

PPDG Internal Review

Project Activity:

Attendees:

Reviewers:

Date:

Documents/URLs to be Read by the Reviewers:

CMS Tutorial : Grid Session (G. Graham, chair)

HYPERLINK "http://heppc16.ucsd.edu/cms_tutorial/agenda.html"

http://heppc16.ucsd.edu/cms_tutorial/agenda.html

CMS Grid Tools for Production Overview (ppt, G. Graham,)

Use of Globus and Virtual Data Toolkit (ppt, R. Cavanaugh)

Use of MOP: Grid Tools in Production (ppt, P. Couvares)

MOP: A System for CMS Monte Carlo distributed Production (J. Amundson, Greg Graham) <http://www.ppdg.net/pa/ppdg-pa/mop/mop.pdf>

MOP Updated Architecture and Status Document (TBA, Peter Couvares, Rick Cavanaugh, and Greg Graham)

List of Significant Bugs Tracked Down Using MOP (TBA, Rick Cavanaugh and Peter Couvares) and Appendix A

Status of the Project:

The project is overall in good shape.

From the Grid perspective, MOP has been outstanding. It has both legitimized the idea of using Grid tools such as DAGMAN, Condor-G, GDMP, and Globus in a real production environment outside of prototypes and trade show demonstrations.

Furthermore, it has motivated the use of Grid tools such as DAGMAN, Condor-G, GDMP, and Globus in novel environments leading to the discovery of many bugs which would otherwise have prevented these tools from being taken seriously in a real production environment.

From the CMS perspective, MOP won early respect for taking on real production problems, and is soon ready to deliver real events. In fact, today or early next week we will update the RefDB at CERN which tracks production at various regional centers. This has been delayed because of the numerous bugs that, while being tracked down, involved several cycles of development and redeployment. The end of the current CMS production cycle is in three weeks, and MOP will be able to demonstrate some grid enabled production capability by then. We are confident that this will happen. It is not necessary at this stage to have a perfect MOP system for CMS Production; IMPALA also has some failover capability and we will use that where possible. However, it has been a very useful exercise and we believe that we are among the first team to tackle Globus and Condor-G in such a stringent and HEP specific environment.

See Appendix A for a list of some of the bugs found in production.

Plans for the Project:

For the next 3 months:

- Commission MOP-slave sites at Caltech, Wisconsin, and FNAL.
- Commission a production MOP-master site at FNAL while continuing to use the development MOP-master at Wisconsin.
- Use production MOP-master in Summer production, which will use a "BigMinBias" style of running.

For PPDG Year 2:

- Evaluate the current MOP paradigm for using Grid tools and re-engineer or re-factor into other Grid tools.
- Continue to put stress onto the underlying Grid components to make sure they are production ready.

Questions for the Project (written responses submitted before the review are appreciated):

What are the deliverables of your project activity, † how † has the project met the deliverables to date, what effort has been contributing to the project?

MOP Integration with CMS IMPALA production tools. (Done, March 2002)

Deployment of Grid Tools using VDT in preparation for MOP running. (UCSD, UFL Done March 2002; Caltech, FNAL not done yet.)

200K simulated min-bias events for official CMS Production (not done yet, but very close within a week)

What is the deployment plan for your project activity and † what is the state of that deployment?

Project is deployed as follows:

MOP slave site UCSD, March 2002

MOP slave site UFL, March 2002

devel MOP master WISC, March 2002

MOP slave site Caltech, May 2002

MOP slave site FNAL, May 2002

MOP slave site at Wisconsin, May/June 2002

production MOP master FNAL, May/June 2002

-Needs to be done for Summer Production

Has the project benefited from being part of the PPDG work and if so how?

The input of Peter Couvares and Alain Roy and Shahzad Muzaffar has been invaluable to the project and it would not have gone forward without them.

Close cooperation between GDMP developers MOP developers has been invaluable.

Close cooperation between Condor developers and Globus experts has been invaluable.

Has the project been hindered by being part of the PPDG work and if so how?

No.

What collaborations does your project activity rely on † and/or contribute to? Have these

been of benefit or a hindrance?

The CMS Collaboration

Close contact with the CMS Production group has been invaluable. CMS Production has set high goals for success that has driven the focus of the MOP team for several months leading to the installation of compatible grid software at the participating sites, integration into production tools, and extensive debugging.

Running on the USCMS test grid was very beneficial to both PPDG and CMS, and Rick Cavanaugh has been very instrumental in that respect.

What is your assessment of the potential for adapting the s/w from this project to other experiments?

It is very good. The requirements are that the experiments provide a software installation method and a mechanism for generating wrapper scripts to run at the remote sites. The underlying software: DAGMAN, Condor-G, GDMP, and Globus must be installed on at least one node of each farm.

What do you see as the future needs, deliverables and effort needed for the Project Activity?

We need the continued support of Peter Couvares to work on the development MOP master and to provide consultation for the production MOP master installation planned for FNAL.

We need the continued support of Alain Roy especially for GDMP in the short term.

Is there anything PPDG should be doing more/differently to help with the project activity?

We need to do a better job of separating the production from the development. A production MOP master site would allow CMS Production personnel to use the software in the field and produce results in real time. This of course implies the need for a development MOP master site as the most valuable long term gains of the project will likely be the bugs that it finds. (But it gets difficult to find support within CMS when the chicken doesn't lay any eggs.)

Speedier and more proactive on documentation.

Devolve some of the more lower level bug finding (ie- at the Globus level), tracking, and fixing to the Globus team if possible. Peter Couvares has borne the brunt of most of this work to date, and it has been very valuable to the MOP project and to CMS; but it is also of general interest to all Globus users.

Reviewers Comments

Appendix A: List of Bugs Found in Grid Infrastructure Tools During MOP

Commissioning

Problem:

GRAM Job submission failed because the job failed when the jobmanager attempted to run it (error code 17)

Solution:

Globus stores the full path to all the programs it runs from its scripts. These are stored in the following files: `$GLOBUS_LOCATION/libexec/globus-sh-tools.sh` `$GLOBUS_LOCATION/libexec/globus-gram-job-manager-tools.sh` The paths to the condor programs were not set. What happened was when Globus was installed for red hat 7.2, `/unsup/condor-test/bin` was in my path, but we don't have binaries in `/unsup/condor-test` for `i387_rh72`. So the installation didn't find the condor programs in my PATH and so couldn't set the variables.

Problem:

the assignment id (id 314) had been silently changed by CERN

Solution:

Changed assignment id

Problem:

Expired host certificate and the person who requested the certificate a year ago was not be in a position to renew it a year later.

Solution:

No clean solution yet. In the future, add a feature to Hawkeye to check for soon-to-expire host certificates as a backup for the Globus reminder emails

Problem:

Bug with `gdmp_publish_catalogue` in GDMP 3.0 alpha2 and alpha3

Solution:

Apply patch provided by Shahzad

Problem:

Grid Sites out of synchronisation. Unfortunately, Globus didn't propagate very helpful info to the client, only "remote side did not like my creds for unknown reason", but logging into the remote system and manually examining the gatekeeper log revealed the more specific error "remote certificate not yet valid".

Solution:

Synchronise all sites using `-s ntp-0.cso.uiuc.edu`. Submitted bug report to globus regarding the lack of a useful error message.

http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=53

Problem:

jobmanager failing because it did not have permission to write to \$GLOBUS_LOCATION/var/jobs directory

Solution:

a) mkdir -m 1777 \$GLOBUS_LOCATION/var/jobs
b) in \$GLOBUS_LOCATION/etc/grid-services/jobmanager, add the following argument to the end of the single line therein: "-state-file-dir \$GLOBUS_LOCATION/var/jobs", where \$GLOBUS_LOCATION has been manually expanded

Problem:

GridFTP had a memory leak and hung if transfers lasted longer than 20 minutes

Solution:

Fixed with new gsift of Globus 2.0

Problem:

proxy expired before Condor-G jobs completed causing Condor-G to appear to be "stuck"

Solution:

a future version of Condor-G will watch out for this for you, send you email before your proxy expires, put your jobs on hold if it expires anyway, and send you another email explaining why it couldn't continue and what to do to pick up where you left off...

Problem: 15 April

GDMP bug in gdmp_publish_catalogue:
Communication Error: read 14bytes (out of 134624544) [Fri Apr 12 18:16:44 2002]
GDMP_Remote_Host::send(): Error: receiving message [Fri Apr 12 18:16:44 2002]
GDMP_Remote_Host::send(): Error: sending message [Fri Apr 12 18:16:46 2002]

Solution:

Shahzad and Alain investigated the problem. Solution, if any, was not recorded.

Problem: 15 April

JM: request failed with error 69 (the job manager failed to create the temporary stdout filename), sending message to client ...resulting in the job failing. However, the filesystem in question is local to the machine, has LOTS of space (>35 GB), has never been full, and the .gass_cache directory and all its parents have all the right perms. Furthermore, simultaneous or near-simultaneous jobs create their output just fine. It's just like a random, inexplicable failure.

This may not be a jobmanager bug, per se, but one thing that the jobmanager could do better is propagate the errno back to us, so we can at least see that it couldn't write the output because of "permission denied" or "device full" or something... right now we know nothing. Jaime is going to look into submitting a patch, or at least a feature-request to the Globus team to do this...

Solution:

can probably be worked around simply by submitting the rescue dag, or adding a DAGMan retry to the node in question

Problem:

451 gdmp processes running on beak and most of them as old as 2 days! caused beak to crash.

Solution:

GDMP Heartbeat and MOP used the same VO, and when GDMP publishes MOP files, it also publishes all heartbeat files (even if they are no longer on disk). Beak then receives this publication into its import catalogue and starts the file transfer. As the heartbeat files are no longer on remote disk, it keeps trying by making stage requests, checking that the file is ready on the remote disk. move to GDMP 3.0 and use different VO's for the GDMP Heartbeat and MOP

Problem:

Inability to easily clean up a remote site after GDMP has published the files-- hence causing the previous problem

Solution:

No good solution yet.

Problem:

If MOP publish script failed in certain way, it prevented itself from being re-run successfully by the rescue DAG.

Solution:

fixed problem in MOP.

Problem:

Not properly checking the return value of globus-url-copy

Solution:

fixed problem in MOP

Problem:

twice seen that the Globus 2.0-beta fork-jobmanager gets stuck in globus_gass_close(), after its forked job has finished and it has sent the gass data back. The first time, the jobmanager was hung and wouldn't respond to gridmanager queries;

the second time it stayed alive and would answer gridmanager queries, but wouldn't actually exit, so the gridmanager wouldn't take the job out of the "run" state, and it just kept it there stuck.

Solution:

No solution yet. In the future, make sure the new gridmanager does something smarter when this happens, since, even if we can fix the bug in the jobmanager, we know old buggy ones are out there

Problem:

GDMP register was supposed to return 0 in the case that a file is already registered.

Solution:

(it's a MOP bug, not a GDMP bug) It returns 0 virtually all the time, in fact, so our post script doesn't rely on the return value, but instead checks whether we see "N of M files registered" in the output, where $N = M$. We just don't take already-registered files into account.

Bug is fixed (I think) in MOP.

Problem:

fork-jobmanager's recovery features are too limited for MOP without lots of extra support.

Solution:

use the condor-jobmanager (running "scheduler universe" jobs) instead. Condor "scheduler universe" jobs are forked locally by the schedd as soon as they're submitted -- so they're just like fork jobs, but with the error-recovery and status-tracking that Condor provides. That way, when everything goes up in flames, MOP won't have to pick up all the pieces itself.

Problem:

GDMP arguments seem to have changed trivially. The "-t" option to `gdmp_register_local_file` previously wanted the argument "flatfile" to publish non-Objectivity files, but now wants the argument "file" instead -- and it's not backwards-compatible.

Solution:

a completely cosmetic change, but it's going to require us to add code to our scripts to somehow detect which gdmp version we're using and then apply the right argument for that version. We may have to do this for other gdmp commands as well (`gdmp_publish_catalogue?`).

Problem:

The gatekeeper runs under inetd, which has code to detect if a service is being flooded or is looping, and temporarily disable it. It turns out if a service starts up 40 times in a short period of time, it triggers this inetd feature, and the service is disabled for 10 minutes. What was happening was that Condor-G was checking the status of the gatekeeper on behalf of 40 jobs in a short period of time, causing inetd to shut down the gatekeeper service, causing Condor-G to detect that it was down, and retry a little while later, causing the whole process to repeat itself indefinitely.

Solution:

simplest solution is to modify the inetd.conf on the gatekeeper machine so that its flooding threshold is higher for the gatekeeper service. This can be done by adding ".400" (or whatever number you'd like) after the nowait entry on the gatekeeper's config line, like so:

```
globus-gatekeeper stream tcp nowait.400 root /.../globus-gatekeeper
globus-gatekeeper -conf /.../globus-gatekeeper.conf
```

Another solution is to throttle the Condor-G gatekeeper "pings", but we could also optimize by doing one ping per machine and caching the answer for a short period, rather than doing it once per job.

Problem:

"Failure: ping successful"? It looks like the gatekeeper authenticates successfully, and then dies with this error.

Solution:

As far as one can tell, however, this turns out NOT to be an error at all, just a terribly misleading log message. This is the normal output of the Globus gatekeeper when it successfully handles such a ping operation. But it wasted a lot of time, and should be fixed. Filed a big report:

http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=67

Problem:

GDMP was refusing to transfer any files because it thought there wasn't enough disk space available

Solution:

fixed (able to track it down, patch GDMP, and rebuild it) and reported to the GDMP development team. Don't know when a new version may come out to help this.

Problem:

Running out of file descriptors causing head node testulix to crash

Solution:

New Bug with no good solution yet! reboot testulix and increase the number of file descriptors (several times). File descriptor limit is now set to over 32000 and testulix still runs out of file descriptors(we think!). It is not currently understood if this is related to the jobmanager, or not. This may be compounded between the # of fd's required for the jobmanager and fd's required for GDMP, etc.

Problem:

found a number of bugs (see below) in the Globus jobmanager that are actively preventing USCMS jobs from completing in the current cmsim run.

Solution:

Here are details on the individual bug-fixes, courtesy of Jaime:

jm-stdio-update.patch

Fixes a bug in the STUDIO_UPDATE signal handling code in globus-job-manager that can cause incorrect handling of a mal-formed RSL. Patched against Globus 2.0 beta. Needs to be submitted as a patch to the GRAM 1.6 branch of the Globus CVS repository.

jm-query.patch

Fixes a bug in globus-job-manager. During processing of a client query, the job status was being read in a non-thread-safe manner (without the appropriate lock being held). Patched against Globus 2.0 beta. Needs to be submitted as a patch to the GRAM 1.6 branch of the Globus CVS repository.

jm-ttl-expire.patch

Fixes two bugs in globus-job-manager related to TTL expiration. Patched against Globus 2.0 beta. Needs to be submitted as a patch to the GRAM 1.6 branch of the Globus CVS repository.

jm-restart-ttl.patch

Fixes two bugs in globus-job-manager. First, the job state file and GASS cache were inappropriately cleaned up on a failed restart (they should be left as-is). Second, a newly-restarted jobmanager failed to detect that a jobmanager was already running due to file buffering in libc. Patched against Globus 2.0 beta. Needs to be submitted as a patch to the GRAM 1.6 branch of the Globus CVS repository.

jm-proxy-expire-hang.patch

Fix bug where if two-phase and save-state are used and the final commit times out, the jobmanager hangs when it thinks that the TTL has expired. At this point in the code, we don't care about the TTL and ignore the expiration, but the hang is caused by a locking bug. Also, we now continue monitoring and updating the TTL during the final flush/close of stdout/err. Patched against Globus 2.0. Needs to be submitted as a patch to the GRAM

1.6 branch of the Globus CVS repository.

jm-restart-stdio.patch

Fixes bug in `globus_1_gram_read_state_file()` where we use the `jm_restart` URL to look up gass cache entries instead of `graml_gass_cache_tag`. Patched against Globus 2.0. May need to be submitted as a patch to the GRAM 1.6 branch of the Globus CVS repository.

jm-close-fds.patch

Fixes bug where fork jobmanager doesn't close any of its file descriptors when forking off the job. May not compile on all platforms. Patched against Globus 2.0. Not a problem in GRAM 1.6.

jm-proxy-timeout-set.patch

Allows the client to set in the RSL how long before proxy expiration the jobmanager should exit with error `USER_PROXY_EXPIRED`. Also changes the default timeout to 60 seconds. Patched against Globus 2.0. Needs to be submitted as a patch to the GRAM 1.6 branch of the Globus CVS repository.

jm-lib-fail.patch

Fixes a couple places in the jobmanager library fork code where the job status was set to `DONE` when it should be `FAILED`. Patched against Globus 2.0. May need to be submitted as a patch to the GRAM 1.6 branch of the Globus CVS repository.

jm-condor-rm.patch

Fixes a bug in `globus-script-condor-rm` where script was looking for the wrong text in the output from `condor_rm`. Caused the jobmanager to think the job removal had failed when it hadn't. Patched against Globus 2.0. May need to be submitted as a patch to the GRAM 1.6 branch of the Globus CVS repository.

openssl-asm.patch

Fix a potential memory-corrupting bug in openssl by disabling the assembly routines (using the C versions instead). Patched against Globus 2.0.

Appendix B: Greg's Answers to Ruth's Questions of 6 May 2002:

Hi Ruth. Thanks for the questions. (Also, I have answered the questions on the standard PPDG internal review document and sent those to Peter, Alan, and Alain for comment. Those should be distributed by tomorrow afternoon.)

>

> 1) The "production" deployment of MOP has seems to have required a large
> fraction of the combined set of US CMS physics grid resources for the past
> few months. Was this inevitable? what could have been done differently to
> reduce this huge load on the many who have contributed so actively over the
> last few months?

>

The deployment of MOP takes place in two phases: A development deployment intended to be at Wisconsin and a production deployment intended to be at FNAL with respect to the "mop-master". Currently, we have but one deployment at Wisconsin which is doubling as both development and deployment right now. This is leading to some problems in getting official results back to CMS production, but it has been fantastic from a development point of view.

Over the past few months, we have utilized resources at Florida and UCSD as "mop-slave" sites and resources at Wisconsin as the "mop-master". In order to explore the underlying technology, it was absolutely necessary to operate with at least two sites: one MOP master and one MOP slave. These sites had to be administratively separated and one might argue had to be geographically separated in order to provide a realistic operating environment. The additional "mop-slave" site was nonetheless very useful for debugging problems arising from mis-configurations.

In my opinion, the load carried by many people in PPDG (and one in CMS ;-)) was inescapable in that any reduction in load would have directly impacted the good results: namely the amount of bugs found, reported, and fixed in globus, Condor-G, DAGMAN, GDMP, etc is linear in the time spent by Peter, Shahzad, and Alain. (Alan DeSmet has just started.) The amount of support from CMS needed to enable this work (a clean integration with IMPALA, lobbying for a production assignment) was directly proportional to the time that I put in.

In the near future, I hope (and propose in the questionnaire) that we move to create a real production MOP master site at FNAL. This assumes that the remaining level of bugs makes it feasible. This should at least decouple the pressures from CMS Production coordinators from the MOP developers. However, I cannot answer the question yet as to whether it is ready to deploy on "non-grid" resources. We may talk about that on Thursday.

>

> 2) Do you have a concern as to how the US CMS PPDG developments

- > overlap/interact with the overall CMS grid developments and program? Is
- > there anything you wish PPDG was doing more or differently to help avoid
- > duplication of effort between the EU and US sides of the program?
- >

This is a wider issue than just MOP, so I will divide my answer:

MOP is essentially a method for packaging CMS Production scripts so that they can run with Condor-G/DAGMAN and Globus. In that respect, I am not concerned with overlap with other CMS grid tools or EU Grid tools since they also tend to use Condor-G/DAGMAN and Globus. A great deal of experience has been gleaned about these packages already, and MOP may either inform other parallel efforts through documentation or may be re-factored into any other parallel efforts.

GDMP is both a replica manager and a file transfer method. The MOP run has exposed some limitations of GDMP as it now exists, but nonetheless GDMP debugging contributed to the overall Globus debugging we have experienced. I am not concerned about overlap with EU grid tools since GDMP is the province of WP2, but the continued PPDG support of GDMP is greatly desired. A lot of time and energy has been invested by CMS to give GDMP a chance to work in a production environment. I am concerned that GDMP overlaps with other CMS file transfer tools, and that if support for GDMP from PPDG wanes then we will lose this important laboratory for doing grid enabled file transfers.

- >
- > 3) Was sufficient and appropriate documentation available for underlying
- > software, interface and protocol specs for the components MOP is
- > integrating? Do you think that this is an area that PPDG should increase its
- > attention and priority to?
- >

I should let others answer the first question. However, I think that it is important that the MOP experiences are documented. I think time is growing short before we start the next big development push for the Summer production starting in June. (Well, that's my story and I'm sticking to it...)

- >
- > 4) How susceptible do you think MOP is going to be to changes in interface
- > in the underlying grid components - for example those included in the
- > Virtual Data Toolkit to date? I know it is difficult to be quantitative
- > here, but we talk at iVDGL meetings about the effort Alain has to put in
- > repackaging and testing VDT when there are new versions of condor, globus
- > and gdmp - but this affects the next layer up as much or more so..?
- >

Since MOP is very generic and is such a thin interface, I think that it is not very susceptible to changes in the underlying interfaces. As long as the interfaces of GDMP are well documented, for example, this calls for

a change in only one DAG node of MOP. If the condor_submit_dag interface changes, then the function within the MOP python code that handles the submit needs to be changed. In my experience adding DAG nodes and cleaning up MOP code while Peter was on vacation, the components were fairly well isolated.

>

> Please ask questions if things seem unclear or unduly pointed. As you
> appreciate the goal of the reviews is to provide constructive ideas and
> input for the steering meeting and planning in May. Thank you, and we look
> forward to talking with you on Thursday.

>

Not at all. I look forward to a productive review!

Thanks, Greg

Appendix C: Peter's Answers to Ruth's Questions of 6 May 2002:

Ruth asked:

>> 1) The "production" deployment of MOP has seems to have required a large
>> fraction of the combined set of US CMS physics grid resources for the
>> past few months. Was this inevitable? what could have been done
>> differently to reduce this huge load on the many who have contributed so
>> actively over the last few months?

By "resources", I assume you mean people? If so, I think in retrospect that this was largely inevitable, yes -- for the simple reason that the grid tools we're relying on, while proven in concept, haven't been used in many (any?) real production applications before, and have a lot of rough edges. There's little we could have done this spring to change that.

We could, however, have recognized it more clearly before starting, and made sure that we had a core developer on board representing each major grid component to help us diagnose and fix problems. One of the reasons we've spent so much time on Globus bugs lately is not because we found fewer bugs in other components (Condor-G, GDMP, etc.), but because we had core developers from those other components on hand to find and fix them immediately. So the turnaround was better.

But the bottom line is that we're early adopters, and we're doing the grid beta-testing whether we like it or not.

Greg wrote:

- > The deployment of MOP takes place in two phases: A development deployment
- > intended to be at Wisconsin and a production deployment intended to be at
- > FNAL with respect to the "mop-master". Currently, we have but one
- > deployment at Wisconsin which is doubling as both development and
- > deployment right now. This is leading to some problems in getting
- > official results back to CMS production, but it has been fantastic from a
- > development point of view.

Right. But the very fact that we need to do all this development is a sign that we're not yet prepared for hands-free production.

I don't think it would be a good idea for us to revert back to some "proof-of-concept" activity -- but I think that we need to recognize the difference between smooth, consistent, hands-free production, and herky-jerky, heavily-babysat production. We're still doing the latter. :-)

- > In my opinion, the load carried by many people in PPDG (and one in
- > CMS ;-)) was inescapable in that any reduction in load would have directly
- > impacted the good results: namely the amount of bugs found, reported, and
- > fixed in globus, Condor-G, DAGMAN, GDMP, etc is linear in the time
- > spent by Peter, Shahzad, and Alain. (Alan DeSmet has just started.) The
- > amount of support from CMS needed to enable this work (a clean integration
- > with IMPALA, lobbying for a production assignment) was directly
- > proportional to the time that I put in.

I agree completely.

- > In the near future, I hope (and propose in the questionnaire) that we
- > move to create a real production MOP master site at FNAL. This assumes
- > that the remaining level of bugs makes it feasible. This should
- > at least decouple the pressures from CMS Production coordinators
- > from the MOP developers. However, I cannot answer the question yet as to
- > whether it is ready to deploy on "non-grid" resources. We may talk about
- > that on Thursday.

Right. My feeling is that we're not there yet, but we're close. It's unpredictable, though, because we consistently discover new bugs only when existing bugs are squashed, and we're able to ramp up production a little more than we had. I think a few times this spring we made the mistake of assuming that once the bugs in front of us were squashed, we'd be ready to rip.

- >> 2) Do you have a concern as to how the US CMS PPDG developments
- >> overlap/interact with the overall CMS grid developments and program? Is
- >> there anything you wish PPDG was doing more or differently to help avoid

>> duplication of effort between the EU and US sides of the program?

This is an area in which I'm confused, to be honest. I've seen a lot of powerpoint presentations, and I get the sense lots of people are making sure the EU and US grand plans don't duplicate each others' work too much. But it's not clear to me what actual software really exists today and might be useful to us, or visa-versa.

It might be useful at one of our cross-project meetings to spend more time just doing some show-and-tell of the current state of our software (not just slides), with the explicit goal of trying to recognize opportunities for immediate collaboration or re-use of software.

I wouldn't want to delay our work waiting for another project's component to be finished, just for the sake of integration. But if related grid software exists, I think we could probably be doing more to utilize it, and feed our experiences back and forth. I'm really just speculating, though, since I don't know what other groups are actually accomplishing today, and how much time it would take to work together.

> MOP is essentially a method for packaging CMS Production scripts so that
> they can run with Condor-G/DAGMAN and Globus.

Right. Although making MOP useful required a lot of integration work with IMPALA, MOP itself is really just a relatively small, simple DAG-generator. It automates the process of generating DAGs to run lots of grid-unaware jobs in parallel on remote sites, creating and filling in the implicit jobs needed to stage data back and forth, etc.

I'm honestly not sure if the actual MOP code will fit cleanly within the GriPhyN virtual-data software, or the EDG software -- both of them will certainly require similar functionality (i.e., a DAG-generator), but may end up re-implementing it within another component, like the "planner".

> GDMP is both a replica manager and a file transfer method. The MOP run
> has exposed some limitations of GDMP as it now exists, but nonetheless
> GDMP debugging contributed to the overall Globus debugging we have
> experienced. I am not concerned about overlap with EU grid tools since
> GDMP is the province of WP2, but the continued PPDG support of GDMP is
> greatly desired. A lot of time and energy has been invested by CMS to
> give GDMP a chance to work in a production environment. I am concerned
> that GDMP overlaps with other CMS file transfer tools, and that if
> support for GDMP from PPDG wanes then we will lose this important
> laboratory for doing grid enabled file transfers.

I'll be honest: I'm not sure we want or need GDMP. I think, on the

testbed, we intended to use it to do things it was not exactly designed to accomplish. But for our purposes, just using a replica catalog and gridftp might be simpler.

>> 3) Was sufficient and appropriate documentation available for underlying software, interface and protocol specs for the components MOP is integrating? Do you think that this is an area that PPDG should increase its attention and priority to?

Documentation is sparse and much of what we know is oral history. We could use more. But I think we need to be careful about exactly what we spend time documenting.

>> 4) How susceptible do you think MOP is going to be to changes in interface in the underlying grid components - for example those included in the Virtual Data Toolkit to date?

>
> Since MOP is very generic and is such a thin interface, I think that it is not very susceptible to changes in the underlying interfaces. As long as the interfaces of GDMP are well documented, for example, this calls for a change in only one DAG node of MOP. If the condor_submit_dag interface changes, then the function within the MOP python code that handles the submit needs to be changed. In my experience adding DAG nodes and cleaning up MOP code while Peter was on vacation, the components were fairly well isolated.

I agree. However, one major caveat is that if MOP is expected to support multiple versions of these tools simultaneously, it may require extensive code to identify which version we're talking to and use the appropriate interface. It's simple only when, like now, we require a single supported version of Globus, GDMP, Condor-G, etc., everywhere. In "production" this may not be possible or desirable.

ûûûûûûû_
ûûûûûûûõ_
http://heppc16.ucsd.edu/cms_tutorial/agenda.html
http://heppc16.ucsd.edu/cms_tutorial/agenda.html
Normal
Normal
Heading 1
Heading 1
Heading 2
Heading 2
Default Paragraph Font

Default Paragraph Font
WW-Absatz-Standardschriftart
WW-Absatz-Standardschriftart
WW-Absatz-Standardschriftart1
WW-Absatz-Standardschriftart1
WW-Default Paragraph Font
WW-Default Paragraph Font
WW-Default Paragraph Font1
WW-Default Paragraph Font1
Numbering Symbols
Numbering Symbols
WW-Numbering Symbols
WW-Numbering Symbols
WW-Numbering Symbols1
WW-Numbering Symbols1
WW-Numbering Symbols11
WW-Numbering Symbols11
Hyperlink
Hyperlink
Heading
Heading
Body Text
Body Text
Subtitle
Subtitle
Doug Olson0C:\users\olson\dev\ppdg-pa\reviews\MOPReview.doc
Doug Olson0C:\users\olson\dev\ppdg-pa\reviews\MOPReview.doc
WW8Num1
WW8Num2
WW8Num3
WW8Num1
WW8Num2
WW8Num3
Unknownŷ_
Times New Roman
Times New Roman
Symbol
Symbol
Courier
Courier
Helvetica
Helvetica
PPDG Internal Review
PPDG Internal Review
Ruth
Doug Olson

Ruth
Doug Olson
PPDG Internal Review
Normal.dot
Doug Olson
Microsoft Word 9.0
Lawrence Berkeley National Laboratory
PPDG Internal Review
_PID_HLINKS
http://heppc16.ucsd.edu/cms_tutorial/agenda.html
http://heppc16.ucsd.edu/cms_tutorial/agenda.html
Root Entry
1Table
1Table
WordDocument
WordDocument
SummaryInformation
SummaryInformation
DocumentSummaryInformation
DocumentSummaryInformation
CompObj
CompObj
ObjectPool
ObjectPool
Microsoft Word Document
MSWordDoc
Word.Document.8