTime`	$>$	`DowntimeEnds`	
`Resources.CEType`	`isSatisfied(`	`Implementation`	`)`
`Resources.TotalWallTime.range.max`	$<$	`MaxWallTime`	
`Resources.TotalWallTime.range.min`	$>$	`MinWallTime`	
`Resources.TotalCPUTime.range.max`	$<$	`MaxCPUTime`	
`Resources.TotalCPUTime.range.min`	$>$	`MinCPUTime`	
`Resources.IndividualPhysicalMemory`	$\leq$	`MainMemory`	`OR`
`Resources.IndividualPhysicalMemory`	$\leq$	`MaxMemory`	
`Resources.IndividualVirtualMemory`	$\leq$	`MaxVirtualMemory`	
`Resources.Platform`	`==`	`Platform`	
`Resources.OperatingSystem`	`isSatisfied(`	`OperatingSystem`	`)`
`Resources.RunTimeEnvironment`	`isSatisfied(`	`ApplicationEnvironments`	`)`
`Resources.NetworkInfo`	`in`	`NetworkInfo`	
`Resources.DiskSpaceRequirement.SessionDiskSpace`	$\leq$	`MaxDiskSpace`	`OR`
`Resources.DiskSpaceRequirement.SessionDiskSpace`	$\leq$	`WorkingAreaTotal`	
`Resources.DiskSpaceRequirement.DiskSpace` `− Resources.DiskSpaceRequirement.CacheDiskSpace`	$\leq$	`MaxDiskSpace`	`OR`
`Resources.DiskSpaceRequirement.DiskSpace` `− Resources.DiskSpaceRequirement.CacheDiskSpace`	$\leq$	`WorkingAreaTotal`	
`Resources.DiskSpaceRequirement.DiskSpace`	$\leq$	`MaxDiskSpace`	`OR`
`Resources.DiskSpaceRequirement.DiskSpace`	$\leq$	`WorkingAreaTotal`	
`Resources.DiskSpaceRequirement.CacheDiskSpace`	$\leq$	`CacheTotal`	
`Resources.SlotRequirement.NumberOfSlots`	$\leq$	`TotalSlots`	
`Resources.SlotRequirement.NumberOfSlots`	$\leq$	`MaxSlotsPerJob`	
`Resources.SessionLifeTime`	$\leq$	`WorkingAreaLifeTime`	
`Resources.NodeAccess is NAT_INBOUND OR NAT_INOUTBOUND`	`AND`	`ConnectivityIn`	
`Resources.NodeAccess is NAT_OUTBOUND OR NAT_INOUTBOUND`	`AND`	`ConnectivityOut`	

Table 1: