



The CAVES Project
Collaborative Analysis Versioning
Environment System
The CODESH Project
Collaborative DEvelopment SHell

Dimitri Bourilkov

University of Florida

GriPhyN

DPF 2004, Riverside, CA, USA, August 30, 2004



Virtual Data

- Most scientific data are not simple “measurements” \Rightarrow produced from increasingly complex computations (e.g. reconstructions, calibrations, selections, simulations, fits etc.)
- HEP (and other sciences) increasingly CPU/Data intensive: **Programs and how-to** become a vital intellectual resource of the scientific community \Rightarrow ***need new ways to collaborate***
- **Virtual data** are data products with a ***well defined method of (re) production***; “virtuality” with respect to existence \Rightarrow can define data products for future production or record the “history” of products that exist now or have existed in the past

Log data provenance by tracking how new data is derived from transformations on other data

We already do this, but manually!

Virtual Data Motivations

- Data **track**-ability and result **audit**-ability: "Virtual Logbook"
 - In the nature of science
 - Reproducibility of results
- **Tools** and **data** sharing and collaboration (data with "recipe")
 - Individuals **discover** other scientists' work and build from it
 - Different Teams can work in a **modular, semi-autonomous** fashion: **reuse** previous data/code/results or entire analysis chains
- **Repair** and **correction** of data – c.f. "make"
- **Workflow** management, **Performance** optimization: data staged-in from remote site **OR** re-created locally on demand?
- **Transparency** with respect to location and existence

The Metaphor



- A cave is a secure place to store stuff
- Usually you need a key to enter
- Stuff can be retrieved when needed (and if the temperature is kept constant, usually in good shape)
- Small caves can be private, larger are usually owned cooperatively
- When a cave is full, a new one is build
- To get something, one starts at the local cave and, if needed, widens the search ...



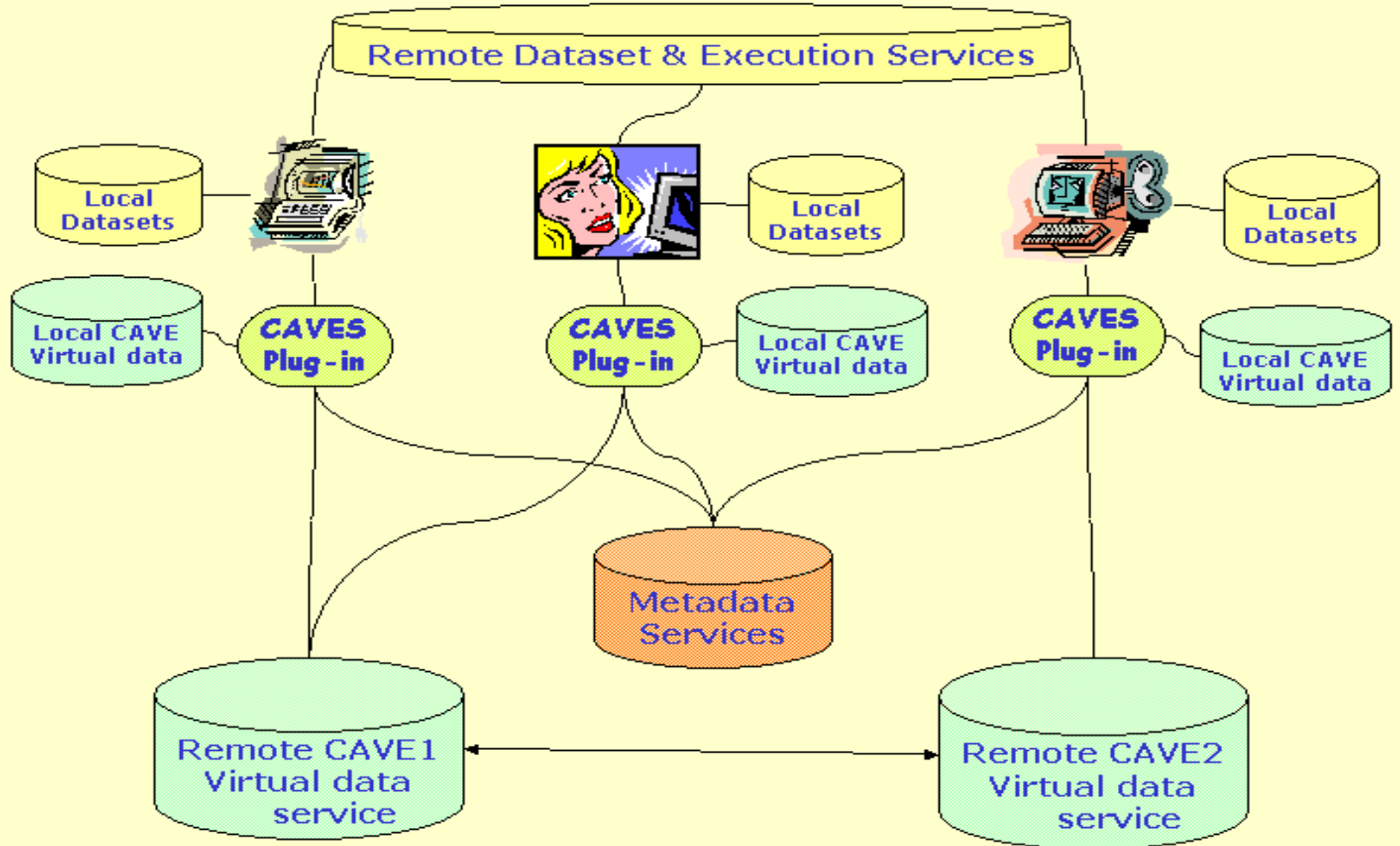


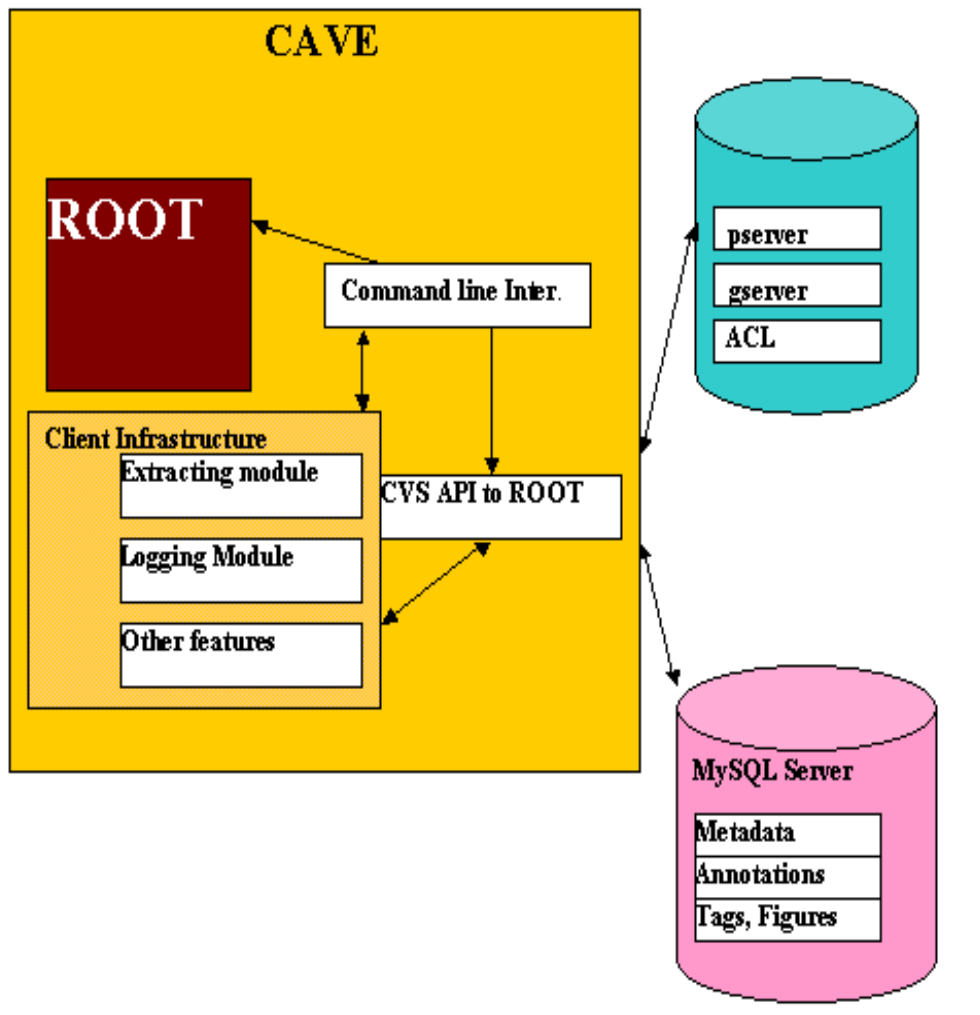
CAVES / CODESH Projects



- Concentrate on the interactions between scientists collaborating over extended periods of time
- Seamlessly log, exchange and reproduce results and the corresponding methods, algorithms and programs
 - Automatic and complete logging and reuse of work or analysis sessions (between checkpoints)
- Extend the power of users working or performing analyses in their habitual way, giving them virtual data capabilities
- Build functioning collaboration suites (stay close to users!)
- First prototypes use popular tools: Python, ROOT and CVS; e.g. all ROOT commands and **CAVES** commands available

CAVES / CODESH Architectures – Scalable and Distributed





- **Three tier architecture:** isolate client from back-end details; different implementations possible
- **Lightweight clients** (use ROOT; C++; Python; e.g. CVS API)
- **Back-ends: e.g. CVS pservers** (remote stores) with read/write access control; **ARCH, Clarens** etc
- Optional MySQL servers for **metadata** (fast search for large data volumes)



CAVES / CODESH Architectures – CVS Server Backend



- **Sandbox programming** – work on per session basis
- CVS provides **version control** by tagging releases
- **CVS tags** act as **unique IDs** for virtual data products (the namespace can be structured by a collaborating group e.g. one big cave or many barrels in a cave, selected on a session basis)
- Both **local** and **remote** modes of working
- CVS pservers (secure, efficient remote stores):
 - Only CVS user accounts with password authentication, no UNIX accounts on the server (gridmapfile uses same idea)
 - read/write access control lists (per user & directory)



Possible scenarios



Case1: Simple

User 1 : Does some analysis and produces a result with tag ***analX_user1***.

User 2: Browses all current tags in the repository and fetches the session stored with tag ***analX_user1***.

Case2: Complex

User 1 : Does some analysis and produces a result with tag ***analX_user1***.

User 2: Browses all current tags in the repository and fetches the session stored with tag ***analX_user1***.

User 2: Does a modification in the program obtained from the session of **user1** and stores the same along with a new result with tag ***analX_user2_mod_code***.

User 1: Browses the repository, finds that his program was modified and decides to extract that session using the tag ***analX_user2_mod_code***.

This scenario can be extended to include an arbitrary number of steps and users in a working group or groups in a collaboration.



Annotations



- Log **annotations** (& possibly **summary results** or **metadata**) in the repository /data equivalence!/
 - **Brief** annotations – use `cvs -m “ Annotation” ...`
 - Optional **complete** annotations - possibly MySQL servers for metadata (fast search for large data volumes e.g. with indexing)
 - The users can browse (a subset of) the existing tags, inspect the annotations, and, if interested, reproduce the results (or just get the howto), modify them etc.



Extensible Command Set



- Session commands

- open <session>
- close <session>

- During analysis

- help <command>
- browse <tag>
- inspect <tag> <b|c>
- startlog
- log <tag> <annot>
- annotate <tag>
- extract <tag>

- Administrative tasks

- copy <tag> <from> <to>
- move <tag> <from> <to>
- delete <tag> <from>
- archive <tag> <to>
- retrieve <tag> <from>

Implemented / To do

CODESH commands:

run, shell
getenv
getalias etc



Working Releases - CAVES



```
Terminal
File Edit Settings Help
bourilkov@vitosh: rctest

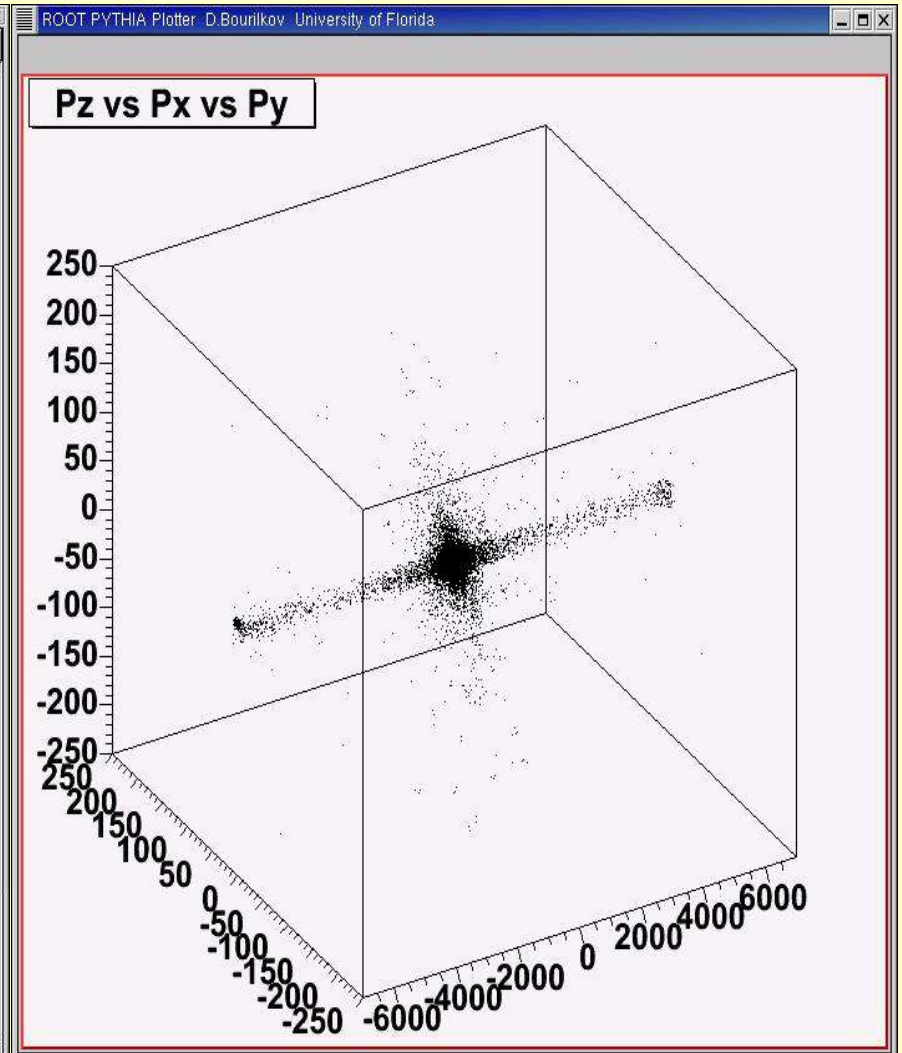
*****
*           Welcome To CAVES                               *
* Collaborative Analysis Versioning Environment System     *
*                                                         *
*           Dimitri Bourilkov & Mandar Kulkarni           *
*           University of Florida                          *
*           Gainesville, USA                              *
*                                                         *
*           You are Welcome to visit our website          *
*           http://ufgrid02.phys.ufl.edu/~bourilkov        *
*****

Please set the cvs pserver OR hit enter for default
CAVES:
Pserver for this session: :pserver:test@ufgrid02.phys.ufl.edu:/home/caves

*****
*           TO GET STARTED:                               *
*           just type help at the command prompt         *
*                                                         *
*           commands beginning with '.' are delegated to  *
*           ROOT                                          *
*****
CAVES: browse
U v01/mylog.1
higgs-ww-plotpxpypz-500      (revision: 1.5)
higgs-ww-plotpxpypz-100     (revision: 1.4)

CAVES: extract higgs-ww-plotpxpypz-500
*****Storing data for usage...*****
ROOT Command is :.x
Macro is :dbpit1web.C
Macro is :dbpit1web.C
U data/dbpit1web.C
You have [0] altered files in this repository.
Are you sure you want to release (and delete) directory `data': y
Argument is 500
Argument is input "http://ufgrid02.phys.ufl.edu/~bourilkov/higgs.root"
Argument is output "higgs-ww-plotpxpypz-500"
Command is :.x dbpit1web.C(500,"http://ufgrid02.phys.ufl.edu/~bourilkov/higgs
.root","higgs-ww-plotpxpypz-500")
TFile**      higgs-ww-plotpxpypz-500.root
TFile**      higgs-ww-plotpxpypz-500.root
KEY: TCanvas canv2;1 ROOT PYTHIA Plotter D.Bourilkov University of Flori
da
.x dbpit1web.C(500,"http://ufgrid02.phys.ufl.edu/~bourilkov/higgs.root","higg
s-ww-plotpxpypz-500")

*****End*****
You have [0] altered files in this repository.
Are you sure you want to release (and delete) directory `v01': y
CAVES: █
```





Working Releases - CODESH



```
Terminal
File Edit Settings Help
dimil@localhost: codesh.py
*****
*                               *
*      C O D E S H               *
*                               *
*  Collaborative DEvelopment SHell *
*                               *
*      Dimitri Bourilkov        *
*                               *
*      bourilkov@phys.ufl.edu    *
*                               *
*****
Change defaults y/n ? :
*****
*                               *
* Type help OR ? at the command prompt *
* to get a full list of commands or *
* help about individual commands *
*                               *
* To EXIT just type the usual ctrl-D *
*                               *
*****
/home/bourilkov/tmp/codesh [1] ?

Documented commands (type help <topic>):
-----
EOF          browse      cd          export      extract
getalias     getenv     help        inspect     log
run          setenv     shell

/home/bourilkov/tmp/codesh [2] browse
browse <tag>:
demo_01          (revision: 1.7)
test-01-005     (revision: 1.6)
test-01-004     (revision: 1.5)
test-01-003     (revision: 1.4)
test-01-002     (revision: 1.3)
test-01-001     (revision: 1.2)

/home/bourilkov/tmp/codesh [3] browse test-01-005
browse <tag>: test-01-005
test-01-005     (revision: 1.6)
Command is: df
Macro: dbtest.py
Command is: dbtest.py
Macro: stringcount.py
Command is: stringcount.py dimi.outagain.txt a
/home/bourilkov/tmp/codesh [4] extract test-01-005
extract <tag>: test-01-005
Command is: df
Filesystem      1k-blocks      Used Available Use% Mounted on
/dev/hda5        2016016        190928   1722676   10% /
/dev/hda9        5044156        2850724   1937200   60% /home
/dev/hda11       17504716       11651112   4964408   71% /tmp
/dev/hda6        2016016        1560904   352700    82% /usr
Macro: dbtest.py
Command is: dbtest.py
Hello COLLABORATION!
Macro: stringcount.py
Command is: stringcount.py dimi.outagain.txt a
String a is => 14 (<= times in file => dimi.outagain.txt
/home/bourilkov/tmp/codesh [5] EOF
```

- Virtual log-book for “shell” sessions
- Parts can be local (private) or **shared**
- Tracks environment variables, aliases etc during a session
- Reproduce complete working sessions
- Complex CMS ORCA example operational



Working Releases



- The client codes are easy to download and install (stored in CVS) – checkout and build
- All you need to run the clients is ROOT 3.05 or higher and CVS (from v 1_0_0 client is fully ROOT-compliant); or Python 1.5 or higher
- Users can browse the virtual data catalogs and reproduce the examples stored there
- They can play with e.g. new analyses and store them locally or on our server, or install new servers for groups collaborating on a project



- Work in progress – **first** CAVES and CODESH releases out
- We are looking forward to **user feedback**
- Possible **future directions**:
 - Different back-ends: web/grid service oriented
 - Extend **remote data access**: e.g. Clarens, xrootd ...
 - Add **GSI security**
 - **Automatically convert session log to workflow**
 - Tune on smaller samples, **schedule on grid** for larger tasks

A picture is better than 1000 words: Try out the releases !

CAVES white paper arXiv: physics/0401007

More info at <http://cern.ch/bourilkov/caves.html>