D1.4: Definition of the architecture, technical plan and evaluation criteria for the resource co-allocation framework and mechanisms for parallel partitionning

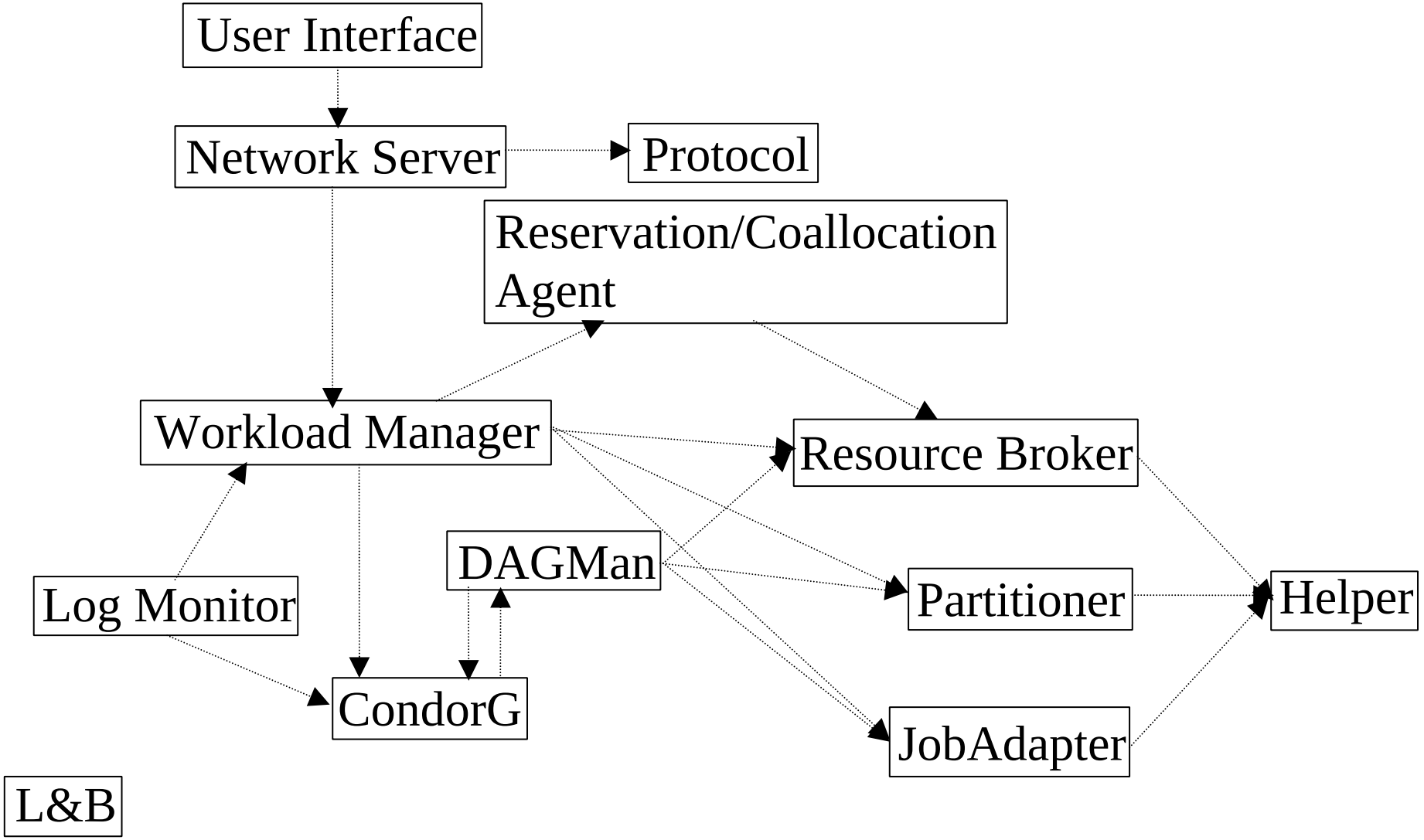
Corresponding Author:	M. Sgaravatto, WP1
Reviewers:	J. Linford, WP9
	D. Bosio, WP2
	M. Malawski, CrossGrid
Moderator:	J. Montagnat, WP10

Document content

Extensions to the Workload Management System.

0. Review of the WMS architecture (section 3)
1. Advance reservation (section 4)
2. Resources co-allocation (section 5)
3. Job partitionning (section 6)
4. Job dependencies (section 7)

0. Architecture overview



1. Advance reservation

Reservation through a JDL extension:

[Type = “ Reservation”;

ReservationType = computing;

ReservationStart = 1021539656;

ReservationEnd = 1021541000;

ReservationDuration = 300;

ReservationParameters = [nodes = 3;]

Requirements = other.SupportReservation;]

Reservation identifier: query, cancel, modify

2 Steps process:

- Make reservation
- Use reservation

2. Resources co-allocation

Co-allocation through a JDL extension:

```
[ Type = “  Coallocation”;
ReservationType = computing;
ReservationStart = 1022248228;
ReservationEnd = 1022255428;
ReservationDuration = 3600;
Res1 = [ Type=”Reservation”;
          ResourceType=”computation”;
         ... ];
Res2 = [ Type=”Reservation”;
          ResourceType=”storage”;
         ... ]; ]
```

Coallocation identifier: query, cancel, modify

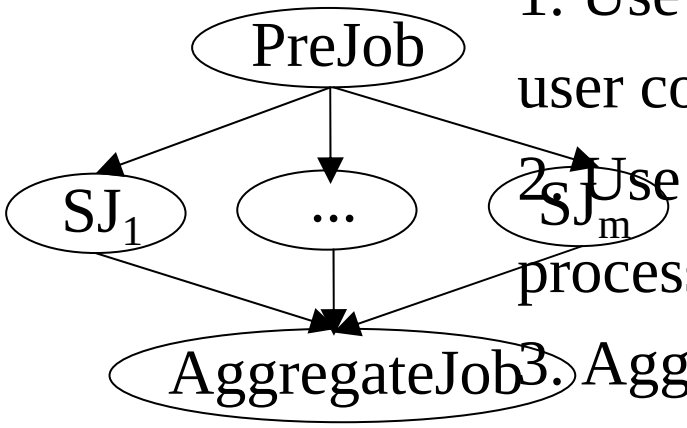
2 Steps process: Allocate, the user the co-allocation

3. Job partitionning

Job checkpointing

- Save <var, value> pairs (store in the L&B)
- Restore state to restart processing at any earlier saved state
- Save large data amount in files and checkpoints LFNs

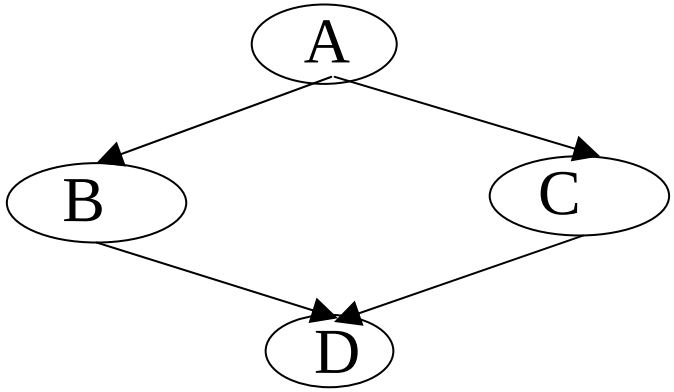
Job partitioning



1. Use checkpointing to instrument user code and save SJi final state.
2. Use the Condor DAG manager to process all SJi in parallel.
3. AggregateJob restore all final states and build the final result.

4. Job dependencies

Represent job dependencies as Directed Acyclic Graph (DAG)



Represent the DAG in a JDL file

Use Condor DAGMan to process the DAG

Discussion

∴ Level of details

∴ English improvements

∴ Merging advance reservation and resources coallocation

∴ Job checkpointing for partitionning