Condor DAGMan

An Introduction

Condor Project
Computer Sciences Department
University of Wisconsin-Madison
condor-admin@cs.wisc.edu
http://www.cs.wisc.edu/condor

Want other Scheduling possibilities? Use the Scheduler Universe

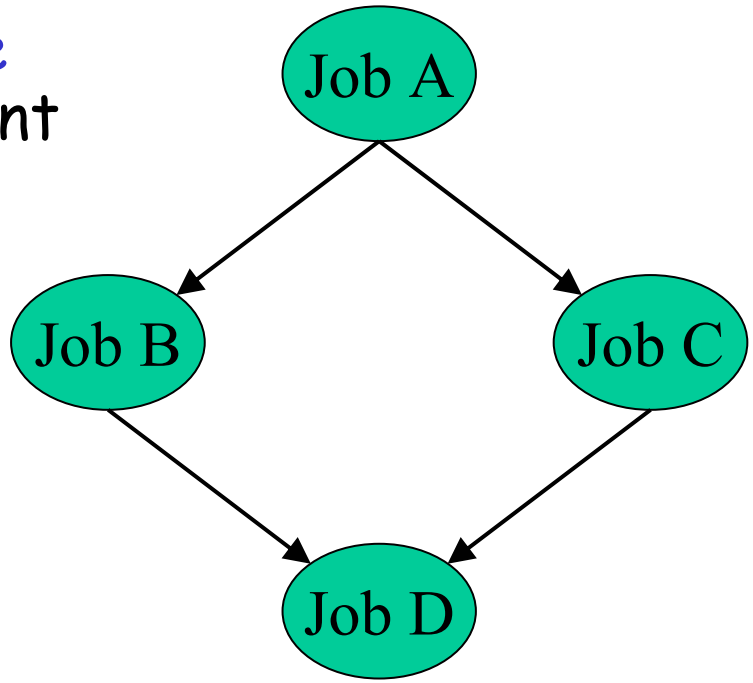
- In addition to Globus, another job universe is the *Scheduler Universe*.
- Scheduler Universe jobs run on the submitting machine.
- Can serve as a meta-scheduler.
- DAGMan meta-scheduler included

DAGMan

- **Directed Acyclic Graph Manager**
- DAGMan allows you to specify the *dependencies* between your Condor-G jobs, so it can *manage* them automatically for you.
- (e.g., "Don't run job "B" until job "A" has completed successfully.")

What is a DAG?

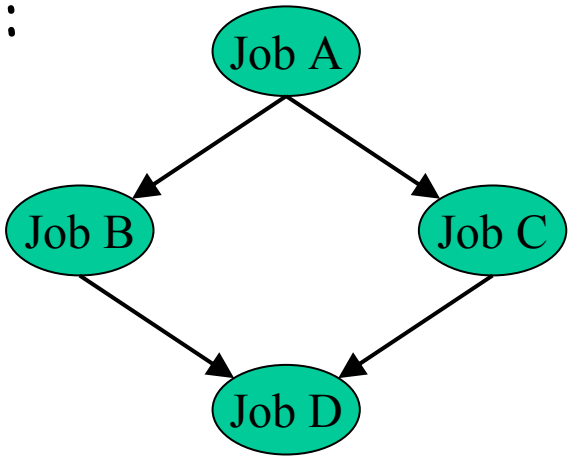
- A DAG is the data structure used by DAGMan to represent these dependencies.
- Each job is a "node" in the DAG.
- Each node can have any number of "parent" or "children" nodes – as long as there are no loops!



Defining a DAG

- A DAG is defined by a *.dag file*, listing each of its nodes and their dependencies:

```
# diamond.dag
Job A a.sub
Job B b.sub
Job C c.sub
Job D d.sub
Parent A Child B C
Parent B C Child D
```



- each node will run the Condor-G job specified by its accompanying Condor submit file

Submitting a DAG

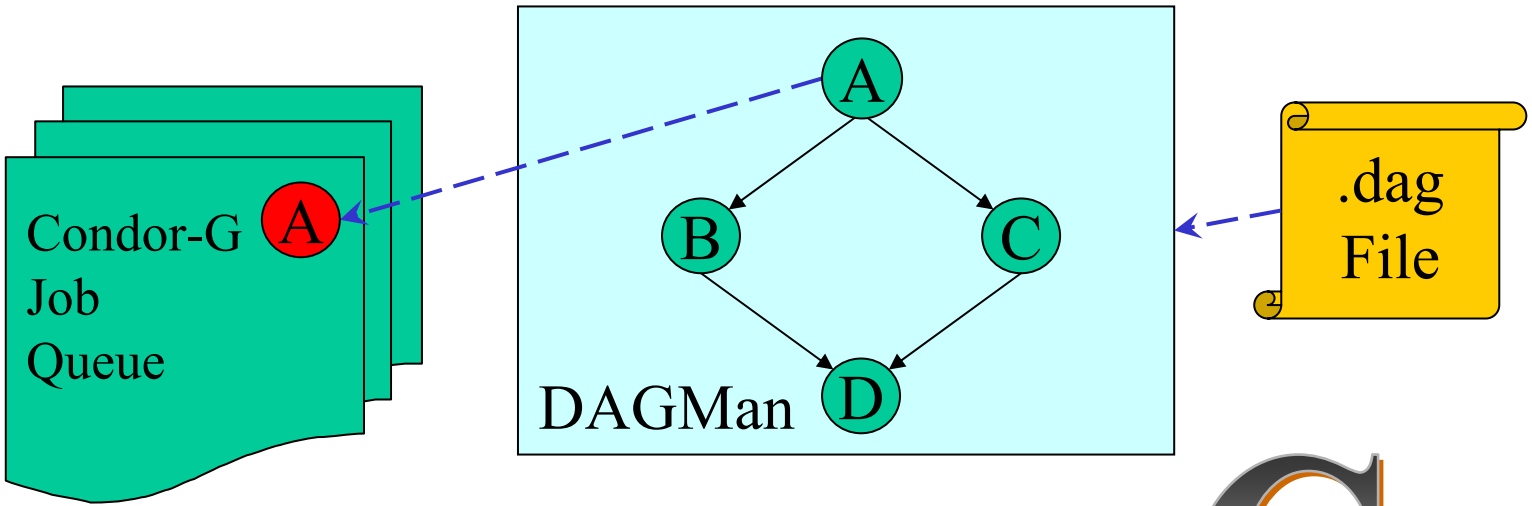
- To start your DAG, just run ***condor_submit_dag*** with your .dag file, and Condor will start a personal DAGMan daemon which to begin running your jobs:

```
% condor_submit_dag diamond.dag
```

- condor_submit_dag submits a Scheduler Universe Job with DAGMan as the executable.
- Thus the DAGMan daemon itself runs as a Condor-G scheduler universe job, so you don't have to baby-sit it.

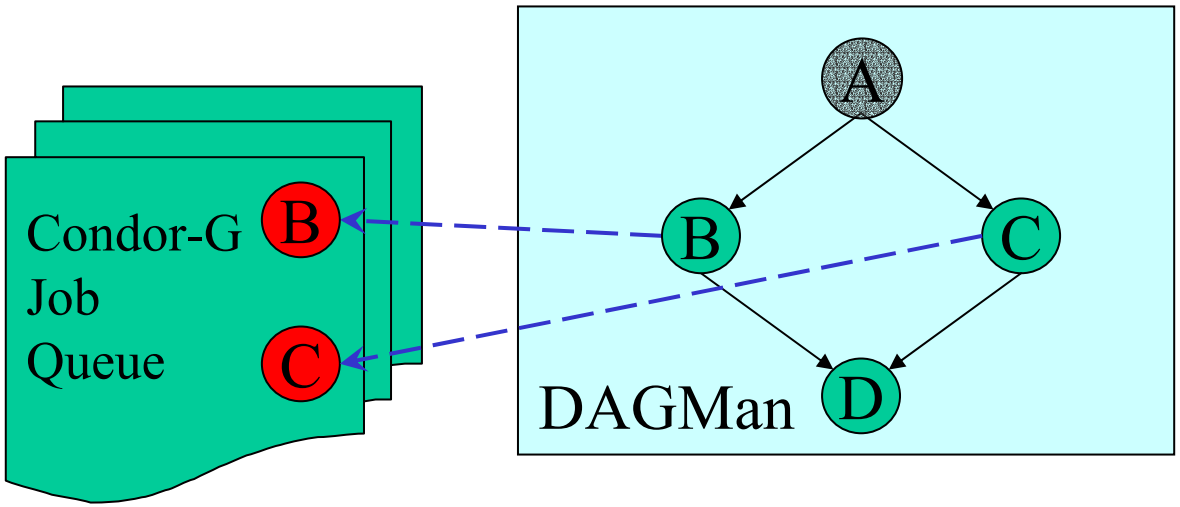
Running a DAG

- DAGMan acts as a "meta-scheduler", managing the submission of your jobs to Condor-G based on the DAG dependencies.



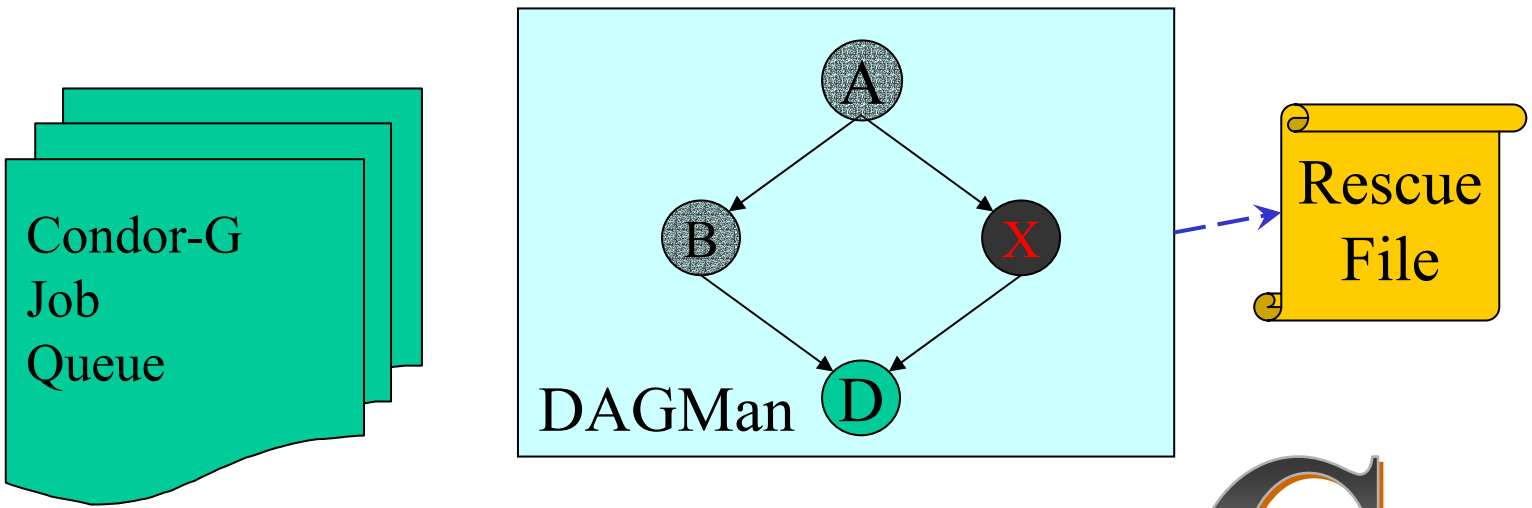
Running a DAG (cont'd)

- DAGMan holds & submits jobs to the Condor-G queue at the appropriate times.



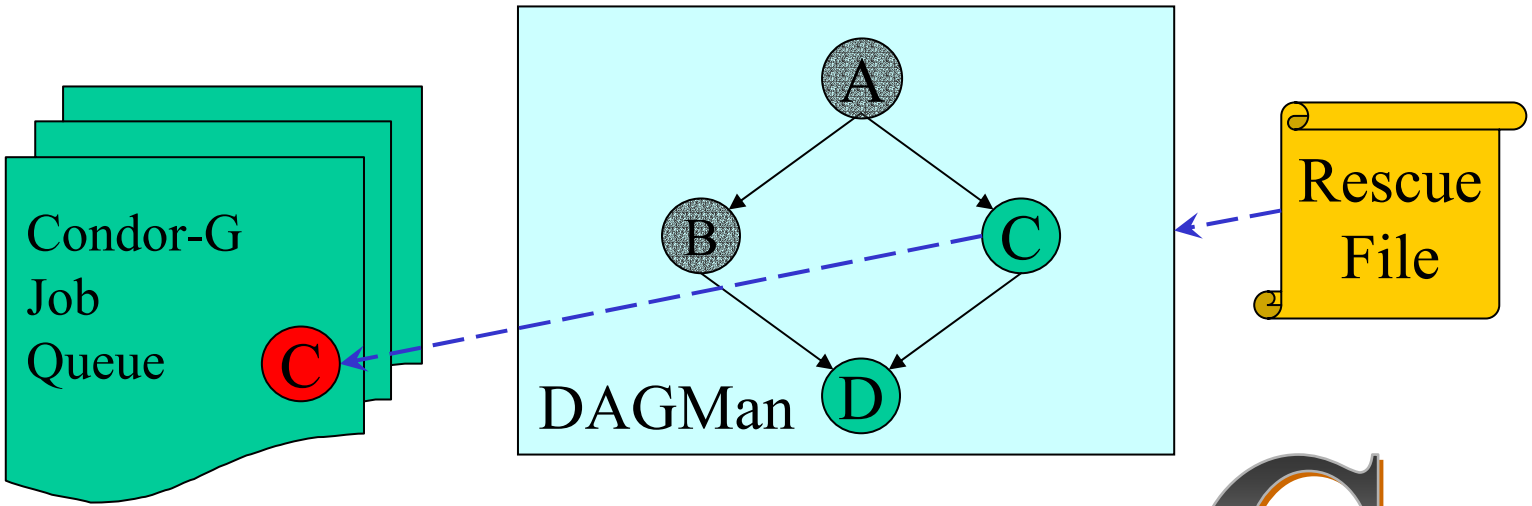
Running a DAG (cont'd)

- In case of a job failure, DAGMan continues until it can no longer make progress, and then creates a *"rescue" file* with the current state of the DAG.



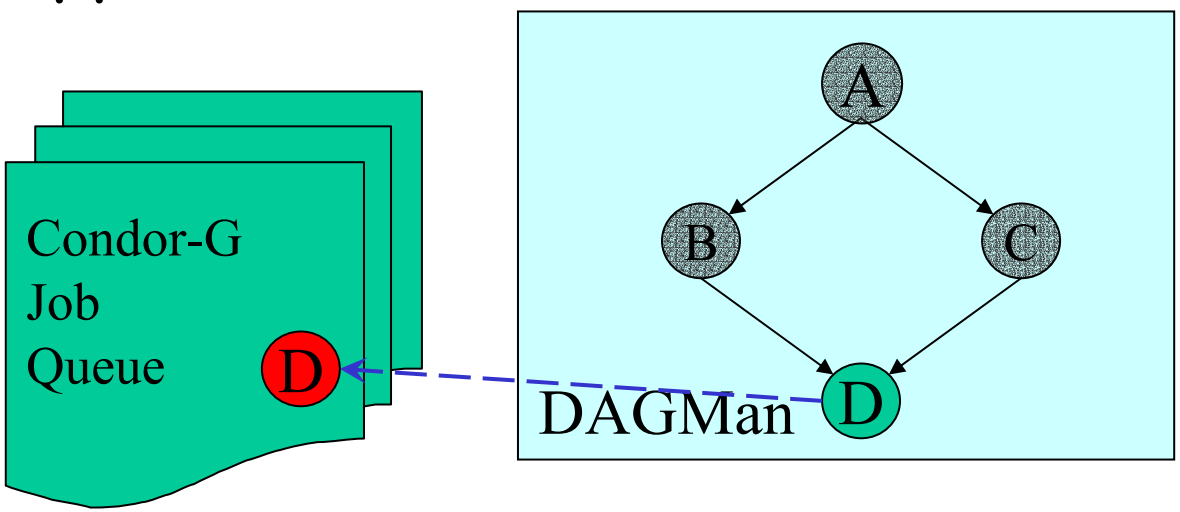
Recovering a DAG

- Once the failed job is ready to be re-run, the rescue file can be used to restore the prior state of the DAG.



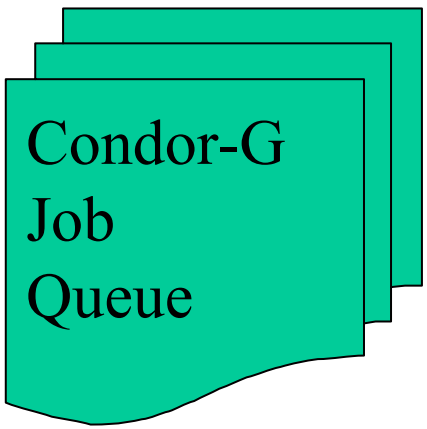
Recovering a DAG (cont'd)

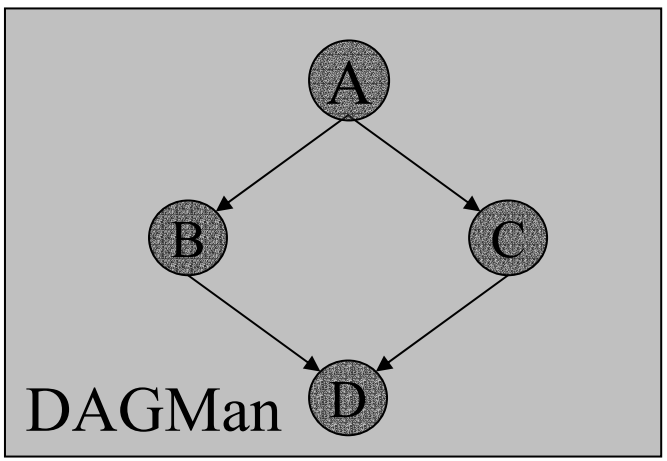
- Once that job completes, DAGMan will continue the DAG as if the failure never happened.



Finishing a DAG

- Once the DAG is complete, the DAGMan job itself is finished, and exits.





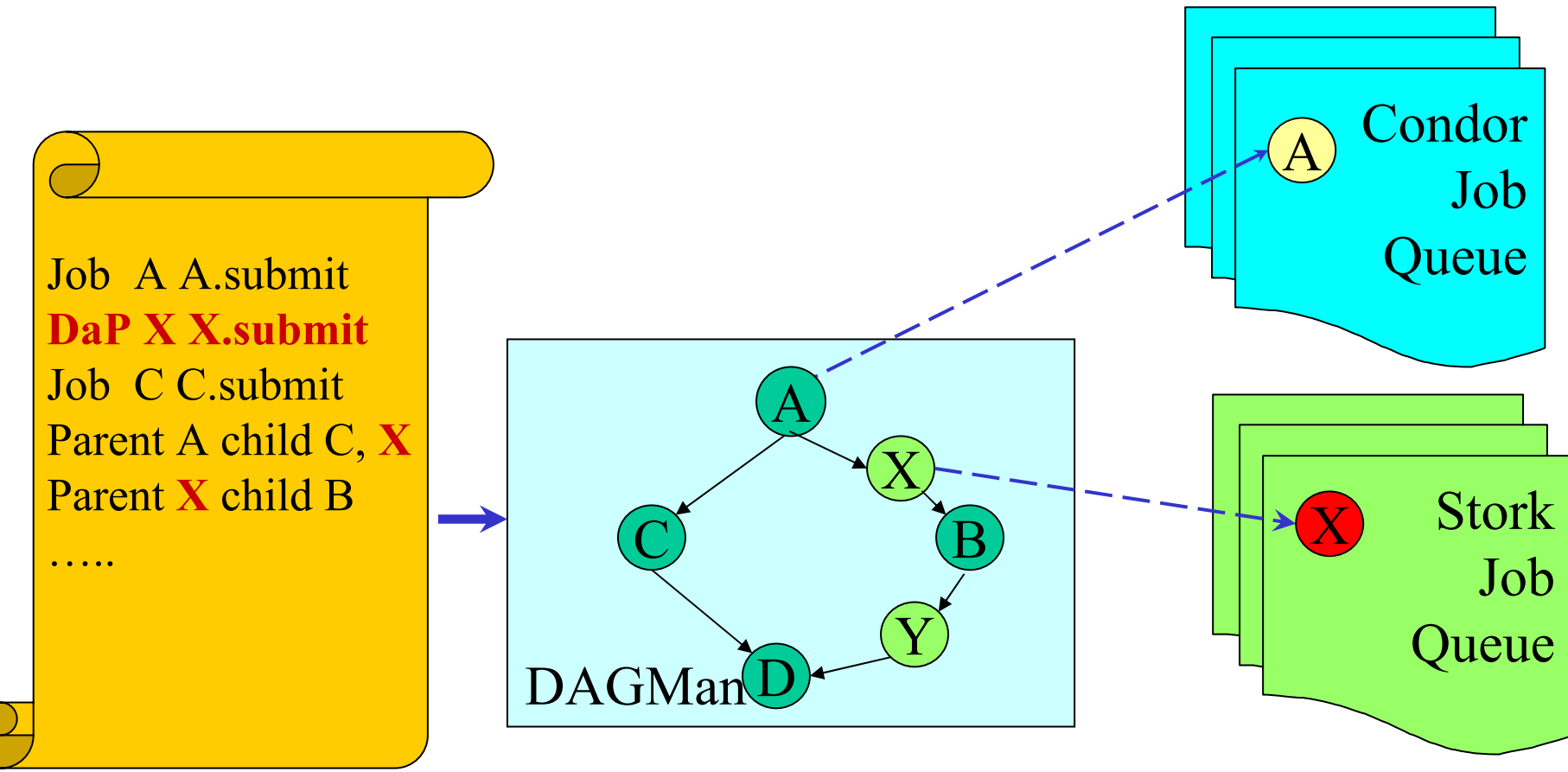
Additional DAGMan Features

- Provides other handy features for job management…
  - nodes can have PRE & POST scripts
  - failed nodes can be automatically re-tried a configurable number of times

Data Placement* (DaP) must be an integral part of the end-to-end solution

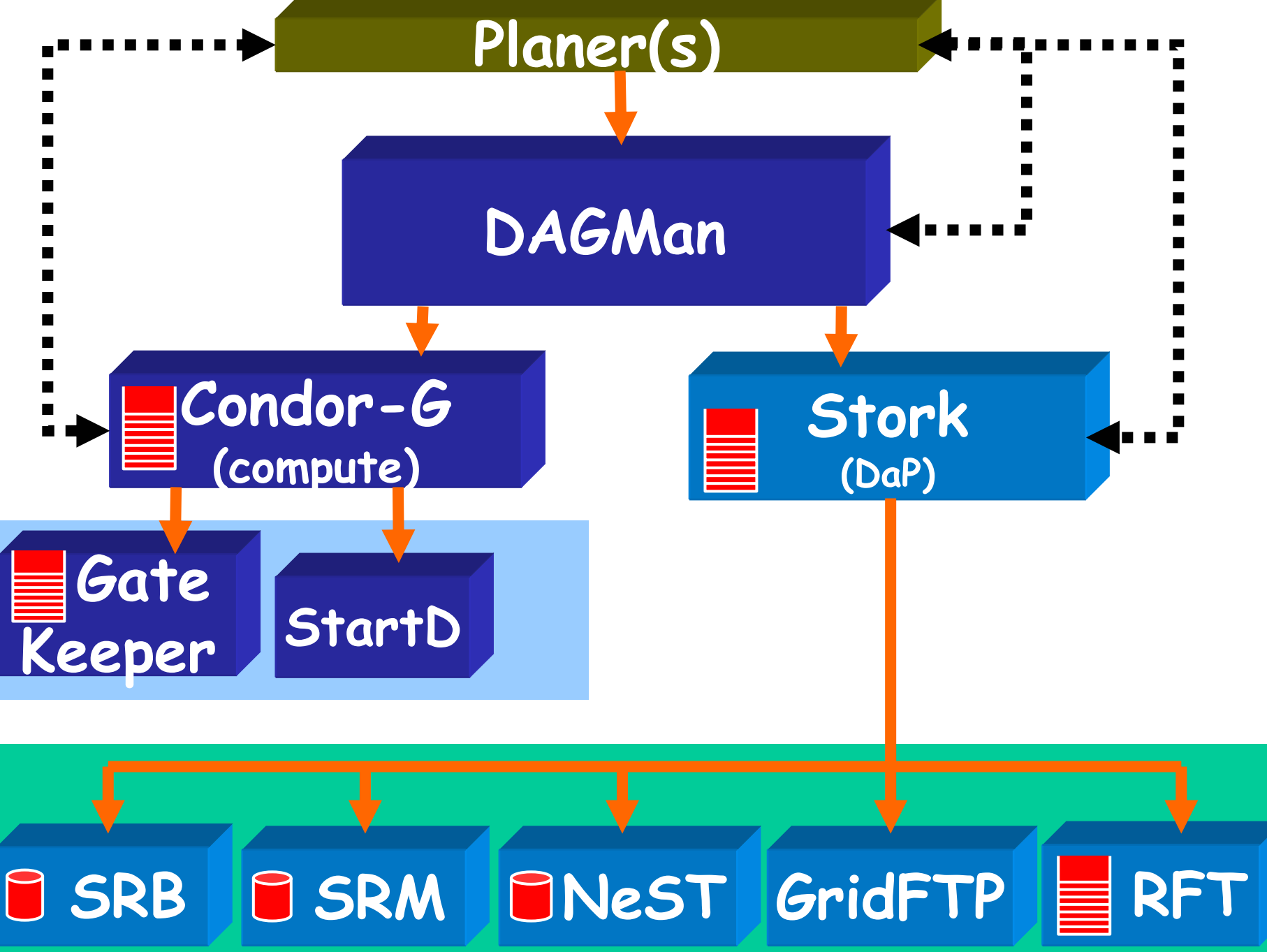
* Space management and Data transfer

Interaction with DAGMan



Stork

- Schedules, runs, monitors, and manages Data Placement (DaP) jobs in a heterogeneous Grid environment & ensures that they complete.
- **What Condor (G) means for computational jobs, Stork means the same for DaP jobs.**
- Just submit a bunch of DaP jobs and then relax..


Planer(s)
DAGMan
Condor-G
(compute)
Stork
(DaP)
Gate
Keeper
StartD
SRB
SRM
NeST
GridFTP
RFT

DAGMan Callouts

- Another mechanism to achieve lazy planning: DAGMan callouts
- Define DAGMAN_HELPER_COMMAND in condor_config (usually a script)
- The helper command is passed a copy of the job submit file when DAGMan is about to submit that node in the graph
- This allows changes to be made to the submit file (such as changing GlobusScheduler) at the last minute

Some Recent or soon to arrive Condor-G / DAGMan features

- Condor-G can submit and manage jobs not only in Condor and Globus managed grids, but also to
  - Nordugrid (http://www.nordugrid.org/)
  - Oracle Database (using Oracle Call Interface [OCI] API)
  - UNICORE
- Dynamic DAGs

Some recent or soon to arrive Condor-G / DAGMan features, cont.

- MyProxy integration w/ Condor-G
  - Condor-G can renew grid credentials unattended
- Multi-Tier job submission
  - Allows jobs to be submitted from a machine which need not be always connected to the network (e.g. a laptop)
  - condor_submit sends job Classad and job "sandbox" to a remote condor_schedd
  - condor_fetch_sandbox used to retrieve output from remote condor_schedd when job completes
- SOAP Interface
- Job submission to Globus Toolkit 3 managed job service

Thank you!

Check us out on the Web:

http://www.cs.wisc.edu/condor

Email:

condor-admin@cs.wisc.edu