Infrastructure for Distributed Analysis

Matevž Tadel

PROBLEM: Provide real-time access to distributed data-storage and CPU resources

In contrast to batch jobs, DA requires immediate response (few minutes):

1. Only staged data really interesting
   Users / user-groups could perform data pre-selection with staging and pinning.
2. When queues are full, jobs can not be spawned when needed
   Computing centers do not provide direct access to nodes nor queues.
   Pull model allows job prioritization on the level of a Virtual Organization.
3. Synchronized operation of distributed jobs
   Results must be merged on the fly with intermediate results observable by the user.

1) & 2) provided by `AliEn`; PROOF is the natural choice for 3)
  PROOF slaves must be started in advance

Glue between the components written by the ARDA team: A.Peters, D.Feichtinger
  User side: provide ROOT classes for user-`grid`-PROOF interaction
  Service for registration of available PROOF slaves

Graphical UI & 3D visuzalization implemented in the Gled framework

Gled is a ROOT-based `C++` framework/toolkit; extends ROOT's functionality for:
- management of object collections and object-interaction (w/GUI)
- dynamic 3D visualization (OpenGL)
- distributed computing (hierarchical server-client model)

Gled = Generick Lightweight Environment for Distributed computing, `http://www.gled.org/`

Main purposes of the DA visualization:

1. `grid`-interaction: monitoring, exploration, visualization

   Allow non-expert users to *browse* the `grid` and display results in different formats
   World-map views: display data in geographical context
   Visualization of open connections and data-transfers

2. Instruction: explaining different elements of the system to new users

3. Showing-off: the demo presented at SuperComputing-2004 trade show

   It **is** important that things look good (small part of the development but very effective).

ALICE Distributed Analysis Demo

Abstraction of basic elements

Virtual environment: world map and amphitheatre

DA User .vs. services (AliEn and PROOF)

Interaction with AliEn

Connect / authenticate

Query sites: select those that participated in the DC
  could include SE/CE status (display on the map)

Query data-set: for DC data the directory structure is sufficient $\rightarrow$ file-loaction query
  need meta-data or event pre-selection in real world (AliEn findEx command)
  display number of files per site
  the data set could undergo further manipulation (unions, exclude data from given site, etc)

Interaction with PROOF

Connect

Send data-set to PROOF master $\rightarrow$ PROOF parses the data-setconnects to available slaves

Start the analysis: event-loop started on PROOF slaves
  PROOF master steers the process and sends progress reports / intermediate results to the user

Status

- The Great Stalemate with `gLite` prototype (thou shall not deploy)
- PROOF being improved even further

Unrealized plans for Distributed Analysis

Deploy DA for users:

impossible before new AliEn deployed and configured (file-catalog updates)

standard ROOT interface provided by ARDA needs to be merged with ROOT development

Web interface via CAROT (ROOT Apache module):

Simplicity of Google: user enters search-path, query and analysis macro

$\rightarrow$ the results appear on the same web-page (with updates)

CAROT can also record the session $\rightarrow$ analyze grid performance / re-play with graphics

Visualization stuff mentioned during the talk:

futher interaction with AliEn: provide interface to CE status, queues and jobs

more detailed visualization / task specific display modes