The USCMS Integration Grid Testbed (IGT)

Greg Graham
Fermilab Computing Division

ISGC 2003
Academia Sinica, Taipei, Taiwan
10-March-2003
What is the IGT?

- USCMS is planning for three distinct “Grids”:
  - Development Grid Testbed (DGT)
    - Focus: testing new middleware products and speculative development projects
    - Hardware: The existing USCMS R&D Grid Testbed
  - Integration Grid Testbed (IGT)
    - Focus: release testing of approved middleware products, focussed development for deployment, documentation and support
  - Production Grid
    - Focus: Providing reliable Grid services to the collaboration

- Tight coordination with the LCG is important!
What is the IGT? (cont’d)

▶ Scenario: Releasing new grid software
  - 1. New grid software is tested on the DGT
    ✓ Either new middleware or newly developed grid software
  - 2. Grid software that passes muster is released to the IGT for further integration testing.
    ✓ Sensible focussed development for release hardening and compatibility
    ✓ Deployment planning, documentation, and support planning
  - 3. Grid software that passes integration testing is released for production.
    ✓ The IGT should be running approximately the same software, but on some fraction of the production resources (“rolling prototype”)
The IGT/DGT have been operational for about 1 year.

- However, middleware problems stopped us from participating in the Spring02 production
  - GASS cache issues
  - Not resolved until August 2002 - with much help from Condor team at UW Madison and Globus team
- Middleware re-engineered and tested successfully in September 2002.
  - 50K events generated in September during 2 week “engineering run” with the revamped middleware.
History of the IGT (cont’d)

- The IGT was commissioned in October 2002
  - After middleware fixes were proven on the USCMS Development Grid Testbed.
  - For our first “integration run” on the Grid, we had assigned all USCMS production resources to the IGT.

- The Production Grid will be commissioned in March 2003
  - Identifying and collecting support personnel
  - Accumulating documentation and support experience
  - Putting software quality control procedures in place
US CMS R&D Development Grid Testbed Resources (Fall 2002)

- **Fermilab**
  - 1+5 PIII dual 0.700 GHz processor machines
  - 0.2 TB dedicated disk + Mass Storage
- **Caltech**
  - 1+3 AMD dual 1.6 GHz processor machines
  - 0.4 TB dedicated disk
- **San Diego**
  - 1+3 PIV single 1.7 GHz processor machines
  - 0.07 TB dedicated disk
- **Florida**
  - 1+5 PIII dual 1 GHz processor machines
  - 0.5 TB dedicated disk
- **Wisconsin**
  - 5 PIII single 1 GHz processor machines
  - 0.02 TB dedicated disk

**Total:**
- ~41 1 GHz dedicated processors
- ~1 TB dedicated storage

- **Operating System: Red Hat 6**
  - Required for Objectivity

Adding groups at Rice, MIT, and Princeton in 2003.
The Current IGT - Hardware

Fermilab: 40 dual 750 MHz nodes + 2 servers, RH6
Florida: 40 dual 1 GHz nodes + 1 server, RH6
UCSD: 20 dual 800 MHz nodes + 1 server, RH6
    New: 20 dual 2.4 GHz nodes + 1 server, RH7
Caltech: 20 dual 800 MHz nodes + 1 server, RH6
    New: 20 dual 2.4 GHz nodes + 1 server, RH7
UW Madison: Not a prototype Tier-2 center, support

CERN: LCG Participates with
    72 2.4 GHz CPU at RH7

Total: 240 0.8 equiv. RH6 CPU
      152 2.4 GHz RH7 CPU
Software Distribution

- IGT Software is (mostly) distributed using PACMAN
  - PACMAN caches set up for the USCMS R&D Grid Testbed and the IGT
    - The VDT is distributed this way
    - CMS Executables can in principle be distributed this way also
  - PACMAN keeps track of what is installed at each site
    - HTTP based package retrieval
    - MDS based monitoring system to come
  - Software not yet fully distributed by PACMAN:
    - CMS Executables, GroupMan, Clarens
Middleware

- Virtual Data Toolkit (VDT) 1.1.3
  - Virtual Data Client
    - Globus Toolkit 2.0 (with improved GASS cache)
    - DAGMAN: A package that models production jobs as Directed Acyclic Graphs
    - Condor-G 6.4.3: A backend that allows DAGMAN to manage jobs on Globus Job Managers
  - Virtual Data Server
    - (the above, plus:)
    - mkgridmap: A tool to help manage the gridmap authorization files
    - GDMP 3.0.7: The EDG WP2 replica manager
    - ftsh: A UW Madison bash-like shell with semantics to handle fault tolerance
Virtual Organization

Virtual Organization Management

- LDAP server maintained at FNAL
  - Keeps all DNs of USCMS Grid Users
- GroupMan (from EDG, PPDG)
  - Periodically downloads LDAP information into grid mapfiles
- Uses DOE Science Grid CA
  - Also accepts Globus CA for EDG compatibility and for certain file transfers
- Currently using a “group certificate”, but will update to using personal certificates soon
CMS Software

- Monte Carlo Production Software
  - Pythia/GEANT distributed in tar files
  - Digitization and Reconstruction distributed using DAR
    - DAR is a SCRAM based tool for packaging CMS software and configuring the runtime environment, and these can be distributed using PACMAN

- Job Preparation
  - MC_RunJob
    - General metadata based workflow planner
    - Chains above CMS executables together into super-jobs
    - Produces output compatible with the legacy Impala based CMS production system - *important for CMS buy-in*
MOP

- MOP is a system for packaging production processing jobs into DAGMAN format
  - DAGMAN format is a Directed Acyclic Graph (DAG)
  - MOP uses the following DAG Nodes for each job:
    - Stage-in: Stages in needed application files, scripts, data from the submit host
    - Run: The application(s) run on the remote host
    - Stage-out: The produced data is staged out from the execution site back to the submit host
    - Clean-up: Temporary areas on the remote site are cleansed
    - Publish: Data is published to a GDMP replica catalogue after it is returned
MOP (cont’d)

- Mop_submitter wraps Impala jobs in DAG format at the “MOP master” site
- DAGMAN runs DAG jobs through remote sites’ Globus JobManagers through Condor-G
- Results are returned using GridFTP. Though the results are also returned to the MOP master site in the current IGT running, this does not have to be the case.

**UW Madison is the MOP master for the USCMS Grid Testbed**

**FNAL is the MOP master for the IGT and the Production Grid**
Monitoring

- MonaLisa is used as the primary Grid-wide monitor.
  - Dynamic discovery of monitoring targets and schema
  - Implemented with Java/Jini with SNMP local monitors
  - Interfaces to/from MDS implemented at FNAL and Florida
    - Important since most schedulers will talk to MDS
  - Interfaces with local monitoring systems
    - Ganglia interface used at Fermilab

- IGT Monitoring information was not used for automatic scheduling
  - For human benefit only so far
The CMS IGT “Stack”

The CMS IGT “stack” comprises nine layers. The Application layer contains only CMS executables. The Job Creation layer comprises CMS provided tools MCRunJob and Impala. Neither MCRunJob nor Impala are specifically “grid aware.” Then there is a DAG Creation layer and a Job Submission layer. Both functionalities are provided by MOP. Jobs are submitted to DAGMAN which, through Condor-G, manages jobs run on remote Globus Job Managers. Finally, there is a local Farm or Batch System used by Globus GRAM to manage jobs. In the case of the IGT, the local Batch manager was always FBSNG or Condor. Scheduling and Integrated monitoring are not present.
Completed CMS Production on the IGT

- Assignment to produce 1 Million “eGamma Bigjets” events by Christmas 2002, all steps.
  - About 500 sec per event on a 750 MHz processor
    - Dominated by the cmsim step
  - Can run only on RH6 machines because of Objectivity licensing issues

- Assignment to produce 500K additional events, cmsim step only
  - This runs on the USCMS Tier-2 hardware that is currently RH7
IGT Results

Time to process 1 event: 500 sec @ 750 MHz

Speedup: Avg factor of 100 speedup during current run

Resources: Approximately 230 CPU @750 MHz equiv

Sustained efficiency: about 43.5%

1M fully simulated and reconstructed events!
IGT Results

Time to process 1 event:
500 sec @ 750 MHz

Speedup:
Avg factor of 100 speedup during current run

Resources:
Approximately 230 CPU @750 MHz equiv.

Sustained efficiency:
about 43.5%
Analysis of Inefficiencies

- Failure semantics currently lean heavily towards automatic resubmission of failed jobs
  - Sometimes failures are not recognized right away
  - Need better system for spotting chronic problems

- The “wrong” people were often the ones to start troubleshooting problems
  - At one point, an application problem was misdiagnosed early as a GASS cache issue
  - Once the application expert looked into it, the problem was solved in 90 minutes
    - But middleware experts had already spent days :-(

11-March-2002 CMS Integration Grid Testbed
IGT Results

» Manpower Estimates
  » 2.65 FTE equivalent during initial phase and debugging
    ✓ Reported voluntarily in response to a general query
  » 1.1 FTE equivalent during smooth running periods
    ✓ The MOP person plus periodic small file transfers
  » Expected to be less than 1 FTE when regular shift procedures are adopted

» File Transfers are a small job so far
  » Only ntuples are kept in much of the production
  » All input files were staged from Fermilab and output staged to Fermilab using globus-url-copy
    ✓ ie- No replica catalog was used DURING production
Comparison to Spring 02 and EDG Stresstest Efforts

- CMS Spring 2002 “Manual” Production
  - CPU utilization was 10-40% during Spring02
    - Comparison not fair because: I don’t know how many CPU should be counted in the denominator and the IGT currently doesn’t have any significant file transfers
  - Manpower was a lot higher

- Comparison to EDG Stresstest
  - Also completed CMS Production in Fall 2002
  - More manpower than IGT run, but also more advanced functionality was tested.
    - We have gained much experience from both “bottom up” IGT and “top down” EDG approaches.
CAIGEE: Remote Analysis System

- **Data Processing Tools**
  - interactive visualisation and data analysis (ROOT, etc)

- **Data Catalog Browser**
  - allows a physicist to find collections of data at the object level

- **Data Mover**
  - embedded window allowing physicist to customise data movement

- **Network Performance Monitor**
  - allows a physicist to optimise data movement by dynamically monitoring network conditions

- **Computation resource browser, selector and monitor**
  - allows a physicist to view available resources (primarily for dev. stages of Grid)

- **Storage resource browser**
  - enables a physicist to ensure that enough disk space is available

- **Log browser**
  - enables a physicist to get direct feedback from jobs indicating success/failure, etc

Many promising alternatives: currently in the process of prototyping and choosing.

Picture taken from Koen Holtman via Rick Cavanaugh
Clarens: Used at SC2002

Clarens is part of CAIGEE that allows for analysis of data residing in remote ROOT ntuple files.

Results produced on the IGT were deposited at remote sites and plots were updated in real time on the SC2002 showfloor!
Conclusions

- The CMS Integration Grid Testbed is a success!
  - First step towards Production Grid services in the US
  - Expect to work closely with the LCG to provide production grid services by Summer 2003

- Software Quality Control Procedures being implemented
  - Distributed Processing Environment (DPE) portion of USCMS S&C project
    ✓ The IGT is DPE 1.0
References and Acknowledgements

- For more information, please see http://www.uscms.org/scpages/subsystems/DPE/
- Thanks to the IGT Team!
  - Caltech: Erik Aslakson, Julian Bunn, Saima Iqbal, Harvey Newman, Suresh Singh, Conrad Steenberg
  - UC SanDiego: Jim Branson, Ian Fisk, James Letts
  - U Florida: Paul Avery, Dimitri Bourilkov, Rick Cavanaugh, Suchindra Kategari, Jorge Rodriguez
  - U Wisconsin: Peter Couvares, Alan DeSmet, Miron Livny, Alain Roy, Todd Tannenbaum