This plan updates and elaborates the plan sent in email Sunday 11/17 and includes some (but not all) updates from our meetings yesterday (Monday)

**Demo Summary**

The purpose of the demo is to show how Chimera's virtual data system can be used to manage science data and workflow, and how it makes use of the iVDGL Grid and VDT technology.

Demo will consist of the following sections (note that former section 1 has been split into 2 parts):

1. Chimera basics
2. Chimera for HEP (ROOT) analysis (if we can polish it).
3. CMS and SDSS production workflow
4. SDSS visual interface and FL technology highlights

The demo will show:

The basics of virtual data, using Chimera running a dag modeled after the SDSS workflow (about 10 DVs)

HEP ROOT examples on a laptop, to show the derivation of data files in a scientific analysis setting.

The mechanism of running high-volume production of simulated HEP collisions, with Chimera managing the workflow and tracking the provenance of results.

An interactive interface to a SDSS galaxy-cluster-finding application where users can request images of the sky on the MicroMural (a large 2x3 screen tiled display), and interact with a background catalog production process.

**Teamwork**

PLEASE help by making sure you know what to do, who you need to integrate with, and by COMMUNICATING efficiently with the whole team.

Keep and communicate detailed lists of action items for you and others

Know the deadlines and demo delivery schedule

Understand the demo delivery environment

Understand how your pieces will integrate with others.

**Demo Configuration**

6 grid sites: UF (3), UW-M, ANL-DG, UC; approx 400 hosts, maybe 500 CPUs?

2 submit hosts: chalant and testulix

GridGanglia running everywhere

dagrunner, dagstat, dagmon, dagtrace running on the submit sites. (I will define these later, but they are as we discussed on the phone).
ROOT and Chimera databases running on ept; maybe others too.
Chimera pg db on shoveled
DBs are in chunk format.
Webserver (w jsp?) on testulix? – needed to support workview?

**Demo delivery environment**
Demo will be given in two places within the ANL booth:
In Globus side of ANL booth with 2 monitors
- one monitor driven by laptop w/ ppt
- one monitor driven by laptop or workstation w/ LINUX: xterms, browsers, and ROOT
In FL side of ANL booth with 1 monitor and microMural (uM)
- one Linux box, oddball2, driving Monitor
- all of above running on oddball2
- printed ppt slides in binder to augment demo

Since, on the Globus side, the demo could be given at either one of two stations (in full, with three displays available) or on a single-display workstation, or the FL station, with only a single display available, the only difference is that on the single display station we will do without powerpoint slides there. We will attempt to have the powerpoint slides placed in a binder for display.

**Demo Software Tools**

*worksubmit*: takes a set of daxes to run, queues them up, and then runs them

*workrund*: the background daemon that runs a workload. Terminology: a workload is multiple workflows for a single purpose; a workflow is a set of jobs linked as a dag; a job is the execution of a single app.

(there is one “workrund” loop on a submit host somewhere for each “VO” that we are representing here: SDSS, CMS, and “SC”, the demo VO.

*workstat*: (-l option) shows a list of dags not yet complete:
- dagid dagfile logfile (and maybe other stats like time queued, time started)
(-s option) shows counts of workflows by:
  - status: queued / running / completed (succeeded, failed)
  - status by site
  - status overall
jobs running since…

jobs completed last 15 minutes, last hour, last 12 last day?
(basically – are we making progress?)

*worktrace*: Jens's dag ploticus plotter, takes a completed dagid and shows it.
**workwatcher:** translate the simple textual output of dagstat -s into a nice HTML page with graphics. (Ian suggested/requested this!); I have an updating option

**workview:** takes a list of dagid's from dagstat and pops up updating graphviz plots of their status (Suchindra's monitor in a wrapper)

workview contains two types of monitoring screens – an overall workload monitoring screen, and a workflow monitoring screen. Each of the screens are repeatedly updated.

### Demo Construction Overview

The demo will use the test grid, the site status of which is displayed at [www.grihyn.org/workspace/grids](http://www.grihyn.org/workspace/grids). Jens will serve as the master demo-integrator and tester. He will ensure that all the necessary pieces are in the right place and working together for demo deliver

We’ll work as 4 interacting teams:

**Team 1: Core grid development (grid, monitoring, work-* toolkit)**
   - Jens, Adam, Catalin, Suchindra

**Team 2: CMS application**
   - Adam (with assistance from Rick as needed)

**Team 3: SDSS application**
   - Jim, Vijay, Suzzanne, Michael, Joe

**Team 4: Chimera ROOT demonstration**
   - Yong, Rick, Mike

#### Team 1: Core Grid Development

[ ] Jens: create demo Step#1: mini-monsters

[ ] Jens: sdss-make demo

[ ] Jens: develop worktrace from current ploticus tool
devolve dagwork
integrate demo tools

[ ] Jens: speed up Chimera tools to support SDSS.

[ ] Catalin: create workbalance script for workrund (deliver to Adam) – low prio

[ ] Suchindra: get workview (web page) ready for demo

[ ] Suchindra: enhance workview to start a new monitor on the next running job in the run list some delay after the current job finishes.

#### Team 2: CMS

[ ] Adam: stable version of worksubmit, workrund, workstat. To be installed on testulix, chalant, hamachi (by

[ ] Adam: CMS code installed at all 6 sites
Adam: CMS jobs (long and short) entered in workrunner; enter enough to take us through next Monday. Should keep running after the show is over!

Adam: Verify that workstat shows the accurate details at each submit pool.

Adam: (optional) integrate workbalance script from Catalin into workrund.

Team 3: SDSS

Vijay: new task: get sdss science code on all sites

Michael: get midsize-VDL through gendax (done? How many are ready?)

Michael: get midsize-VDL through genedag (stuck on genedag problem????)

Adam: work with Michael to get a workrund running on chalant

Michael: get SDSS workload running on chalant

Michael: delivery catalog files from midl-level VDL workload to Vijay

Vijay: install catalog files from mid-level VDL into demo for interactive viewing

Team 4: Chimera ROOT Demo

Rick, Yong: Chimera/ROOT demo

Setup demo software on the three (or all – 5 total) Globus and one FL workstation (oddball2):
Logins, scripts, X settings, any other demo tools or props. Powerpoint slides / PDFs, or just use laptop for PDFs

Demo Delivery Scenarios

0. Preparation

Ensure that the following is running in the test grid, and set up on the displays:

- ganglia running
- workrund running on three submit sites: chalant, testlix, and hamachi
- weeks supply of demo-monsters running at a load-level of one per site, to keep one demo-monster running all the time. Should contain mix of CPU jobs that sleep and run.
- Weeks supply of CMS jobs running load level decided by Adam; can be adjusted during show.
- Weeks supply of SDSS jobs running at load level decided by Adam and Jens, coordinated with Michael, when they are ready to run.
- Workview (web page) running on a monster (needs to be restarted ever time a monster finishes?) – can workview be enhanced to start running on the next job in a workload once the job its watching terminates?

1. Chimera basics - Jens, can you do a 2-step demo similar to last years:

step a) on a single mock-sdss dag, show execution and make-semantics under shell planner
step b) run a middle size monster across *all* pools and show dagview and dagtrace of it.

Show Catalin's GridGanglia while monster and other background work is running
2. Show a sequence of Physics analysis steps in click-next fashion, highlighting event plots, histograms, a variety of file types, metadata, good searches, and if possible: a) a make-style remake of data objects after a parameter change, with a visible histogram change, b) a small parameter sweep loop after a "mainline" analysis sequence was "perfected", and 3) good provenance "stack traces"

3. Show that a large volume of CMS work is running, with Ganglia and with a cms-specific "workview" output display

4. Same as 3 but for SDSS

5. When demo-ing on the microMural, show the interactive SDSS sequence

**Other Demo Work**

As an aside, we need to regularly test and/or monitor that the 6 sites and 2 submit-hosts are fully functional.

The tests should be done at the end of the day and as early in the morning Tues, Wed, and Thu as possible.

The tests should consist of a mini-monster run at each site plus a check of demo segments 1 & 2, and a check to make sure that all background workflow is still running correctly and making progress.

**Demo Work Checklist**

Catalin: ensure that Ganglia is correctly monitoring all 6 exec sites, and that the output is as pretty and as clear as possible. Will be sending you frequent enhancement/tune-up requests.

Adam: create workstat and workmon; get and keep the CMS work running through Thursday

Michael: get and keep the SDSS work running through Thursday; create medium-catalog data ASAP; full running soon after

Vijay: integrate sdss catalog from Michael into interactive demo; test with Joe on Susanne's interface

Suchindra: create dagview; test asap on Jen's large dags to make sure it scales.

Mike: demo handouts and presentation slides; SDSS Chimera talk for Thursday.