

Reducing REX Department GDM Effort

GDM

May 30, 2006

Current GDM-Related Effort

Rough estimates!

	<u>CD</u>	<u>Experiment</u>
Minos DH operations	0.5 FTE	0 (?)
D0 DH operations	0.75	0.5
System administration	1.5	0
CDF DH operations	2.5	1
SAMGrid	3.75	0

(It's not clear how much to count for each experiment's SAM shifters)

How to decrease effort?

MINOS: ?

D0 DH operations: ?

System administration:

- Contract out hardware monitoring
- Purchase more reliable (and expensive) disk storage

CDF data handling operations

Major current activities:

1. Responding to user, system problems 0.7 FTE
2. Upgrading and testing SAM db servers, stations \
3. DFC -> SAM migration \ 0.8
4. Recycling tapes /
5. Migrating to new hardware, OS on pool nodes /

How to reduce CDF DH operations effort?

Options:

- Complete the DFC -> SAM migration
- More tapes and robot space, for flexibility when migrating tapes
- Freeze current system configuration: hardware, OS, etc.
- Work toward point where fewer SAM code releases are necessary: fewer changes in requirements, fewer bugs (more on this later)
- More robust system: less vulnerable to network or database availability

CDF DH operations effort

But these options are difficult to achieve:

- Configuration must change to incorporate new hardware, new requirements, new design
- Even when requirements do not change, still need code changes to fix bugs, use new software versions (grid, Python, Oracle)
- Tapes and robots are expensive

CDF DH operations effort

And some tendencies toward **more** effort:

- Higher data rate will increase load
- Specter of decreasing experiment contribution

CDF DH operations: What is being done now?

- DFC -> SAM migration is nearing completion (this will **increase** load initially as more people use SAM, but decrease in the long run as DFC no longer needs support)
- Work on improved SAM code testing, monitoring will also help here; more on this below

CDF DH operations: Possible savings

Complete DFC -> SAM migration	0.1 FTE
More reliable disk hardware, more stable system configuration	0.25

DØ Operational Load (From Gavin)

- ~ 0.75 FTE
- Already reducing load by...
 - Migrating to SAM v7 (though this will raise the load short term)
 - Data handling / SAMGrid deployment post
 - Joint FNAL / collaboration effort
- For Analysis:
(Practice so far driven by experience and model driven by rate to tape)
 - Improved use of cache space and disk management
 - Increased metrics (MIS will help here!)
 - Improved use of Cab
 - Improved file delivery, user quotas, planning
 - Improve documentation

SAMGrid effort

1. Operations (bug fixes, responding to stakeholder problems/requests)	1-2 FTE
2. Core development	2-3
3. Deployment	1
4. Project management	0.5

How to reduce SAMGrid effort?

1. Fewer code releases:

- Improve pre-release code testing to reduce bug-fix releases
 - ❖ Already test dbserver; adding integrated station testing, then client testing.
- Better communication with stakeholders to understand present & future use cases (and better anticipation by stakeholders)

But:

- Use cases change over time
- Data rates, and user demand, increase
- Other software (grid, Oracle, Python) changes

Reduce SAMGrid effort

2. Decrease support load:

- Improve pre-release code testing, so fewer bugs
- Get better handle on numerous versions and configurations at different sites

But:

- Will never catch **all** bugs before release
- May not be practical to reduce # configurations

Reduce SAMGrid effort

3. Better monitoring to catch problems earlier and understand the system better

- MIS:

- ❖ SAM applications and services are instrumented to send monitoring events to a server for action and collection
- ❖ Server stores events in a database for static displays (e.g. samHDTV)
- ❖ Server backends can watch for certain events to automate alerts
- ❖ It will vastly improve our monitoring capabilities.
- ❖ It is nearing deployment!

- May increase load in the short term, as new issues are found; but should decrease in the long term.

SAMGrid: What is being done now?

1. Pre-release testing: Upgrading test harness, writing integration tests
2. Improved monitoring: MIS
3. Use case management: Monthly stakeholders meeting; improve planning at both ends
4. Improved documentation (reduce user errors, reduce SAM shifters having to ask experts): Low activity
5. Site-to-site variation: Hard to reduce this; working on better communication so it's at least understood better

SAMGrid: Possible savings

Improved pre-release testing, monitoring--fewer bugs	0.25 FTE
Improved planning, coordination with stakeholders	0.25-0.5

Conclusion: Savings for REX GDM effort

System administration	0.25 FTE?
CDF DH operation	0.5
SAMGrid	0.5

We are always looking for efficiencies to improve what we can do. This exercise has not found ways to reduce effort by 50%; to do that will be an even more difficult exercise.