HERA-DP@Max-Planck Institut für Physik

Andrii Verbytskyi

DPHEP Collaboration Workshop
Geneva,
June 9, 2015

Content

- **HERA-DP@MPP goals and strategy**
- **HERA-DP@MPP status**
  - **ZEUS**
  - **H1**
- **Problems and what is possible to learn from them**
- **Conclusions**

HERA-DP@MPP goals and strategy

HERA-DP@MPP goals

The MPP aims to preserve the data of the following experiments:

- **ZEUS@HERA**
- **H1@HERA**
- OPAL@LEP
- JADE@PETRA

The main intention is to provide facilities for the physics analysis and to do physics analysis for those experiments.

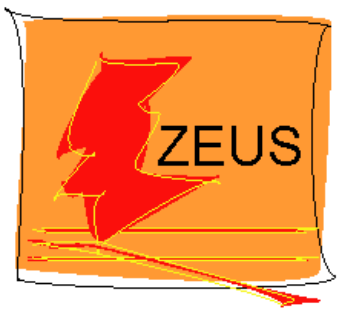
HERA-DP@MPP strategy

The general strategy is:

- Save the bits(files) $\rightarrow$ copy the data from DESY to RZG[1];
- Provide an access to the saved bits(files) $\rightarrow$ define access and archiving policy;
- Save the software $\rightarrow$ provide installation of virtual machines with the software;
- Save the documentation $\rightarrow$ rely on DESY and open databases like InSpire;
- Share experience $\rightarrow$ provide documentation for the software preservation and software deployment.

[1] Rechenzentrum Garching, Gießenbachstraße 2, D-85748 Garching

ZEUS-DP@MPP



Bits for ZEUS@MPP

ZEUS data to preserve: processed data and MC in ROOT/PAW format and corresponding logs.
All of the data is now at RZG:

- 300Tb;
- 1 Mio files.

Access for ZEUS@MPP

- ZEUS data is accessible worldwide via gsidcap, webdav, gridftp, srm protocols:
  - https://grid-dav.rzg.mpg.de:2880/zeus/
  - srm://grid-srm.rzg.mpg.de:8443/pnfs/rzg.mpg.de/data/zeus/
  - gsidcap://grid-srm.rzg.mpg.de:22128/pnfs/rzg.mpg.de/data/zeus/
  - gsiftp://grid-gftp2.rzg.mpg.de:2811/pnfs/rzg.mpg.de/data/zeus/
- To access the data via all protocols grid certificate and ZEUS Grid VO membership is needed;
- Soon most of the data will be moved to tape archives leaving only the most popular data/MC samples permanently on-line.

Documentation for ZEUS@MPP

- MPP policy is to rely on ZEUS-DP@DESY and open databases like InSpire.
  - http://zeusdp.desy.de
  - http://inspirehep.net/search?p=collaboration:%27ZEUS%27&ln=en
  - Internal notes accessible on InSpire with password

ZEUS software

ZEUS analysis software logically split in four parts.

- ROOT5 is the main analysis framework (PAW is deprecated);
- ZMCSP is a MC production package, that contains generators, conversion utilities, software for detector simulation with GEANT3 (so far provided as is);
- CNINFO is a file catalogue package, contains a text file with parsing scripts and SQLite3 database with ROOT/C++ interface (MPP only);
- ZEVIS is an event display based on Qt/ROOT.

In addition packages that are widely used in HEP are considered: g++, gfortran/g77, python, make, CLHEP, cernlib, fastjet, HEPMC, pythia8 and others.

Software preservation for ZEUS

- The analysis[2] software is preserved in a form of CentOS5/CentOS6/CentOS7 x86_64 customised ISO images;
- The images include ZEUS software packaged in default RedHat rpm packages;
- The images can be installed automatically and w/o Internet access via default build-in RedHat anaconda installer;
- The images suitable for real machines and all kind of i386/x86_64 emulators with CD drive support;
- The images can be found on https://wwwzeus.mpp.mpg.de/dphep.html (use default ZEUS password if you are ZEUS member/contact ZEUS management otherwise);

[2] Not reconstruction

Documentation for ZEUS-DP@MPP

- A short documentation is available on https://wwwzeus.mpp.mpg.de/dphep.html;
- A set of scripts for creation of customised ISOs is provided as well;
- Technical details on customisation of ISO images can be found in RedHat documentation.

ZEUS analysis in DP mode step-by-step

VM installation

- Create a virtual machine (e.g. with VirtualBox, qemu ...);
- Get a ZEUS DPHEP ISO image from https://zeus-files.mpp.mpg.de/software/images;
- Boot the virtual machine from the ISO image for automatic, kickstart-based, offline installation.

VM installation

VM setup

Setup a network for VM, e.g. using nat for VirtualBox:

```
1 VBoxManage modifyvm yourvirtualmachinename --nic1 nat
  VBoxManage modifyvm yourvirtualmachinename --natpf1 "myguestssh,tcp,,yourportnumber,,22"
```

Prepare to access the data

To access data with a Grid certificate, create a proxy ...

```
someuser@somempppc:~$ ssh lxplus.cern.ch -l cernuser
Warning: Permanently added the RSA host key for IP address 'yyy.yyy.yy.yy' to the list of
    known hosts.
Password:
* ************************************************************
* Welcome to lxplus0087.cern.ch, SLC, 6.6
* Archive of news is available in /etc/motd-archive
* Reminder: You have agreed to comply with the CERN computing rules
* http://cern.ch/ComputingRules
* Puppet environment: production
* Puppet hostgroup: bi/inter/plus/live/login
* LXPLUS Public Login Service
* ************************************************************
[cernuser@lxplus0087 ~]$ cat zeus-grid-voms.desy.de
"zeus" "grid-voms.desy.de" "15112" "/C=DE/O=GermanGrid/OU=DESY/CN=host/grid-voms.desy.de"
    "zeus"[cernuser@lxplus0087 ~]$
[cernuser@lxplus0087 ~]$ voms-proxy-init -voms zeus -valid 300:00 --vomses=./zeus-grid-
    voms.desy.de --out=./mygridproxy.txt
Enter GRID pass phrase for this identity:
Contacting grid-voms.desy.de:15112 [/C=DE/O=GermanGrid/OU=DESY/CN=host/grid-voms.desy.de]
    "zeus"...
Remote VOMS server contacted succesfully.

grid-voms.desy.de:15112: The validity of this VOMS AC in your proxy is shortened to 691200
    seconds!

WARNING: VOMS AC validation for VO zeus failed for the following reasons:
        LSC validation failed: LSC file matching VOMS attributes not found in store.
        AC signature verification failure: no valid VOMS server credential found.

Created proxy in ./mygridproxy.txt.
```

Prepare to access the data

. . . and copy it to VM:

```
Your proxy is valid until Sun Jun 21 09:02:48 CEST 2015
[cernuser@lxplus0087 ~]$ exit
logout
Connection to lxplus.cern.ch closed.
someuser@somempppc:~$ scp   cernuser@lxplus.cern.ch:mygridproxy.txt ./
Warning: Permanently added the RSA host key for IP address 'xxx.xxx.xx.xx' to the list of
     known hosts.
Password:
mygridproxy.txt
                                          100% 5839      5.7KB/s   00:01
```

The Grid proxy is usefull also for mounts:

```
[someuser@io ~]$  mkdir zeus
[someuser@io ~]$ gfalFS ./zeus davs://grid-dav.rzg.mpg.de:2880//zeus
[someuser@io ~]$ ls zeus
CNINFO_mysql.sql  CNINFO_sqlite3.db  CNINFO_txt.txt  jade  MPISP-CentOS-7.0-1406-x86_64-
     DVD.iso  z  zeus
[someuser@io ~]$
```

Login to VM and access the data

Login with the password "zeusdp" as zeus:

```
someuser@somempppc:~$  ssh localhost -l zeus -X -p 10006 -o StrictHostKeyChecking=no -o
    UserKnownHostsFile=/dev/null
Warning: Permanently added '[localhost]:10006' (RSA) to the list of known hosts.
zeus@localhost's password:
/usr/bin/xauth:  creating new authority file /home/zeus/.Xauthority
[zeus@mpp-dphep-CentOS-6 ~]$
```

Setup the environment and provide a Grid proxy copied from elsewhere:

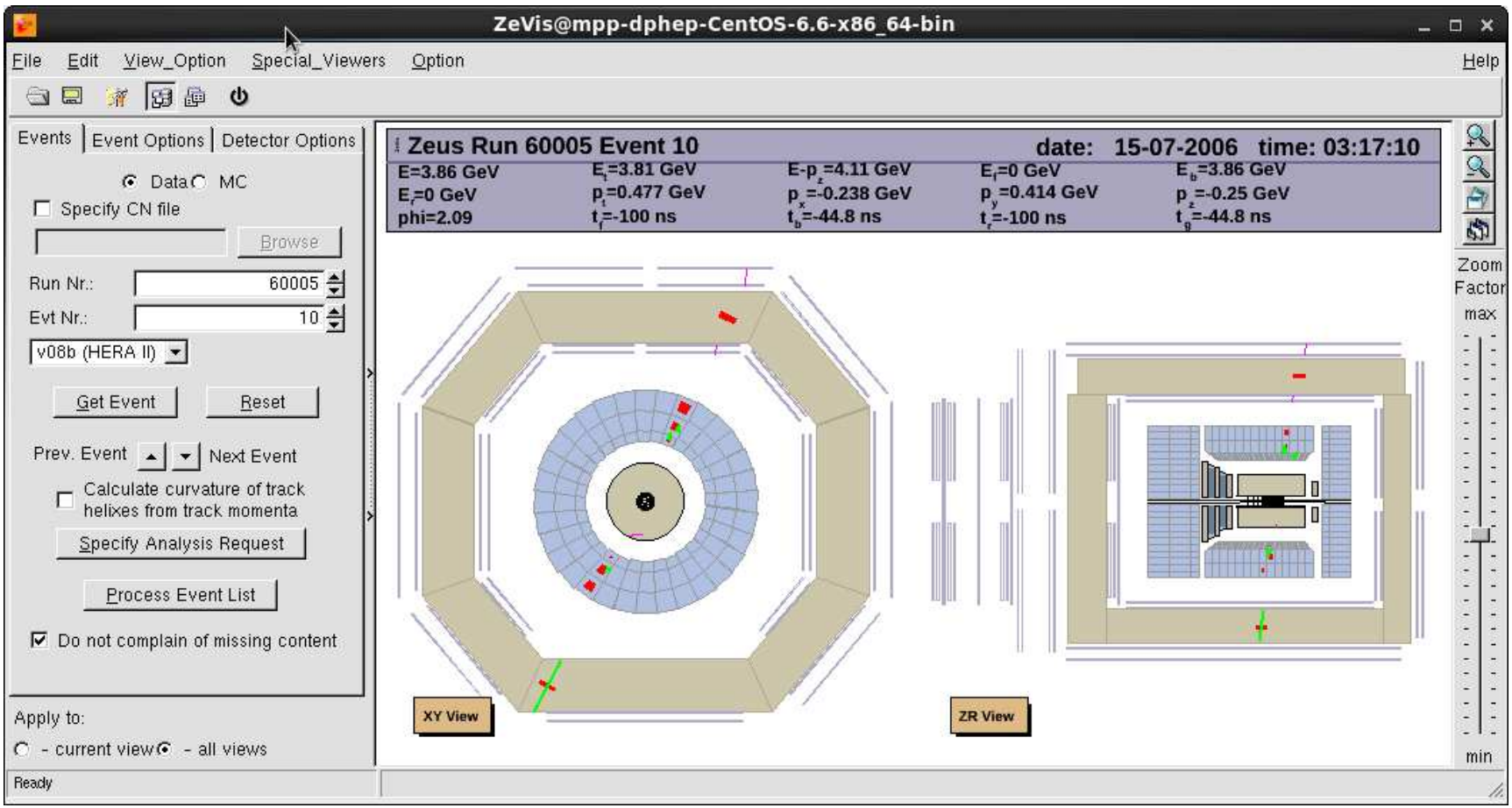
```
[zeus@mpp-dphep-CentOS-6 ~]$ . /usr/bin/ZEUS-RZG-env.sh
[zeus@mpp-dphep-CentOS-6 ~]$ export  X509_USER_PROXY=./mygridproxy.txt
```

Run event display:

```
[zeus@mpp-dphep-CentOS-6 ~]$ zevis
Variable ZARCH_TYPE is not set, looking for zevis.exe in the same directory
Using ZEVISDIR=/usr/share/zevis
Using CNDATAPREFIX=gsidcap://grid-srm.rzg.mpg.de:22128/pnfs/rzg.mpg.de/data/zeus/
Opening gsidcap://grid-srm.rzg.mpg.de:22128/pnfs/rzg.mpg.de/data/zeus/z/ntup/06p/v08b/data
    /root/data_06p_60005_60010_01.root
Opening gsidcap://grid-srm.rzg.mpg.de:22128/pnfs/rzg.mpg.de/data/zeus/z/ntup/06p/v08b/data
    /root/data_06p_60005_60010_01.root
Error in <TStreamerInfo::Build>: TPolyLine3D, discarding: Float_t* fP, illegal [3*fN]
    Array of 3-D coordinates  (x,y,z)

Error in <TStreamerInfo::Build>: ZPolygon2D, discarding: Float_t* fP, illegal [fN*3]
    Points in world coordinates
```

Event display



Write an analysis in ROOT using filecatalog

For example, plot $M(D^0\pi^+) - M(D^0)$ for D^{*+} candidates:

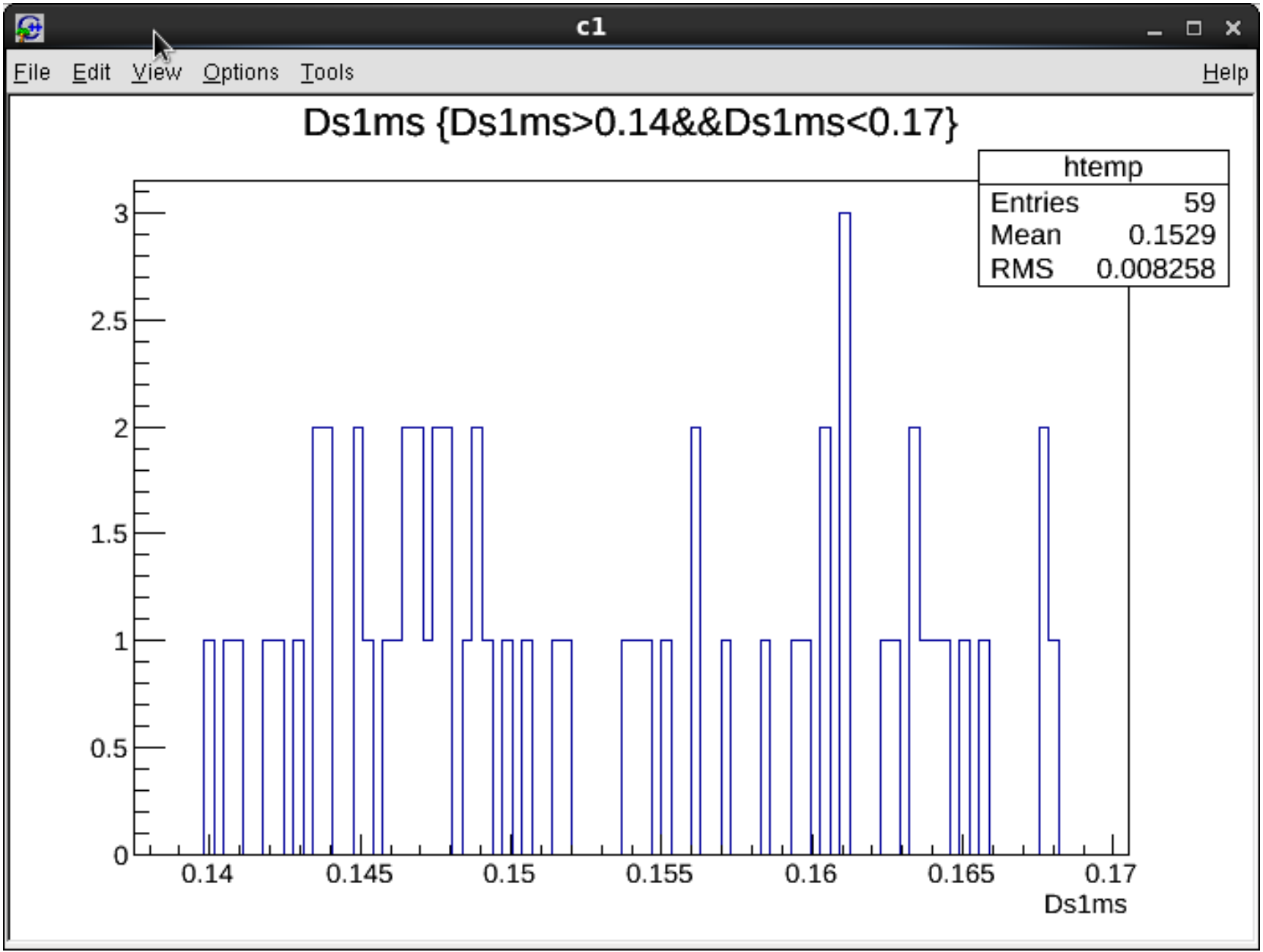
```
[zeus@mpp-dphep-CentOS-6 ~]$ vi main.c
[zeus@mpp-dphep-CentOS-6 ~]$ cat main.c
#include <string>
#include <vector>
#include "TChain.h"
#include "TApplication.h"
#include "TCanvas.h"
#ifdef __CINT__
gSystem->Load("/usr/lib64/libCNINFO.so");
int main(){
#else
#include "CNINFO.h"
int main(int argc, char **argv){
    TApplication* theApp = new TApplication  ("tapp", &argc, argv);
#endif
    CNINFO *A= new CNINFO();
    std::vector<std::string> test=A->GetFilesList("v08b","07p");
    TChain* C= new TChain("orange");
    for (int i=0; i<5; i++)              C->Add(test.at(i).c_str());
    TCanvas* T= new TCanvas();
    T->cd();
    C->Draw("Ds1ms","Ds1ms>0.14&&Ds1ms<0.17","",1000,1000);
    T->SaveAs("Ds1ms.pdf");
#ifndef __CINT__
    theApp->Run();
#endif
    return 0;
}
[zeus@mpp-dphep-CentOS-6 ~]$
```

Write an analysis in ROOT using filecatalog

Compile and run:

```
[zeus@mpp-dphep-CentOS-6 ~]$  g++ $(root-config --glibs --cflags) $(pkg-config --libs --cflags  CNINFO) main.c -o main
[zeus@mpp-dphep-CentOS-6 ~]$ ./main
Using CNDATAPREFIX=gsidcap://grid-srm.rzg.mpg.de:22128/pnfs/rzg.mpg.de/data/zeus/
Using CNSQLITEDB=/usr/share/CNINFO/CNINFO_sqlite3.db
This is CNINFO class, designed to read ZEUS Common Ntuples data base. It has the following functions:
    CNINFO();
    The class reads  CNDATAPREFIX and CNSQLITEDB environment variables or pick ups the default values.
    CNINFO(std::string CNDATAPREFIX, std::string CNDB );
    The class gets CNDATAPREFIX and CNSQLITEDB as arguments, pick ups environment variables or pick ups the default values.
    std::vector<std::string> GetFilesList(std::string version, std::string year, int run1=-1, int run2=-1);
    Returns a list of files.
    void InsertRequest(std::string line);
    Creates a nickname in the database for a data sample, the line should be in format
    user_version=v08b|year=mer_fl_only|request=v08b_v2012a.Data.MER_fl_only_fl_only_mer|type=data|
    void InsertFile(std::string line,std::string request);
    Inserts an entry to the database, the entry is in format
    fname=/acs/z/ntup/07p/v08b/mc/ntp/zeusmc.hfix627.f10688.grape_ela_2pt_0_5k12_15.ntp.gz|dtype=mc|ftype=paw|events_expected=50000|lumi=0|size=126063847|date=2014-01-22_21:01:37|run1=0|run2=0|events_processed=50000|sample=BH_el_PHP_0607p|version=v08b|
    request is an arbitrary name.
    void DumpToText(std::string file);
select name  from files  where version='v08b_v2012a.Data.0607p_gr2_07p'  and ftype='root' order by name;
Info in <TCanvas::Print>: pdf file Ds1ms.pdf has been created
```

$D^{*\pm}$ mass peak



ZEUS-DP@MPP TODO

- Provide more examples on analysis in DP mode;
- Provide an example for MC generation and cninfo data base update;
- Make virtual machines run on cloud services, e.g. CERN OpenStack.

H1-DP@MPP

Bits for H1@MPP

H1 data to preserve raw and processed data and corresponding logs.
Most of H1 data is copied to RZG (the decision on the datasets to preserve was made in May 2015):

- 300Tb
- 0.6Mio files

The anticipated end of copying date is mid. June 2015

Other aspects of H1@MPP

The others aspects of H1@MPP preservation are similar to those of ZEUS, but:

- The H1 software is going to be recompiled the deployment will be more complicated than for the ZEUS software.

Problems and what is possible to learn from them

Software preservation problems

With the chosen model of data preservation the most complicated task is to deploy the software in an isolated environment of virtual machines.

- Hard-coded paths in software packages;
- Absence of configuration, compilation and deployment scripts;
- Absence of control version systems and dependency tracking between packages;
- Absence of source code(!);
- Strong dependence on environment: OS type, compilers, system libraries;
- Reconstruction packages require many versions of huge executables;
- Using non-standard self-made tools even if a well-established standard alternative exists.

What we can learn from software preservation problems

- HEP software rarely packaged in a standard way, e.g. no .rpm or .deb packages, "configure" scripts, etc. Standard ways of installation would improve portability;
- Smaller pieces of software with VCS and well-defined dependencies are easier to install;
- Well-established formats for the data bases and other files help to avoid problems;
- The most standard software produces the smallest amount of problems.

Bits preservation problems

With the chosen model of the bit preservation the most complicated task is to preserve directory structure.

- Huge amount of data, order of hundreds of terabytes;
- Many-many-many files: $10^6 - 10^7$;
- Very non-homogeneous set of files: different types and sizes, different handling;
- No fast and standard tools to synchronise big amounts of data and directory structures;
- Low I/O rate on vitalised hosts;
- No multiprotocol tools;
- Lack of some useful features in many good tools: creation of directories, overwrite options, recursive copy and removal;
- Security policy limitations and restrictions.

What we can learn from bit preservation problems

- No single tool provides enough functionality for a simple copying. Possible solution would be to introduce the lacking functionality to dccp and uberftp. gridftp-enabled rsync would be a tool of a dream;
- Sometimes intensive I/O comes with intensive CPU load. A problem for VM hosts;
- Trivia: supporting more protocols is better. Good example is RZG;
- Trivia: less files is better. Even simple writing of log files to the ROOT file would reduce the number of files by 30%.

Conclusions

- The data preservation project at MPP is moving forward.
- A virtual machine for ZEUS was prepared; it is a template for other experiments: OPAL and potentially H1.

- DP for the active and future experiments will require more standardisation in the approach, tools, software and documentation.