



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" ⇒ 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 ⇒ need new ways to collaborate
- Virtual data are data products with a well defined method of (re) production; "virtuality" with respect to existence ⇒ 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







CAVES / CODESH Architectures





- 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)







- 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)



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.

Possible scenarios

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	ROOT PYTHIA Plotter D.Bourilkov University of Florida
File Edit Settings Help	
bourilkov@vitosha: rltest	
**************************************	Pz vs Px vs Py
* * * * Dimitri Bourilkov & Mandar Kulkarni * * University of Florida * * Gainesville, USA *	
* * You are Welcome to visit our website * * * * * * * * * * * * * * * * * * *	250
Please set the cvs pserver OR hit enter for default	200
CAVES: Pserver for this session: :pserver:test@ufgrid02.phys.ufl.edu:/home/caves	150
**************************************	100
* * * commands beginning with '.' are delegated to ROOT * ***********************************	
biggs-ww-plotpxpypz-500 (revision: 1.5)	-50
higgs-ww-plotpxpypz-100 (revision: 1.4)	-100
CAVES: extract higgs-ww-plotpxpypz-500 ***********************************	-150
Macro is :dbpit1web.C Macro is:dbpit1web.C U data/dbpit1web.C	-200
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 dbpitlweb.C(500,"http://ufgrid02.phys.ufl.edu/~bourilkov/higgs	200
.root","higgs-ww-plotpxpypz-500") TFile** higgs-ww-plotpxpypz-500.root TFile* higgs-ww-plotpxpypz-500.root	50 50
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	-50 -100 -20004000000
s-ww-piptpxpypz-500°) жжжжжжжжжжжжжжжжжепджжжжжжжжжжжжжжжжжжж	-150 -200 -4000
You have [U] altered files in this repository. Are you sure you want to release (and delete) directory `v01': y CAVES:	-250 -6000

CAVES / CODESH projects



Working Releases - CODESH



Ello Edit Sottor	25.3				
The East Setting	is Help				
<mark>/imi@localhos</mark> t	: codesh.py				
*****	******	*******	кж		
<u>.</u>	CODECH		*		
	CUDESH		*		
COllabora	tivo DEvolopmont	SHo11	ж Ф		
< COLLADORE	cive beveropment	onerr	*		
< Dim	itri Bourilkov		*		
ĸ			*		
< bouri	lkov@phys.ufl.edu		ж		
ĸ			*		
*****	*****	*****	кж –		
Change default	sy/n ?:				
******	*****	****	кж		
k Tupo bolo (P 2 at the common	d promot	*		
k to get a f	ull list of comma	nde on	*		
k heln abou	t individual comm	ands	*		
<			*		
K To EXIT jus	t type the usual	ctrl-D	*		
ĸ			*		
*****	*****	*****	кж		
/home/bourilkc	w/tmp/codesh [1]	?			
S	manda (h	(+			
Jocumented com	mands (type neip	(topic):			
TOF	browco	cd	evport	extract	
retalias	geteny	heln	inspect	log	
10	seteny	shell			
demo_0 test-0 test-0 test-0	1 1-005 1-004 1-003		revision: 1.7) revision: 1.6) revision: 1.5) revision: 1.4) revision: 1.3)		
test-C test-C 'home/bourilkc prowse <tag>: test-C Command is: c 'acro: dbtest. Command is: c facro: stringc Command is: c</tag>	11-002 11-001 test-01-005 11-005 If py btest.py ount.py tringcount.pu dim	browse tes	revision: 1.2) st-01-005 (revision: 1.6) 1.txt a		
test-C test-C /home/bourilkc prowse (tag): test-C Command is: c Macro: dtbest Command is: c Macro: stringc Command is: s /home/bourilkc	11-002 11-001 12-005 11-005 11-005 11-005 16 10-005 10	browse tes i.outagair extract te	revision: 1.2) st-01-005 (revision: 1.6) n.txt a sst-01-005		
test-0 /home/bourilko prowse (tag): test-0 Command is: command is	11-002 11-001 test-01-005 11-005 11-005 14 py lbtest.py count.py tringcount.py dim v/tmp/codesh [4] test-01-005	browse tes i.outagair extract te	revision: 1.2) st-01-005 (revision: 1.6) n.txt a sst-01-005		
test-C thome/bourilkc prowse (tag): test-C Command is: c dacro: dbtest. Command is: c dacro: stringc Command is: s fhome/bourilkc extract (tag): Command is: c	M1-002 M1-001 v/tmp/codesh [3] test-01-005 M1-005 M2 py botest.py count.py tringcount.py dim v/tmp/codesh [4] test-01-005 f	browse tes i.outagair extract te	revision: 1.2) st-01-005 (revision: 1.6) n.txt a sst-01-005		
test-C test-C thome/bourilkc prowse <tag>: Command is: c dacro: dbtest. Command is: c dacro: stringc Command is: c Adacro: stringc Command is: c extract <tag>: Command is: c Filesystem</tag></tag>	<pre>h1-002 h1-001 hv/tmp/codesh [3] test-01-005 hf py blotest.py count.py tringcount.py dim v/tmp/codesh [4] test-01-005 f 1k-blocks outputs</pre>	browse tes i.outagair extract te	revision: 1.2) st-01-005 (revision: 1.6) 1.txt a sst-01-005 Available Use%	Mounted on	
test-(/home/bourilkc prowse (tag): test-(Command is: c Macro: dtbest. Command is: c /home/bourilkc command is: c /home/bourilkc command is: c /dev/hda5	M1-002 M1-001 test-01-005 M1-005 M1-005 M2 bletest.py bletest.py bletest.py iount.py tringcount.py dim w/tmp/codesh [4] test-01-005 f 1k-blocks 2016016	i.outagair extract te 190928	revision: 1.2) st-01-005 (revision: 1.6) n.txt a sst-01-005 Available Use% 1722276 10%	Mounted on	
test-C thome/bourilkc prowse (tag): test-C Command is: c Macro: stringc Command is: s /home/bourilkc extract (tag): Command is: c ilesystem /dev/hda5 /dev/hda5	11-002 11-001 12-001 12-005 14-005 14-005 19 10-05 19 10-05 10	i.outagair extract te Used 190928 2850724	revision: 1.2) st-01-005 (revision: 1.6) h.txt a sst-01-005 Available Use% 1722676 10% 1937200 60 206400 744	Mounted on / /home	
test-C test-C (home/bourilkc prowse <tag>: Command is: c dacro: dbtest. Command is: c dacro: stringc Command is: c flactor: stringc Command is: c flactor: stringc Command is: c filesystem fdev/hda9 /dev/hda9 /dev/hda9</tag>	11-002 11-001 12-001 14-005 14-005 14-005 16 py 16 15 15 16 16 16 16 16 16 16 16 16 16	i.outagair extract te Used 190928 2850724 11651112	revision: 1.2) st-01-005 (revision: 1.6) 1.txt a sst-01-005 Available Use% 1722676 1005 1937200 60% 4964408 71 352700 99%	Mounted on / /home /temp	
test-C /home/bourilkc browse <tag>: test-C Command is: c Macro: dbtest Command is: c /home/bourilkc extract <tag>: Command is: c /home/bourilkc extract <tag>: Command is: c filesystem /dev/hda5 /dev/hda5 /dev/hda11 /dev/hda11</tag></tag></tag>	11-002 11-001 12-001 12-005 14 15-005 14 15-005 14 14 14-005 14 14-005 14 14-01-005 14 14-blocks 2016016 5044156 17504716 2016016	i.outagair extract te Used 190928 2850724 11651112 1560904	revision: 1.2) st-01-005 (revision: 1.6) h.txt a est-01-005 Available Use% 1722676 10% 1937200 60% 4964408 71% 352700 82%	Mounted on / /home /temp /usr	
test-C test-C /home/bourilkc prowse <tag>: test-C Command is: c Macro: stringc command is: s /home/bourilkc extract <tag>: Command is: c filesystem /dev/hda5 /dev/hda5 /dev/hda6 Macro: dbtest. Command is: c</tag></tag>	M1-002 M1-001 NV/tmp/codesh [3] test-01-005 MF py Botest.py count.py tringcount.py dim NV/tmp/codesh [4] test-01-005 MF 1k-blocks 2016016 5044156 17504716 2016016 py brest.py	i.outagair extract te Used 190928 2850724 11651112 1560904	revision: 1.2) tt-01-005 (revision: 1.6) h.txt a sst-01-005 Available Use% 1722676 10% 1937200 60% 4964408 71% 352700 82%	Mounted on // /home /temp /usr	
test-C test-C /home/bourilkc prowse (tag): test-C Command is: c Macro: dbtest. Command is: c /home/bourilkc command is: c /home/bourilkc Command is: c /dev/hda5 /dev/hda5 /dev/hda5 /dev/hda6 Macro: dbtest. Command is: c Home/bourilkc /dev/hda6	11-002 11-001 12-001 12-005 13 14-005 14 py 15-005 14 10-005 14 14 14 14 14 14 14 14 14 14	i.outagair extract te 190928 2850724 11651112 1560904	revision: 1.2) st-01-005 (revision: 1.6) 1.txt a est-01-005 Available Use% 1722676 100 1937200 60% 4964408 71% 352700 82%	Mounted on // /home /temp /usr	
test-C /home/bourilkc browse <tag>: test-C Command is: c Macro: dbtest. Command is: c /home/bourilkc command is: c /home/bourilkc command is: c /dev/hda5 /dev/hda5 /dev/hda5 Macro: dbtest. Command is: c Hello COLLABOB Macro: stringc</tag>	11-002 11-001 12-001 11-005 11-005 11-005 14 py 15-05 14 14-05 14 14-05 14 14-05 14 14-0105 14 14-0105 14 14-0105 14 14-0105 14 14-0105 14 14-0105 14 14-0105 14 14-0105 14 14-0105 14 14-0105 14 14-0105 14 14-0105 14 14-0105 14 14-0105 14 14-0105 14 14-0105 14 14-015 14 14 14 14 14 14 14 14 14 14	i.outagair extract te 190928 2850724 1165112 1560904	revision: 1.2) st-01-005 (revision: 1.6) h.txt a est-01-005 Available Use% 1722676 10% 1937200 60% 4964408 71% 352700 82%	Mounted on / /home /temp /usr	
test-C thome/bourilkc prowse <tag>: test-C Command is: c Macro: dbtest. Command is: c Macro: stringc command is: c filesystem /dev/hda5 /dev/hda5 /dev/hda5 /dev/hda6 Macro: dbtest. Command is: c Hello COLLABOR Garco: stringc Command is: s</tag>	<pre>14-002 14-001 14-001 14-005 14-005 14-005 14 19 10 14-005 14 14-005 14 14-0105 14 14 14 14-0105 14 14 14 14 14 14 14 14 14 14 14 14 14</pre>	i.outagair extract te Used 190928 2850724 11651112 1560904 i.outagair	revision: 1.2) tt-01-005 (revision: 1.6) 1.txt a set-01-005 Available Use% 1722676 10% 1937200 60% 4964408 71% 352700 82% 1.txt a	Mounted on / /home /temp /usr	
test-(/home/bourilkc prowse (tag): test-(Command is: c /home/bourilkc command is: c /home/bourilkc /command is: c /home/bourilkc /command is: c /home/bourilkc /dev/hda5 /dev/hda5 /dev/hda5 /dev/hda1 /dev/	<pre>id=002 id=001 id=001 itest=01=005 if py istringcount.py dim iv/tmp/codesh [4] test=01=005 if 1k=blocks 2016016 5044156 17504716 2016016 py btest.py ATIDN! ount.py dim 14 <= times in f</pre>	i.outagair extract te 190928 2850724 11651112 1560904 i.outagair ile => dir	revision: 1.2) tt-01-005 (revision: 1.6) 1.txt a est-01-005 Available Use% 1722676 100 1937200 60% 4964408 71% 352700 82% h.txt a h.txt a h.txt a	Mounted on / /home /temp /usr	

- 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 ROOTcompliant); 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

D.Bourilkov