FACADE – Financial Analysis Computing Architecture in Distributed Environment

V. Motoška, L. Slebodník, M. Jurečko, M. Zvada

May 4, 2011
Outline

- Motivation
- CADE Middleware
- Future work
Motivation

What?

- financial markets prediction and analysis
- simulations of statistical and computational finance models
  - From the Currency Rate Quotations onto Strings and Brane World Scenarios \(^1\) (Phys.Rev.E)
  - Agent Based Simulations of Prediction Algorithms
  - Memory Prediction Framework Model for Clustering the Financial Market Behaviors

\(^1\)http://arxiv.org/abs/1104.4716
Motivation

How?

- **Hardware**
  - Intel Xeon E5530 quad-core based worker nodes
  - NVIDIA GeForce GTX 480, GTX 285, Tesla C1060

- **Software**
  - CentOS 5.5 x86_64 managed by Puppet
  - Latest Red Hat MRG Condor with our middleware
  - Central authentication and account management via LDAP and Kerberos
  - GNU Octave, MATLAB, ROOT
  - Custom applications developed in Python, Perl, C++, OpenCL

Why?

- **Condor**
  - huge processing power (HTC)
  - high customizability
  - community support

- **MRG**
  - messaging for our Java based trading application
  - almost real-time computation (algorithmic trading)
  - rebuilt under CentOS in mock environment

---

[^3]: [http://fedoraproject.org/wiki/Projects/Mock](http://fedoraproject.org/wiki/Projects/Mock)
Why?

- **NVIDIA GPUs** – according to our tests
  - best price vs. performance
  - OpenCL maturity and completeness
  - Linux drivers stability
- **CentOS**
  - RHEL clone
  - MRG compatible

---

4 http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5695244
Challenges

- long running jobs
  - processing quite large amount of WORM data
  - producing comparable amount of results

- data manipulation
  - portable (data locality independence)
  - reliable (fault tolerance during computing)
  - optimal (file transfer control)

- isolate users from unnecessary technical details

- obtain a higher degree of Condor customizability
To solve storage related problems we introduced CADE URL

cade:// sors/ users/ foo/bar

- taking care not to overwrite concurrently modified data
- possibility to append data to the existing files
- warning on any possibly unsafe action
- map concrete storage according to configuration file
UI

Submit

Stage In

WN

Exec

Stage Out

translate

cade://sors/users/johndoe/data/file

/home/users/johndoe/data/file
/home/users/johndoe/data/file

cade://sors/users/johndoe/data/file

1. /export/home/users/johndoe/data/file (on NFS)
2. copy out to Condor sandbox (/var/execute/1234/…/data/file)
1. `/export/home/users/johndoe/data/file` (on NFS)

2. copy out to Condor sandbox (`/var/execute/1234/.../data/file`)

Executing job...
1. **/export/home/users/johndoe/data/file** (on NFS)
2. copy out to Condor sandbox (**/var/execute/1234/.../data/file**)

1. **output URL (cade://sors/results/strings/1.csv)**
   - **SOURCE:** /var/execute/1234/home/.../1.csv
   - **TARGET:** ftp://carina.sors.local/strings/1.csv
2. copy in SOURCE to TARGET

---

**FURT Solutions, s.r.o.**

**Condor as a resource for financial market analysis**

**May 4, 2011**
Benefits of CADE URLs

- **portability**
  - GRID/SRM ready
  - can access various data back-ends
    - NFS/dCache/PSE/Xrootd/gFTP...
- **reliability**
  - protection against accidental overwrites
  - no storage SPOF during job execution
- **optimization possibilities**
  - storage side filtering/preprocessing
  - online data compression
  - caching
CADE Job Description Language

- similar to `condor_submit` syntax (no need for backward compatibility)
- domain specific syntax
- additional layer covering various submission methods
  - `condor_submit/Web Services/QMF`
- created for user convenience
Example .jdl file

#!/usr/bin/env cade_submit

arguments = subdir%ix%/infile %o1%       # o1 alias
executable = copyProgram

input = $HOME/subdir%ix%/                # stage in

output = "output file %ix%"            # $(ProcID) + 2
output = common.txt :append            # do not overwrite
stdout = %ix%/stdout.txt

parametric = 2:5
Example .csf file

arguments = subdir%ix%/infile cade://sors/.../work_dir/outfile%25ix%25
+cade_argument0 = "c3ViZGlyJWl4JS9pbmZpbGU="
+cade_argument1 = "Y2FkZTo8L3NvcnMvdXN1cnMvcmV2x1Ym9kbmlrL3N2bmtvL2NhZGUvdWkvdGVzdmVMDNfY2FkZV9zdWJtaXQ="
+cade_arguments = true
+cade_cwd = "cade://sors/.../work_dir/
+cade_executable = "cade://sors/.../work_dir/copyProgram"
+cade_input0 = "cade://sors/.../home/.../subdir%25ix%25"
+cade_orig_cwd = "/home/.../work_dir/
+cade_orig_executable = "copyProgram"
+cade_orig_input0 = "/home/users/.../subdir%ix%/
+cade_orig_jdl = "cade://sors/.../work_dir/03_positional_templates.jdl"
+cade_orig_output0 = "outfile%ix%"
+cade_orig_output1 = "common.txt:append"
+cade_output0 = "cade://sors/.../work_dir/outfile%25ix%25"
+cade_output1 = "cade://sors/.../work_dir/common.txt%3Aappend"
+cade_parametric_end = 5
+cade_parametric_start = 2
+cade_stdout = "cade://sors/.../work_dir/%25ix%25/stdout.txt"
+cade_submit_directory = "cade://sors/service/submits/ui01dev.cade.sors.local/000083_2011-04-27/"

executable = 03_positional_templates:copyProgram
log = /service/submits/ui01dev.cade.sors.local/000083_2011-04-27/logs/condor_user
notification = Complete
notify_user = slebodnik@sors.com
transfer_executable = false
transfer_input_files = /service/submits/ui01dev.cade.sors.local/000083_2011-04-27/csf
transfer_output_files =
universe = vanilla
wantremoteio = false
whentotransferoutput = ON_EXIT
queue 4
CADE Service directory

- **structure**
  
  ```
  /submits/
  <submitter>/
  000001_day/
  job/

  /jobs/
  <submitter>/
  <cluster id>/
  submit/
  stages/
  0001_<stage name>/
  <section id>/<worker node>/0001_<stage name>/

  /users/
  <user>/
  jobs/
  <submitter>/
  <cluster id>/
  submits/
  000001_day/
  ```

- **contents and purpose**

  - middleware logs, Condor user and failure logs
  - working directory snapshots for each stage
  - job and machine class adds
CADE Development

- development environment
  - XCP based (fast re-provisioning)
  - Condor instances and configuration per developer
    - $ dev_env_reset
    - $ dev_env_run all query
    - $ dev_env_run head install
    - $ dev_env_run ui01dev start
- integration tests – following whole job life-cycle
  - from submission (condor_submit interoperation)
  - through run on worker nodes
  - to results checking
Future work

- migrate Cfengine to Puppet
- move service directory from NFS
- Cumin LDAP integration (ready on Dev env)
- cade_query
  - interactive monitoring
  - status of each executed stage
  - history of stages on worker
- user requests: enhanced mail notifications, application checkpointing (DMTCP)…