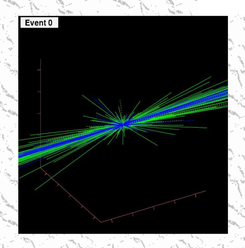
Automatic Tools for Enhancing the Collaborative Experience in Large Projects

CHEP 2013, Amsterdam, the Netherlands, 2013

Presented by

Dimitri Bourilkov, Jorge L. Rodriguez

University of Florida, Gainesville, FL 32611, USA
Florida International University, Miami, FL 33174, USA



Heraclitus asked: How can you bathe in the same river twice?
Quine answers: It's easy, though it is hard to bathe in the same water twice.

Abstract

With the explosion of big data in many fields, the efficient management of knowledge about all aspects of the data analysis gains in importance. A key feature of collaboration in large scale projects is keeping a ***log*** of what is being done - for private use and reuse and for sharing selected parts with collaborators and peers, often distributed geographically on an increasingly global scale. Even better if the log is ***automatically*** created on the fly without the need for the scientist or software developer to alter his or her usual habits. This saves time and enables teams to work more effectively. The **CODESH (COllaborative DEvelopment SHell)** - and **CAVES (Collaborative Analysis Versioning Environment System)** projects address this problem in a novel way. They build on the concepts of ***virtual states*** and ***transitions*** to enhance the collaborative experience by providing automatic persistent virtual logbooks. **CAVES** is designed for sessions of distributed data analysis using the popular ROOT framework, while **CODESH** generalizes the same approach for any type of work on the command line in typical UNIX shells like bash or tcsh. Repositories of sessions can be configured dynamically to record and make available the knowledge accumulated in the course of a scientific or software endeavor. Access can be controlled to define logbooks of private sessions or sessions shared within or between collaborating groups. A typical use case is to build working scalable systems for analysis of Petascale volumes of data as encountered in the LHC experiments. Our approach is general enough to find applications in many fields.

CAVES / CODESH Projects

- **Concentrate on the interactions between scientists collaborating over extended periods of time**
- **Automatic and complete logging and reuse of work or analysis sessions (between checkpoints)**
- **Extend the power of users working or performing analyses or developing software in their habitual way, giving them virtual logbook capabilities**
- **Build functioning collaboration suites (stay close to users!)**
- **Seamlessly log, exchange and reproduce results and the corresponding methods, algorithms and programs**
- **Working releases use popular tools: C++, Python, ROOT**
- ***CVS, SVN, Git and plain files backends***
- **All ROOT/shell commands + CAVES/CODESH commands available**

Glossary

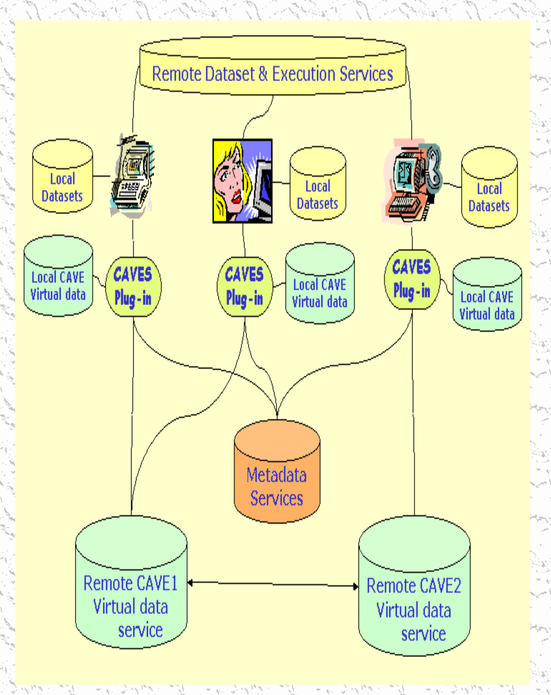
Virtual state: can be reproduced on demand

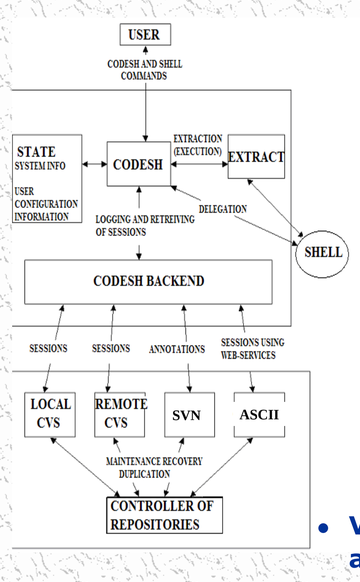
|State> = |Logged part, Environment>
this (somewhat arbitrary) split is called a collaboration contract

Virtual transition: all actions to go from initial |I> to final |F> state
|F> = T |I> **recorded (logged) automatically**

Virtual session: delimited by virtual states (checkpoints)

CAVES / CODESH Architectures – Scalable and Distributed





Backends: ASCII (flat files), CVS, SVN or Git Servers

- Work on **per session** basis
- We provide **version control** by tagging virtual sessions (like CVS **tag**s)
- **Tags** act as **unique IDs** for virtual sessions (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
- **Flat file (ASCII)** backend: local or remote using **ssh** and **scp**
- **CVS pservers/SVN/Git servers** (secure, efficient remote stores):
- Only CVS/SVN/Git user accounts with password authentication, no UNIX accounts on the server
- CVS and SVN servers in production, Git backend under development
- read/write access control lists (per user & directory)

Extensible Command Set

During work/analysis

- **help <command**
- **browse <tag> <search>**
- **inspect <tag> <b|c**
- **log <tag> <annotation>**
- **extract <tag>**
- **record <file>**

Administrative tasks

- **tagcopy <tag> <to> <from>**
- **tagdelete <tag> <from>**

CODESH commands:

run, shell
getenv, getalias
setenv etc

Virtual directory tools: provides easy log/reuse of whole directory trees; same tagging mechanism and persistent backend repositories as for virtual logbooks

- **takesnapshot <tag> <annotation>**
- **getsnapshot <tag>**

Possible scenarios

Case1: Simple
User 1 : Does some work and produces a result with tag ***workX_user1.***
User 2: Browses all current tags in the repository and fetches the session stored with tag ***workX_user1.***

Case2: More Complex, etc.
User 1 : Does some work and produces a result with tag ***workX_user1.***
User 2: Browses all current tags in the repository and fetches the session stored with tag ***workX_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 ***workX_user2_mod_code***.
User 1: Browses the repository, finds that his program was modified and decides to extract that session using the tag ***workX_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.

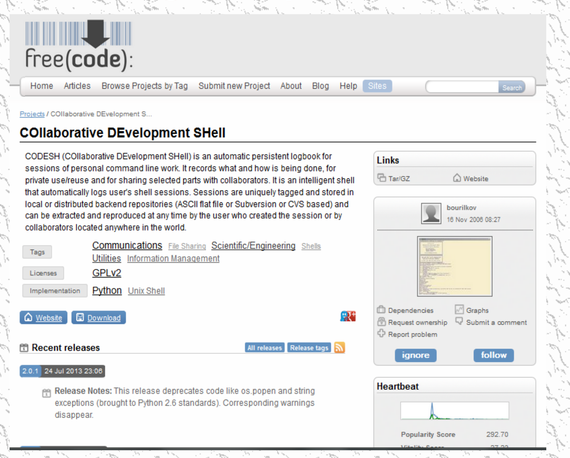
The Metaphor

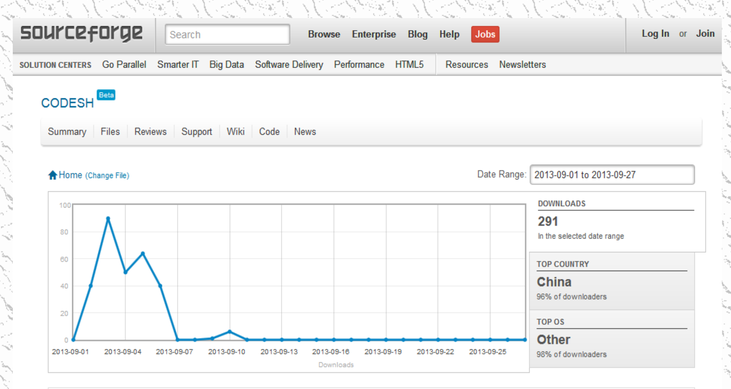
- A cave is a secure place to store stuff
- Usually you need a key to enter
- Stuff can be retrieved when needed (and if the temperature is kept constant, usually in good shape)
- Small caves can be private, larger are usually owned cooperatively
- When a cave is full, a new one is build
- To get something,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

Open source projects

http://freecode.com/projects/codesh

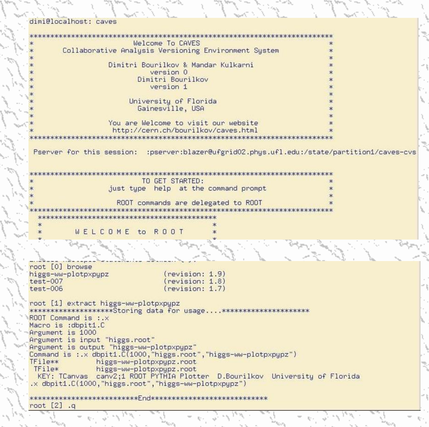
http://sourceforge.net/projects/codesh

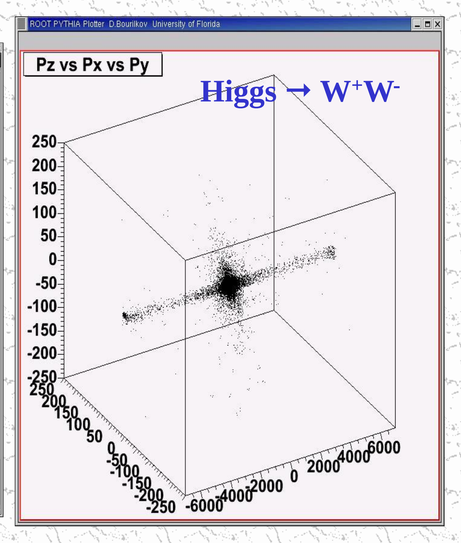




Working Releases

- **Virtual log-book for "shell" or analysis sessions**
- **Parts can be local (private) or shared**
- **Tracks environment variables, aliases, executed scripts etc during a session**
- **Reproduce complete working sessions**
- **Complex Particle Physics examples operational**





Outlook

- Work in progress – **stable** CAVES and CODESH releases available
- New features driven by **user feedback**
- **Future directions**:
  - Different back-ends: adding **Git**
  - **Web interface for the file sharing tools**
  - Automatically convert session log to **workflow**

Home Page:

http://cern.ch/bourilkov/caves.html