Monte Carlo production for the CLIC CDR

Our experience

S. Poss, *C.B. Lam, J. Strube, C. Grefe*

CERN

September 27, 2011

Table of contents

Introduction

- 6 benchmark analysis:
  - $e^+e^- \to h\nu_e\overline{\nu}_e$,
  - $e^+e^- \to H^+H^-$, $e^+e^- \to H^0A$,
  - $e^+e^- \to \widetilde{q}_R\overline{\widetilde{q}}_R$,
  - $e^+e^- \to \widetilde{\ell}\overline{\widetilde{\ell}}\,(\ell = e, \mu)$,
  - $e^+e^- \to \widetilde{\chi}^{\pm}{}_i\widetilde{\chi}^{\mp}{}_j$, $e^+e^- \to \widetilde{\chi}^0{}_i\widetilde{\chi}^0{}_j$,
  - $e^+e^- \to t\bar{t}$ (500 GeV).
- Plus all the backgrounds (Standard Model)
- 2 detector models

Number of events processed in the last 9 months: $17 \cdot 10^6 \times 3$ processed.

Part I

Framework

Production framework

DIRAC has a Transformation System:

- Create a work flow object (XML representation of a job), and let the system create the jobs for you
- Productions are identified with a unique ID
- Automatic resubmission of failed jobs: minor monitoring needed

It saved us a huge amount of time and man power.

Production framework

The DIRAC transformation system comes with a useful set of API commands: get properties, change them, update inputs, etc.

Full interface to the ILD and SiD software frameworks[1], and to WHIZARD and PYTHIA code. Additionally we have an interface for generator level cuts (StdhepCut tool from Lars Weuste), Overlay input, etc.

[1] Look at http://lcd-data.web.cern.ch/lcd-data/doc/ilcdiracdoc/ILCDIRAC.Interfaces.API.NewInterface.Applications-module.html

Defining steps

- Decided to split all steps in different jobs, implying specific productions for:
  - Generation: WHIZARD/PYTHIA.
  - Simulation: Mokka/SLIC.
  - Reconstruction: Marlin/LCSIM-SLICPandora-LCSIM.
- Because of long simulation and reconstruction CPU time, few number of events per step ($10 - 50$) are required $\Rightarrow$ small files to manage.
- Chaining of productions is straightforward, more on that later: data driven production management.

All files produced are stored in the DIRAC replica and Metadata catalog

Replica catalog

- DIRAC provides a catalog: fast and efficient
- Has the possibility to write and read in the Lcg FC via the same interface.

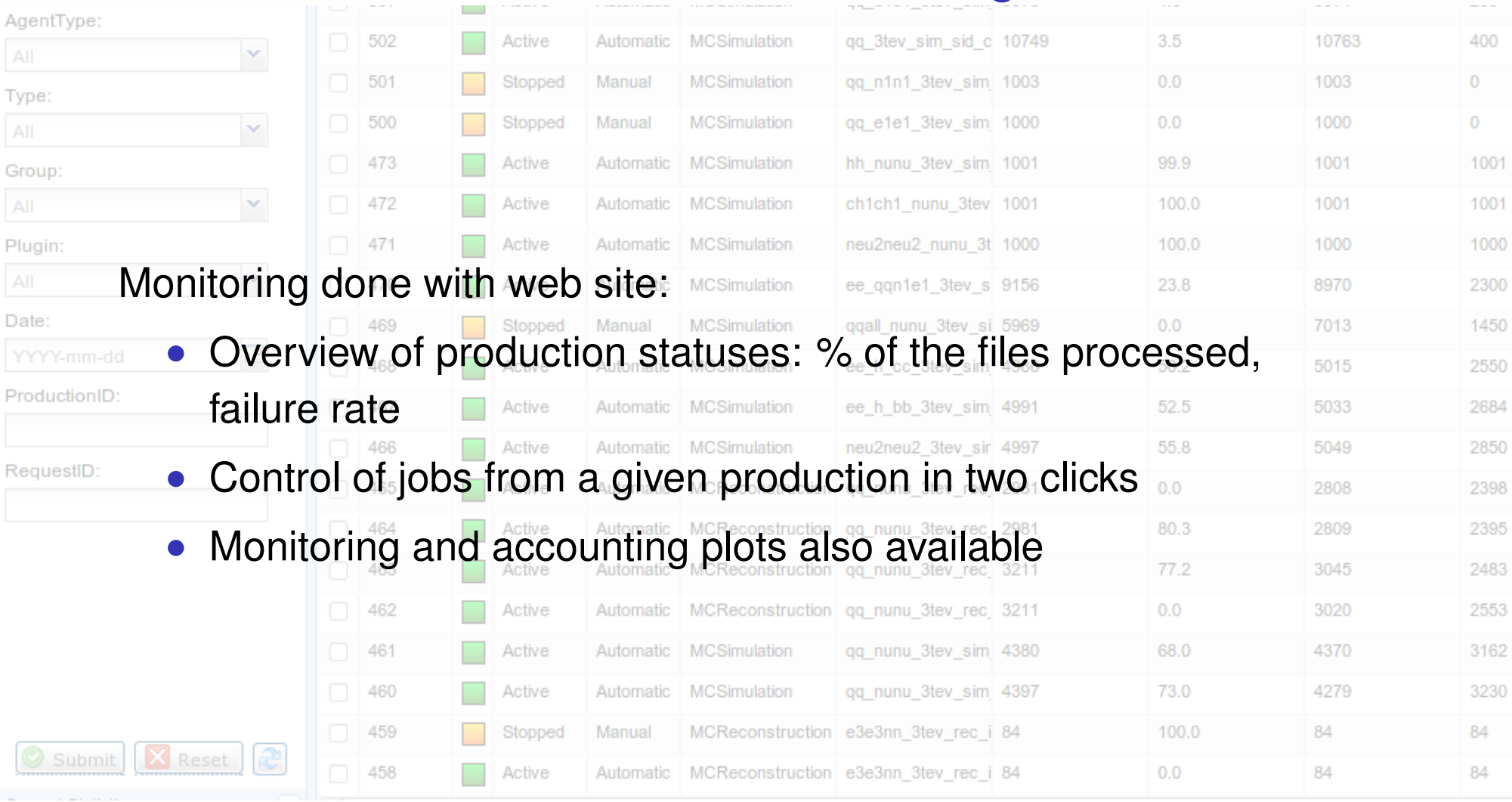
In the futur (coming month), when adding a file in the DIRAC FC, it will also enter in the ILC LFC.

Metadata catalog

- ILC stores the metadata encoded in the file names.
- Somehow restricts the number of meta info that one can add to a given file once the file is created.
- DIRAC FC comes with a metadata catalogue, that sets the metadata at directory level (e.g. production ID, software packages, cross section, polarisation, etc.) and at File level (number of events, etc.).
- Flexible as one can add (and remove) metadata on the fly.
- Searchable: find EvtType=tt Datatype=SIM
- Possibility to set ACLs.
- Also comes with convenient API commands.

Web interface is under development, should be available for tests in the coming weeks.

Production Monitoring



Monitoring done with web site:

- Overview of production statuses: % of the files processed, failure rate
- Control of jobs from a given production in two clicks
- Monitoring and accounting plots also available

Part II

CLIC Production

Defining a production

1. Define the input data (if any) via a metadata query (e.g. meta['ProdID']=186)
2. Define the application and versions you want to use
3. Define the output files and output storage

Data driven procedure: if new files are added to the catalogue that correspond to the metadata query, jobs processing those files are automatically created.

Generating Events

Use WHIZARD 1.95 (PYTHIA 6.4) for most samples:

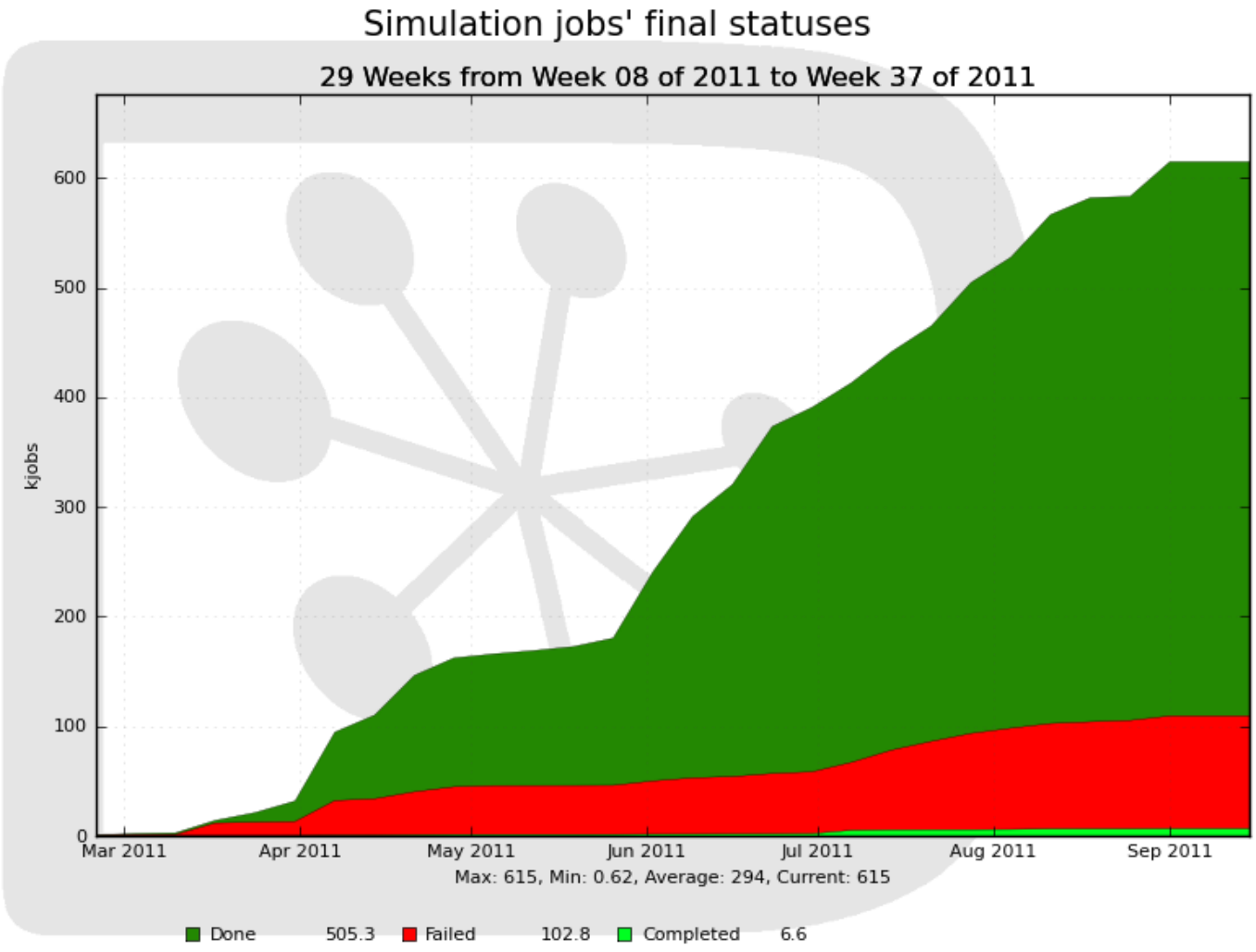
- Handling of beam spectrum and ISR included
- Adding a new process is straight forward
- Does the correct computation of diagrams and interference: accurate cross section and phase space
- Limitation: final states have no width, cannot do $t\bar{t}$, W^+W^- or Z^0Z^0.

Use PYTHIA 6.4 for those 3 samples:

- Final states have the correct width
- Less accurate estimation of cross section and phase space, but compatible to those produced by WHIZARD
- Has beam spectrum interface via CALYPSO, ISR is PYTHIA default
- Limitation: less flexibility, adding a channel cannot be done easily

Simulation

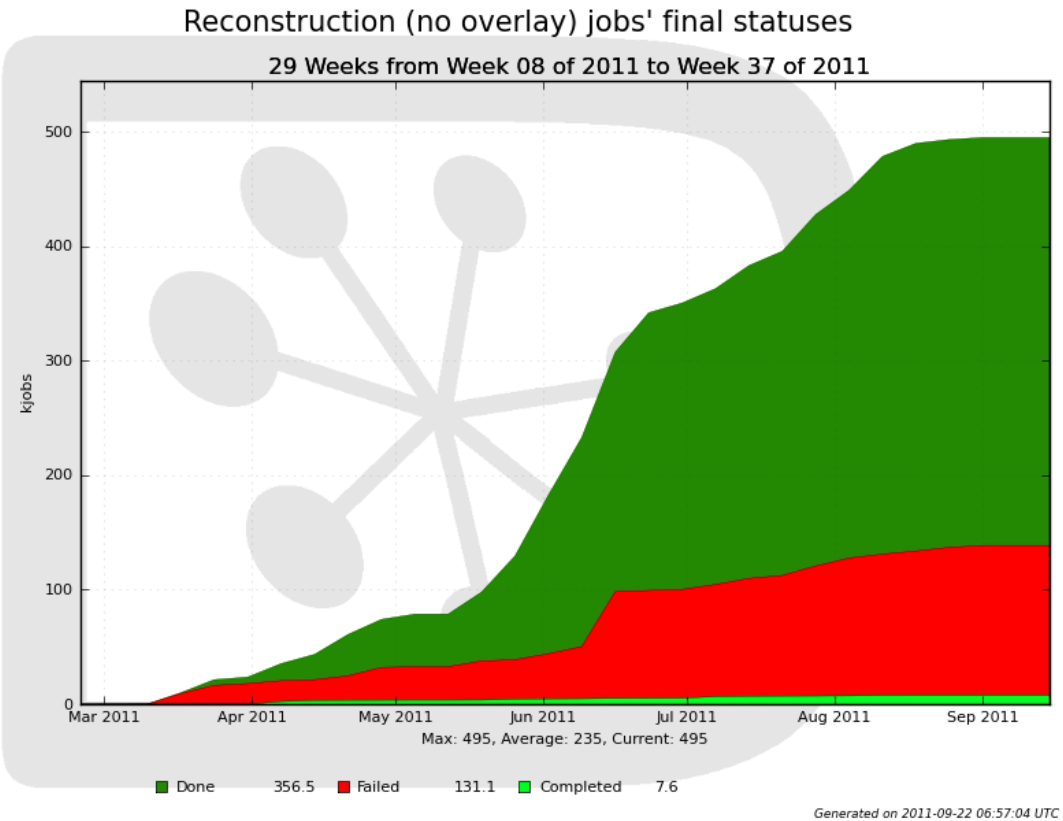
Define productions for both detector concepts, but very similar interface. Not an issue in the production framework:



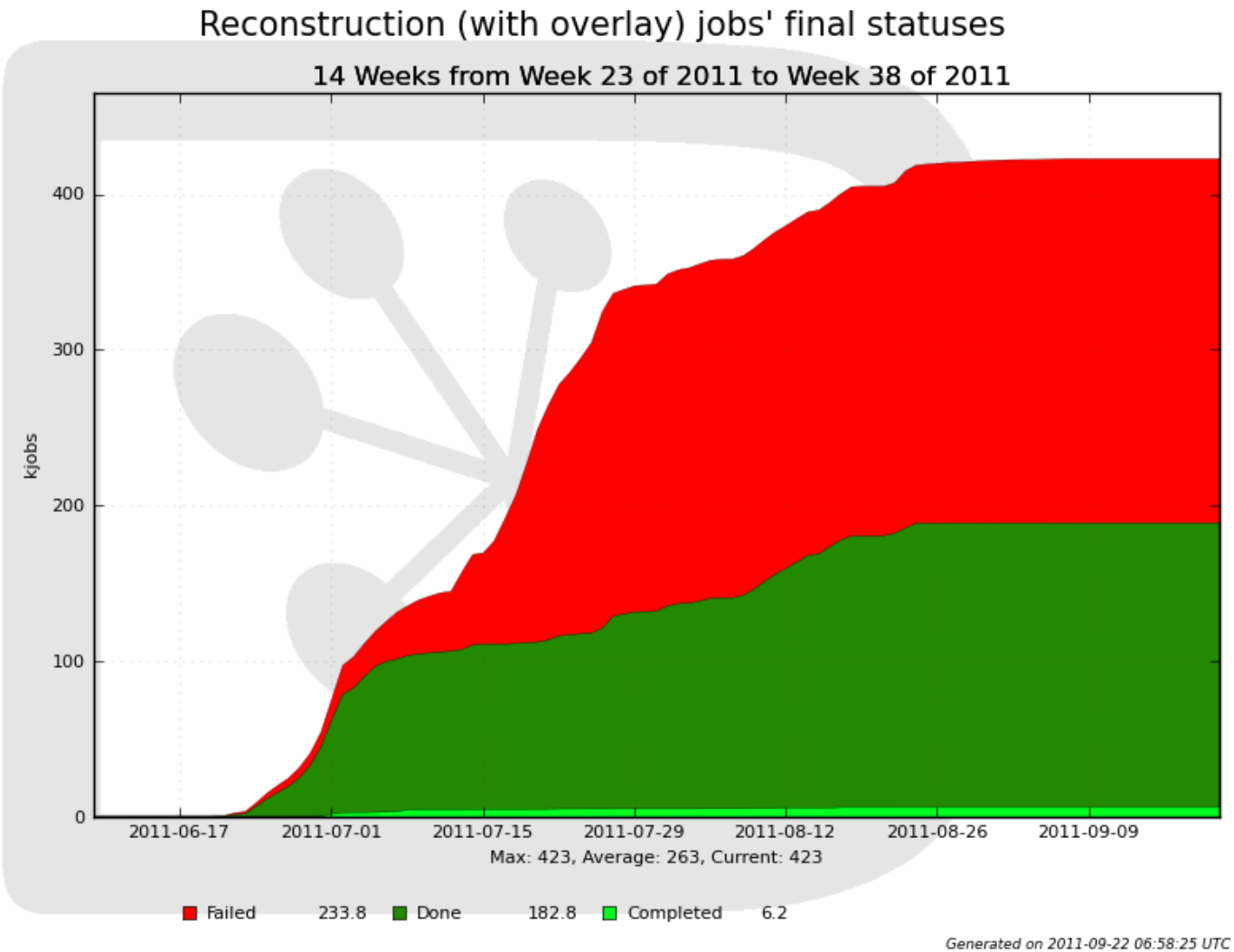
Reconstruction

As for the simulation: dedicated productions are created, automatically using files produced by the simulation step.

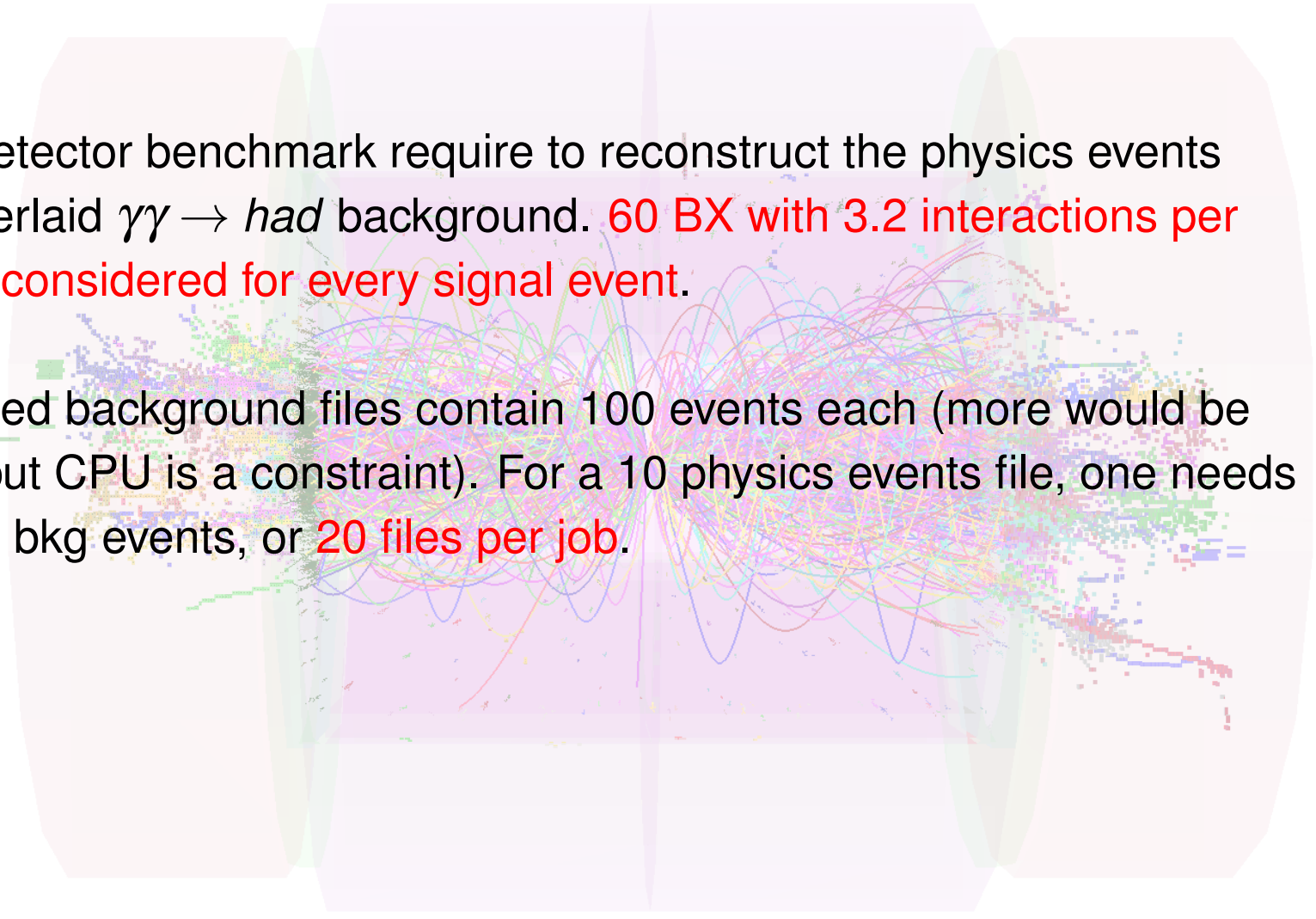
Without Overlay:



Reconstruction with overlay



Overlay handling (1)



CLIC detector benchmark require to reconstruct the physics events with overlaid $\gamma\gamma \rightarrow had$ background. 60 BX with 3.2 interactions per BX are considered for every signal event.

- Simulated background files contain 100 events each (more would be better but CPU is a constraint). For a 10 physics events file, one needs $\sim$ 2000 bkg events, or 20 files per job.

Overlay handling (2)

This did not work: storages (CERN in particular, but also IN2P3) do not cope with such a load (2000 jobs trying to access 20 random files each represent a huge number of queries).
Solution: merge the files randomly once to reduce the number of files needed: merge them by groups of 200 files.

- Works but requires manual intervention, and one needs a very large number of files to have enough combinations.

Needed solution: use random access (LCIO v1.51 has it, but not LCSIM). Then only one big file is needed on every site with direct access. No more transfers are needed.

Part III

Prospects

Prospects

Now running in the US sites: need to see for disk space. CERN disk requests will be made: for the moment, only 40Tb are available on disk (all used, but as mush as needed on TAPE though).

Will finish implementing dedicated service that will know the available processes, to ease users' lives. For the moment, a text file holds all the processes available and the corresponding WHIZARD version.

Will push to get the File Catalog web browser.

Part IV

Conclusion

Conclusion

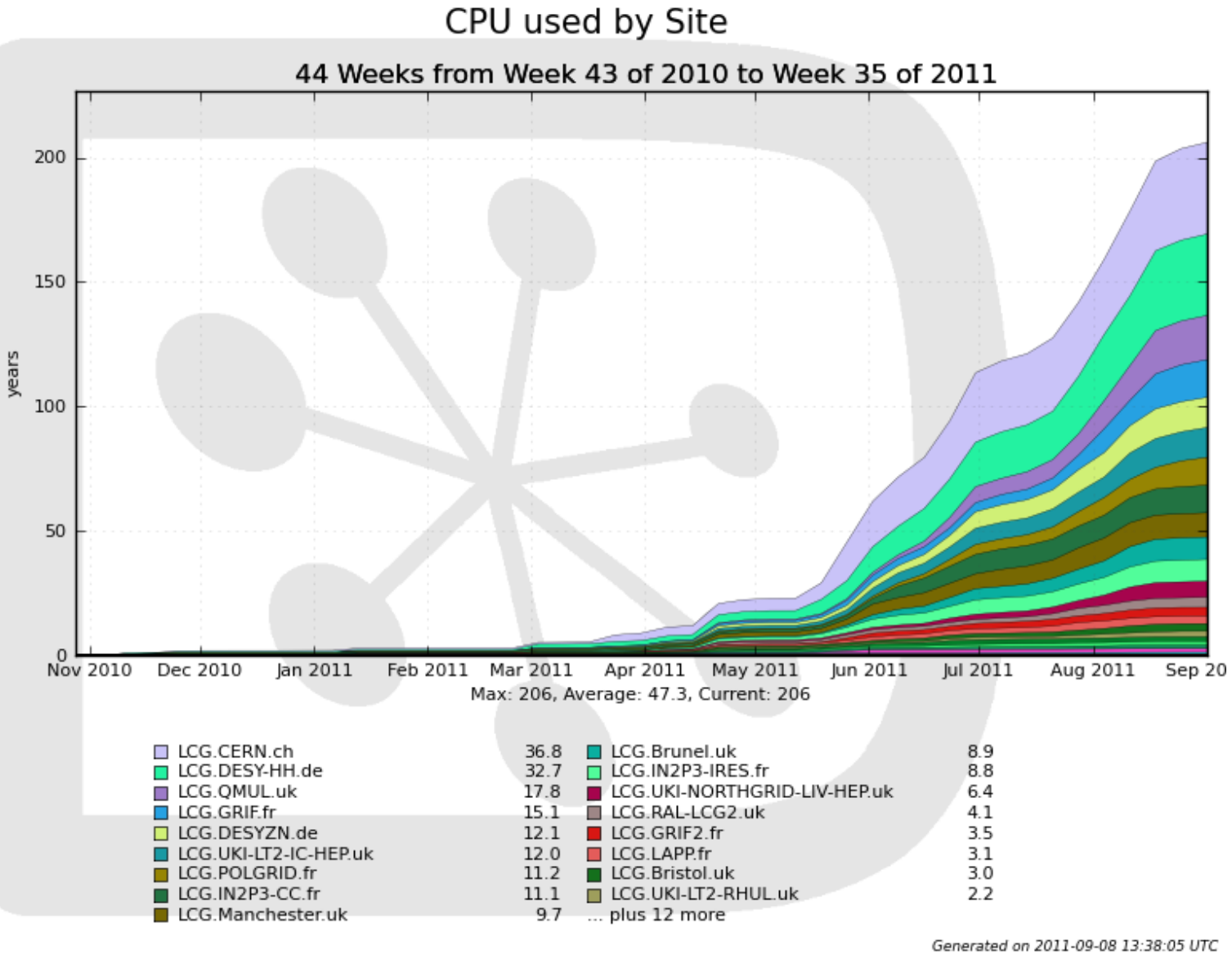
- $\approx$ 60 million events processed in 9 months
- Used $2 \cdot 10^{10}$ KSI2K seconds of CPU in $\sim$ 20 sites
- Data produced correspond to $\sim$ 130 TB
- Most issues were due to storage access (overlay in particular), not to software

Overall worked well: CDR (physics & detectors) is out!
https://edms.cern.ch/document/1160419

Part VI

Backup slides

Resources used: CPU



Resources used: Storage

Site	Production (TB)	User (TB)
CERN	128	20
IN2P3	4	9
RAL	4	28
KEK	0.02	0
IMPERIAL	1.6	0
TAU	$4 \cdot 10^{-4}$	0

Sites that can be used in addition: DESY, Bristol, BONN, RALPP

Technical aspects

Dropped pilot mode as some sites did not support it, using private pilot mode

Steering files are installed on the sites like applications (dependency relation), so no need to pass them in the input sandbox of the job.

Software installed in the Shared areas of the sites when/where possible. Does not use SAM framework (yet) as DIRAC takes care of the software installation and removal.

Services availability

60 million events correspond to $\sim$ 4 million jobs. On average 1000 concurrently running.

File catalog is the most expensive service in terms of CPU and simultaneous queries to DB. Had to duplicate service (not DB).

It would be useful to replicate the DB and other services to have more instances available (more VO boxes needed).

Same problem with JobManager with high load.

File names conventions

- Base path is /ilc/prod to separate from /ilc/user data.
- Path holds most relevant info (machine, energy, process, detector, data type (gen, SIM, REC, DST), prodID)
- File name is process_prodID_jobID.ext (ext=stdhep,slcio)

Data validation

When simulating data with 2 frameworks, it's necessary to validate it before running. For that purpose TOMATO was designed: convert the stdhep to slcio (stdhepjob supported by DIRAC), and run TOMATO (Marlin processor) to create histograms of "significant" distributions.

"Significant" is specific to the different analysis, so every working group needs to provide the variables to be plotted.

Was used for the $t\bar{t}$ analysis.