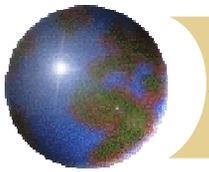


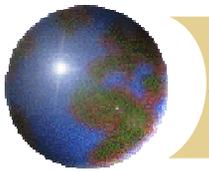
SAMGrid for CDF MC (and beyond)

Igor Terekhov, FNAL/CD/CCF/SAM
for JIM team



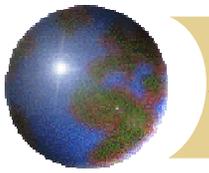
Plan of Attack

- General (but technical!) intro into Grid computing
- Overview of some of the benefits of SAMGrid computing, for CDF MC etc.
- Architectural perspective
 - SAMGrid as a whole
 - SAM data handling
 - JIM job submission
- A more practical, detailed description of CDF MC process with JIM/SAMGrid
- JIM project status



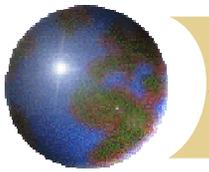
Global and Grid Computing in HEP: the Evolution

- I. Globally distributed computing
- II. Automated, Grid-like Globally Distributed Computing ← SAMGrid
- III. True Grid computing



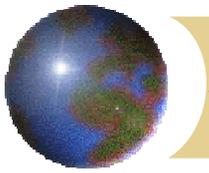
Globally Distributed Computing

- Multiple participating sites (especially MC)
- Experts on sites
- Centrally provided KITS and other s/w repositories
- Locally developed/modified infrastructure for production tracking, workflow and job management, etc
- E-mail and phone communications (what to install, how to patch, who's doing what)



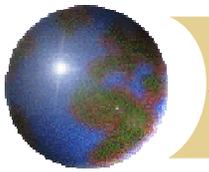
Grid-like GDC: SAMGrid

- Sites have “standard” infrastructure – SAM stations and other SAMGrid servers, but no pre-installed D0/CDF software or data
- ⊖ All data files are delivered from the SAM data grid:
 - ▣ D0 example: minbias mix-in files used to be all different
 - ▣ JIM uses a SAM dataset thus guaranteeing consistency
- ⊖ All job files are delivered from the SAM data grid:
 - ▣ Release files are globally distributed and cached, no need for explicit software synchronization
- Remote job submission, with placement directed by the system or user:
brokering
 - ▣ in (D)CAF, peer->peer submission
 - ▣ In JIM, client->system->execution site
- ⊖ Spooling of **small** input and output files
 - ▣ For fun: web-based retrieval of output
- ⊖ Expertise on sites <1 person, **almost never** beyond the initial SAM install phase
- Monitoring of the overall state of the system



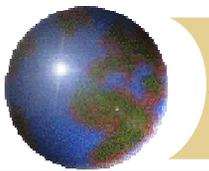
La Grille Pure

- ⊕ Computer Science origins
- ⊕ Common “middleware” infrastructure
- ⊕ True distributed ownership of resources
 - ⊞ Run MC on a biologist’s cluster
- ⊕ No preinstalled software except “standard” tools like Globus
- ⊕ A bit of utopia?

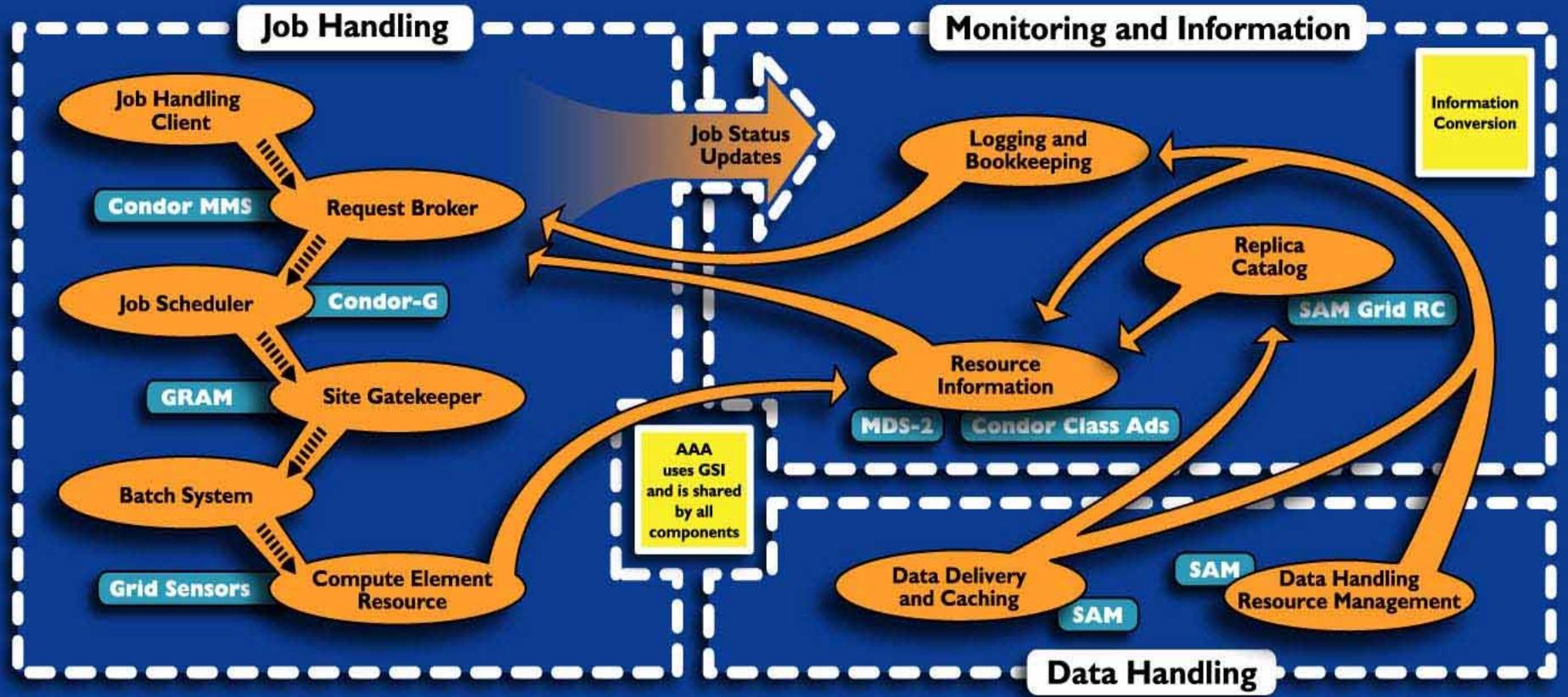


We're at midpoint -- SAMGrid

- ⊕ Principal benefits for you, CDF MC physicists:
 - ⊠ Higher degree of automation makes MC easier and “more fun”
 - ⊠ Considerably higher degree of consistency and independence of physics from site (job/data files, request details in DB)
 - ⊠ Better utilization of resources (eventually)
 - ⊠ Reduction of expertise at sites from $O(N)$ -> $O(1)$
 - ⊠ Core SAMGrid software (SAM+JIM) – common across D0 and CDF (but not necessarily with LHC) – CD support etc
- ⊕ Possible future of SAMGrid and Run II computing
 - ⊠ I'm not authorized to predict it
 - ⊠ Full integration into *The Grid* (la grille pure) unlikely IMHO; will probably continue to run on resources at least partially affiliated with Run II experiments
 - ⊠ Gradual convergence with LHC technologies, while providing stable services to Run II physicists
 - ⊠ And/or integration into US grid efforts (Openscience Grid)



SAM-Grid Architecture



Principal Component

Service

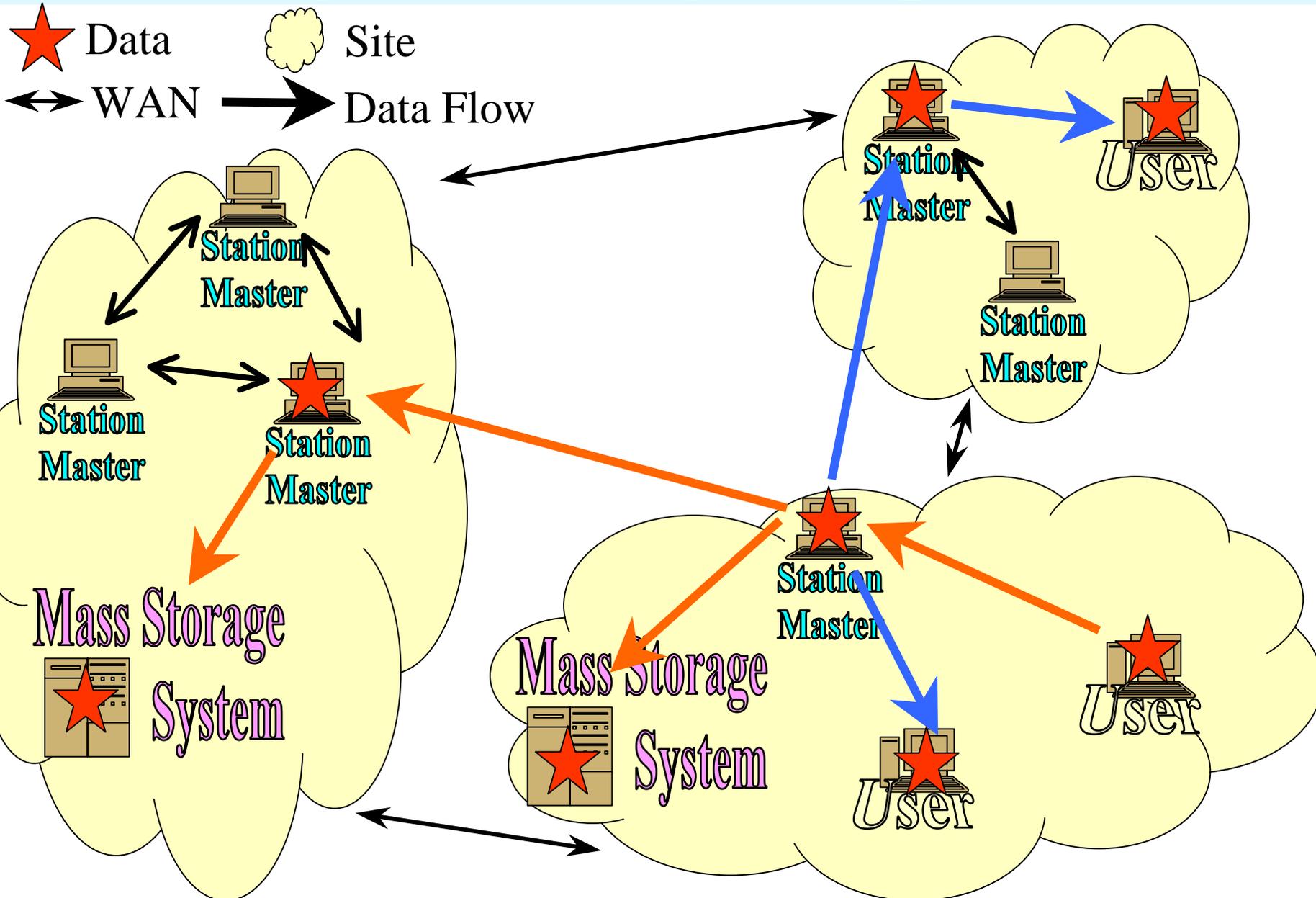
Control

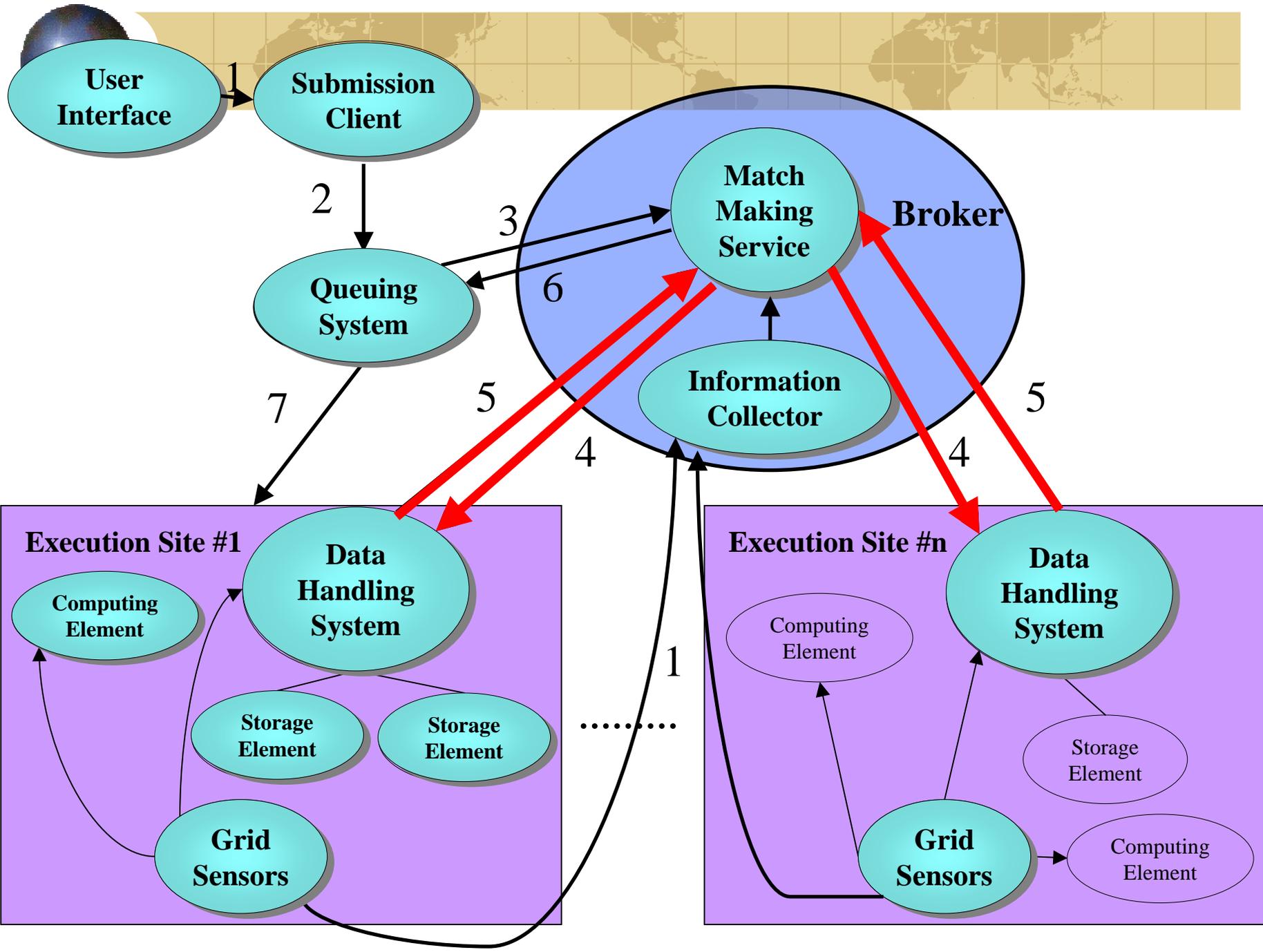
Implementation or Library

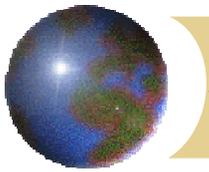
Shared Resource

Information

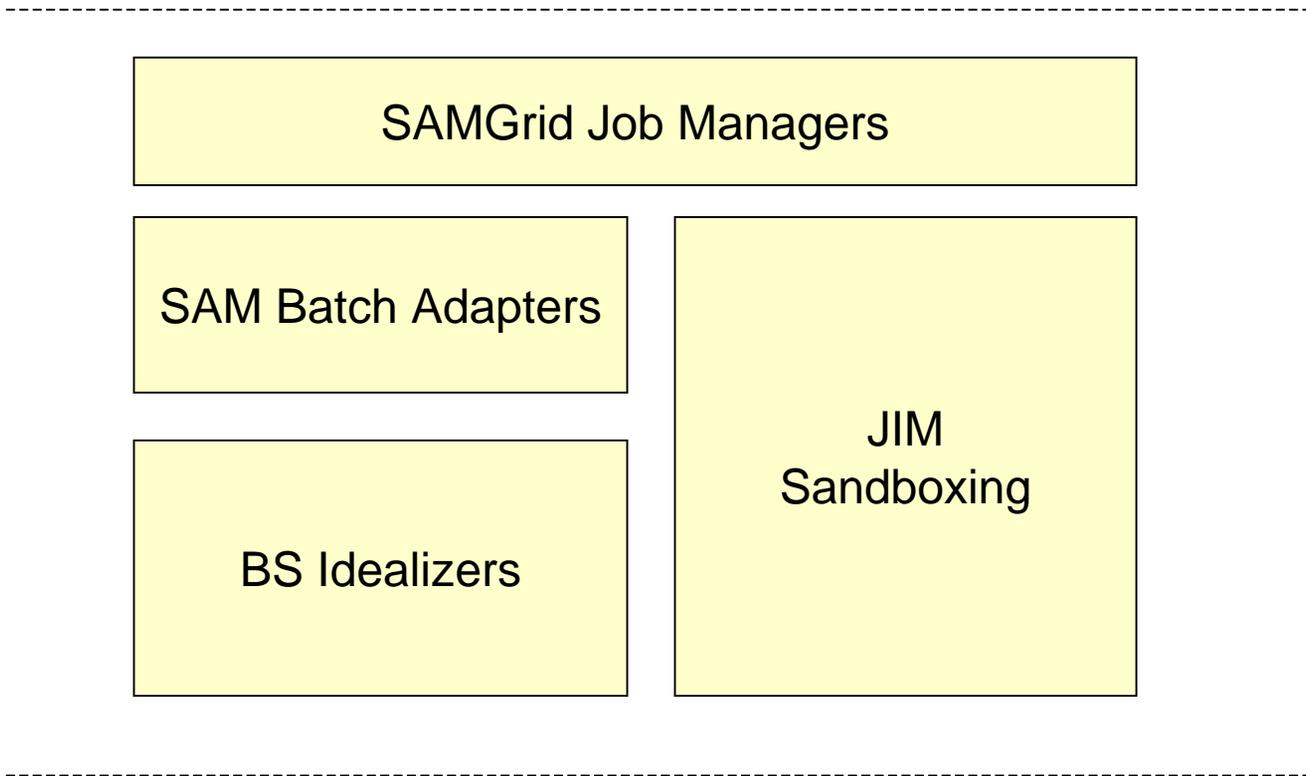
Routing + Caching = Replication



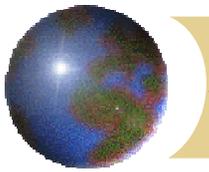




Grid to Fabric The Grid *Job Submission*

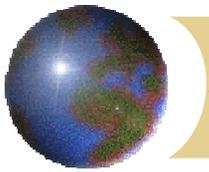


Local Batch System



Enough of General Stuff

- Install and configure SAMGrid software at participating sites
 - SAM station
 - JIM++ software. Very good document, <http://www-d0.fnal.gov/computing/grid/SAMGridManual.htm>
- Prepare an input sandbox!!!
- Create a request in the SAM DB!
- Write a small job description file (JDF)
- Do "samg submit"
- *Et voila*, see <http://samgrid.fnal.gov:8080> etc.



Sample CDF MC JDF

```
job_type = cdfmc
```

```
# Experiment and universe
```

```
sam_experiment = cdf
```

```
sam_universe = prd
```

```
# SAM group and station
```

```
group = test
```

```
station_name = samgfarm
```

```
# CDF job details
```

```
requestid = 34
```

```
numevts = 1000
```

```
events_per_job = 500
```

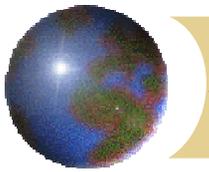
```
job_specification = cdf_mc_jobspec.xml
```

```
input_sandbox_tgz = /tmp/cdfuser.tar.gz
```

```
# Jobfile dataset
```

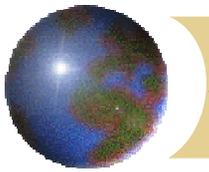
```
jobfiles_dataset = jobset_igor_2
```

```
instances = 1
```



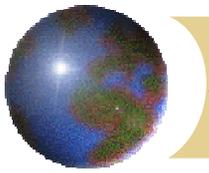
Present CDF features

- ⊕ Takes a “job dataset” and delivers to worker node
- ⊕ Takes job specification files, an XML map: run number -> number of events (if you prefer, a list of run/numevents pairs)
- ⊕ Accepts user .tar.gz (will transfer to the worker node)
- ⊕ Having routed the job to an execution site, will compute the detailed plan
 - ▣ Each local job is assigned 1 or more (run, event range) pairs
 - ▣ Total number of local jobs is a function of *both* the job specification (total number of events) and the site’s capabilities (e.g. “optimal” CPU per local job).
- ⊕ User-supplied “run1run” script is invoked for each run’s event range
- ⊕ All output data files stored back to SAM
- ⊕ Output non-data files (stdout, logs, etc) are viewable on the Web.
- ⊕ Output data files can be merged later (see next slide)



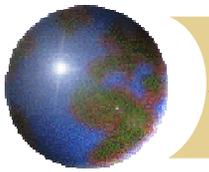
Output merging(concatenation)

- One of JIM/SAMGrid benefits
- ⊖ The problem is caused by the existing Storage Systems being unable to swallow large number of small files
- Important for both D0 and CDF.
- Our plan has been implemented for D0 (CDF to come):
 - Put output data files to “durable storage”, sam store –dest=XXX
 - Define a SAMGrid job that looks like a SAM analysis job, taking a SAM dataset as input
 - Submit it to any execution site (possibly site of original production will be preferred)
 - Merged output is automagically stored back to SAM
- Principal benefits:
 - Can merge files produced at *very* different times/places
 - Bookkeeping, robustness features of SAM are leveraged
- Difficulties:
 - Bookkeeping backfires (mix of merged/unmerged files)
 - All at once approach overfills scratch space, need real streaming (as in true SAM)
- ⊖ Core SAM is enhanced accordingly to overcome issues/improve service



Near (and not) Future for CDF MC

- ⊕ Decouple MC production phases:
 - ⊕ Be able, for example, to retrieve generated files that were previously produced
 - ⊕ Read that input from SAM
 - ⊕ Has been in D0 JIM for quite a while already
- ⊕ Improve concatenation (first implement it for CDF)
- ⊕ Fuller MC request system, integration with CDF JIM
- ⊕ Incorporate any new requirements from **you**, the users
- ⊕ Perhaps “workflow manager” (application manager) such as D0/CMS mc_runjob
- ⊕ Perhaps full-fledged brokering (employ multiple sites for a single large request)
- ⊕ Continuous monitoring improvements
- ⊕ Understand relation with CAF



Manpower resources

- ⊕ Unfortunately, I am moving out of SAMGrid
- ⊕ The remaining person (Gabriele Garzoglio), and two JIM students *will* have to be split between D0 and CDF
- ⊕ Expertise **must** grow within the experiment to:
 - ⊕ Setup new sites
 - ⊕ Understand the JIM software and tweak the “job managers” etc accordingly
- ⊕ Morag Burgon-Lyon and Valeria Bartsch are ramping up. Ulrich Kerzel is expanding expertise SAM->SAMGrid
- ⊕ CD/Run II department/SAMGrid project (co-led by Rick St Denis and Wyatt Merritt) **will** cough up other resources
- ⊕ But once again, this will die if the experiment doesn't pick up!