Event generation for CLIC

WHIZARD and PYTHIA

S. Poss

CERN, Switzerland

May 24, 2011

Outline

Part I

Introduction

Introduction

Conceptual Design Report: 6 channels considered to show detectors are able to deal with CLIC environment.
This talk:

- Generation process discussed
- Pros and cons of the chosen solutions, WHIZARD and PYTHIA

Part II

Generation for CLIC detectors

Generating events for CLIC detectors

Benchmark processes considered:

- $e^+e^- \to h\nu_e\bar{\nu}_e$
- $e^+e^- \to H^+H^-$, $e^+e^- \to H^0A^0$
- $e^+e^- \to \tilde{q}_R\bar{\tilde{q}}_R$
- $e^+e^- \to \tilde{\ell}\bar{\tilde{\ell}}$
- $e^+e^- \to \tilde{\chi}_i^+\tilde{\chi}_j^-$, $e^+e^- \to \tilde{\chi}_i^0\tilde{\chi}_j^0$
- $e^+e^- \to t\bar{t}$ at 500 GeV

Not listed here: background samples needed, among which $e^+e^- \to q\bar{q}$, $e^+e^- \to W^+W^-$ or even $e^+e^- \to W^+W^-Z^0$.

Total number of events needed for the CDR: 3.6 million.

Global framework used for production: same tools for generation, simulation, reconstruction: ILCDIRAC for job submission to the GRID.

Generating events for CLIC detectors

Large number of channels considered (more than 30), need proper tools:

- ILC decided to use WHIZARD (see next slides), we are grateful to benefit from the experience
- We added PYTHIA for some channels (see later)

Constrain:

- Simulation times (and reconstruction times when overlaying $\gamma\gamma \to \mathrm{Hadrons}$ events) become very large at 3 TeV: need small input samples. E.g. $\mathrm{e^+e^-} \to \mathrm{t\bar{t}}$: 10 events per job.

$\Rightarrow$ At generator level, produce many small files. Run several times the generator per CPU slot (reduce the risk of failure per execution).

Part III

WHIZARD

WHIZARD

Multi-purpose generator dedicated to linear colliders.

> *Designed for efficient calculation of multiple scattering cross sections and simulated event samples.*

Developed by Wolfgang Kilian, Thorsten Ohl, Jürgen Reuter, and Christian Speckner.

Version used: v1.95 modified by Tim Barklow for ILC case handling. Cannot use v2 because no beam spectrum handling yet.

Akiya Miyamoto and Mikael Berggren use it also for ILC DBD generation. Very good collaboration – THANKS!

WHIZARD v1.95: Pros (1)

- Defining a process is as simple as

```
   ww_n1n1      e1,E1    W+,W-,n1,N1    omega
ddcse1n1_o      e1,E1    d,D,c,S,e1,N1  omega
```

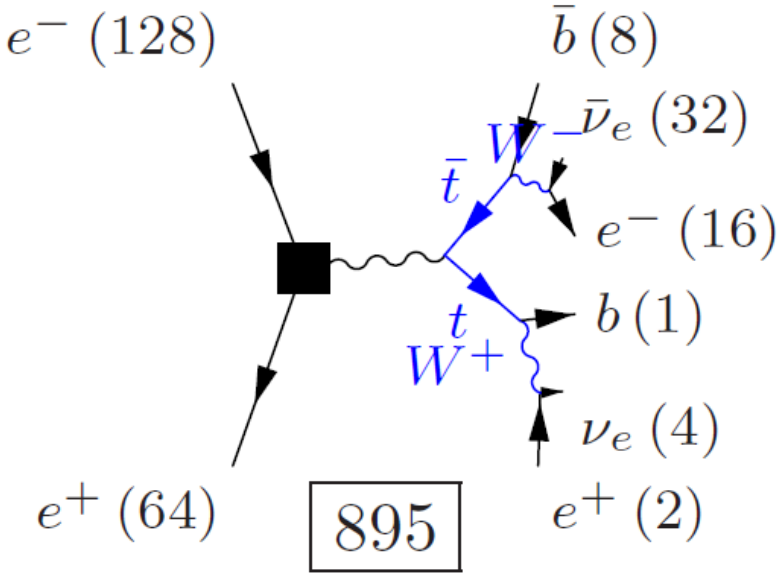
- Supports aliases: alias q u:d:s:c:b, alias Q U:D:S:C:B and define

```
qq    e1,E1    q,Q    omega
```

- Simplifies process definitions
- Provides diagrams illustrations to be run with LaTeX

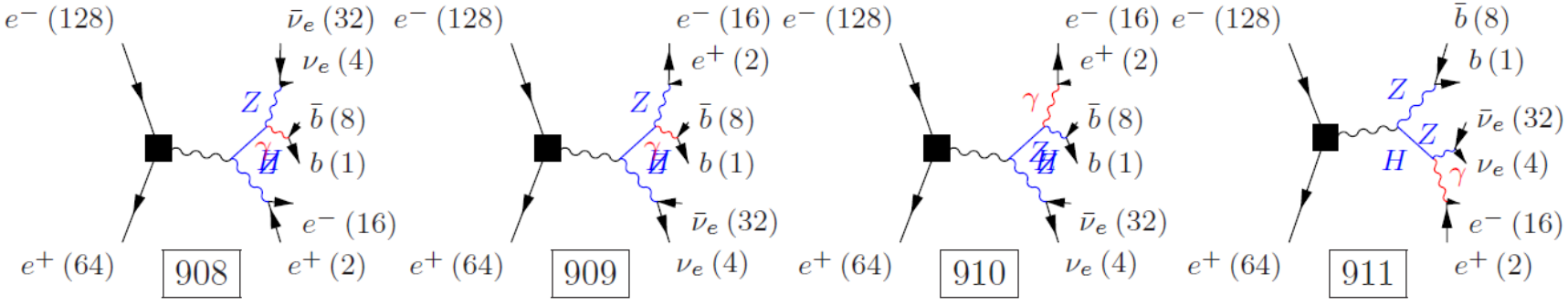
WHIZARD v1.95: Pros (2)

- Computes the cross sections taking into account all possible intermediate diagrams: can lead to surprises.
  Example : benben e1,E1 b,E1,n1,B,e1,N1 omega



WHIZARD v1.95: Pros (3)

But it also corresponds to:



plus $\approx$ 100 diagrams, in total only 1 contains the $t\bar{t}$ pair that we want.
Generated sample for that channel contains 32% of $e^+e^- \to t\bar{t}$ decays.

WHIZARD v1.95: Pros (4)

- Able to generate a number of events corresponding to wanted luminosity

This was (and still is) used for the ILC production. We don't use it, for bookkeeping reasons: not all files produced have the same number of events.

WHIZARD v1.95: Cons (1)

- The final states provided by WHIZARD have no width: not possible to generate directly $e^+e^- \to t\bar{t}$, $e^+e^- \to W^+W^-$, $e^+e^- \to Z^0Z^0$, $e^+e^- \to W^+W^-Z^0$, $e^+e^- \to Z^0Z^0Z^0$, $e^+e^- \to H\nu_e\overline{\nu}_e$, etc.

Possible solution: provide the exact final state

- for $t\bar{t}$, ask for bbqQqQ, WHIZARD to PYTHIA interface rebuilds the top pair,
- then integration time is very large: several days, cannot run on the GRID.

WHIZARD v1.95: Cons (2)

The following are my return on experience:

- Written in Fortran: I don't know that language (too young)
- Tedious to install: thanks to M. Berggren, convenient install script available.
- Requires dedicated beam spectrum interface: provided by Tim.
- Generator level cuts are not simple to setup. Some are forbidden (divergence during integration): use L. Weuste's tool "stdhepCut" to cut after generation $\rightarrow$ generate a very big sample and keep only the events passing the cuts.
- Cannot select diagrams to use, only visualization.

WHIZARD v1.95: Cons (2)

The following are my return on experience:

- Written in Fortran: I don't know that language (too young)
- Tedious to install: thanks to M. Berggren, convenient install script available.
- Requires dedicated beam spectrum interface: provided by Tim.
- Generator level cuts are not simple to setup. Some are forbidden (divergence during integration): use L. Weuste's tool "stdhepCut" to cut after generation $\rightarrow$ generate a very big sample and keep only the events passing the cuts.
- Cannot select diagrams to use, only visualization.

WHIZARD v1.95: Cons (2)

The following are my return on experience:

- Written in Fortran: I don't know that language (too young)
- Tedious to install: thanks to M. Berggren, convenient install script available.
- Requires dedicated beam spectrum interface: provided by Tim.
- Generator level cuts are not simple to setup. Some are forbidden (divergence during integration): use L. Weuste's tool "stdhepCut" to cut after generation $\rightarrow$ generate a very big sample and keep only the events passing the cuts.
- Cannot select diagrams to use, only visualization.

WHIZARD v1.95: Cons (2)

The following are my return on experience:

- Written in Fortran: I don't know that language (too young)
- Tedious to install: thanks to M. Berggren, convenient install script available.
- Requires dedicated beam spectrum interface: provided by Tim.
- Generator level cuts are not simple to setup. Some are forbidden (divergence during integration): use L. Weuste's tool "stdhepCut" to cut after generation $\rightarrow$ generate a very big sample and keep only the events passing the cuts.
- Cannot select diagrams to use, only visualization.

WHIZARD v1.95: Cons (2)

The following are my return on experience:

- Written in Fortran: I don't know that language (too young)
- Tedious to install: thanks to M. Berggren, convenient install script available.
- Requires dedicated beam spectrum interface: provided by Tim.
- Generator level cuts are not simple to setup. Some are forbidden (divergence during integration): use L. Weuste's tool "stdhepCut" to cut after generation $\rightarrow$ generate a very big sample and keep only the events passing the cuts.
- Cannot select diagrams to use, only visualization.

Part IV

PYTHIA

PYTHIA

For the channels that cannot be done using WHIZARD because of the zero width problem for final states involving e.g. Z^0, we use PYTHIA directly.

Code was provided by Marco Battaglia.

PYTHIA: Pros

- Program provided has a straight forward interface: defining the process is just setting MSEL (PYTHIA common block).
- Generates without any problems $t\bar{t}$,W^+W^-,Z^0Z^0, Higgs channels.
- Fast!
- Daniel Schulte provides Calypso: input the beam spectrum directly without the need for conversion.
- Generator level cuts are easier to apply, as they are directly implemented in the event loop.

PYTHIA: Pros

- Program provided has a straight forward interface: defining the process is just setting MSEL (PYTHIA common block).
- Generates without any problems $t\bar{t}$,W^+W^-,Z^0Z^0, Higgs channels.
- Fast!
- Daniel Schulte provides Calypso: input the beam spectrum directly without the need for conversion.
- Generator level cuts are easier to apply, as they are directly implemented in the event loop.

PYTHIA: Pros

- Program provided has a straight forward interface: defining the process is just setting MSEL (PYTHIA common block).
- Generates without any problems $t\bar{t}$,W^+W^-,Z^0Z^0, Higgs channels.
- Fast!
- Daniel Schulte provides Calypso: input the beam spectrum directly without the need for conversion.
- Generator level cuts are easier to apply, as they are directly implemented in the event loop.

PYTHIA: Pros

- Program provided has a straight forward interface: defining the process is just setting MSEL (PYTHIA common block).
- Generates without any problems $t\bar{t}$,W^+W^-,Z^0Z^0, Higgs channels.
- Fast!
- Daniel Schulte provides Calypso: input the beam spectrum directly without the need for conversion.
- Generator level cuts are easier to apply, as they are directly implemented in the event loop.

PYTHIA: Pros

- Program provided has a straight forward interface: defining the process is just setting MSEL (PYTHIA common block).
- Generates without any problems $t\bar{t}$,W^+W^-,Z^0Z^0, Higgs channels.
- Fast!
- Daniel Schulte provides Calypso: input the beam spectrum directly without the need for conversion.
- Generator level cuts are easier to apply, as they are directly implemented in the event loop.

PYTHIA: Cons

- Cannot produce $W^+W^-Z^0$, $Z^0Z^0Z^0$: not part of PYTHIA's process list. Need to use comphep: M. Battaglia will generate them.
- Written in Fortran.
- Tedious to install: compilation very sensitive to gfortran version, Calypso had to be modified to compile on 64bit machine.
- Not simple to configure on the fly: not possible to use configuration file (FFREAD routine fails on 64bits), rather use environment variables.
- Generator level cuts not configurable at run time.

Last 2 points not too much a problem as PYTHIA is used only for 3 channels.

PYTHIA: Cons

- Cannot produce $W^+W^-Z^0$, $Z^0Z^0Z^0$: not part of PYTHIA's process list. Need to use comphep: M. Battaglia will generate them.
- Written in Fortran.
- Tedious to install: compilation very sensitive to gfortran version, Calypso had to be modified to compile on 64bit machine.
- Not simple to configure on the fly: not possible to use configuration file (FFREAD routine fails on 64bits), rather use environment variables.
- Generator level cuts not configurable at run time.

Last 2 points not too much a problem as PYTHIA is used only for 3 channels.

PYTHIA: Cons

- Cannot produce $W^+W^-Z^0$, $Z^0Z^0Z^0$: not part of PYTHIA's process list. Need to use comphep: M. Battaglia will generate them.
- Written in Fortran.
- Tedious to install: compilation very sensitive to gfortran version, Calypso had to be modified to compile on 64bit machine.
- Not simple to configure on the fly: not possible to use configuration file (FFREAD routine fails on 64bits), rather use environment variables.
- Generator level cuts not configurable at run time.

Last 2 points not too much a problem as PYTHIA is used only for 3 channels.

PYTHIA: Cons

- Cannot produce $W^+W^-Z^0$, $Z^0Z^0Z^0$: not part of PYTHIA's process list. Need to use comphep: M. Battaglia will generate them.
- Written in Fortran.
- Tedious to install: compilation very sensitive to gfortran version, Calypso had to be modified to compile on 64bit machine.
- Not simple to configure on the fly: not possible to use configuration file (FFREAD routine fails on 64bits), rather use environment variables.
- Generator level cuts not configurable at run time.

Last 2 points not too much a problem as PYTHIA is used only for 3 channels.

PYTHIA: Cons

- Cannot produce $W^+W^-Z^0$, $Z^0Z^0Z^0$: not part of PYTHIA's process list. Need to use comphep: M. Battaglia will generate them.
- Written in Fortran.
- Tedious to install: compilation very sensitive to gfortran version, Calypso had to be modified to compile on 64bit machine.
- Not simple to configure on the fly: not possible to use configuration file (FFREAD routine fails on 64bits), rather use environment variables.
- Generator level cuts not configurable at run time.

Last 2 points not too much a problem as PYTHIA is used only for 3 channels.

PYTHIA: Cons

- Cannot produce $W^+W^-Z^0$, $Z^0Z^0Z^0$: not part of PYTHIA's process list. Need to use comphep: M. Battaglia will generate them.
- Written in Fortran.
- Tedious to install: compilation very sensitive to gfortran version, Calypso had to be modified to compile on 64bit machine.
- Not simple to configure on the fly: not possible to use configuration file (FFREAD routine fails on 64bits), rather use environment variables.
- Generator level cuts not configurable at run time.

Last 2 points not too much a problem as PYTHIA is used only for 3 channels.

Part V

Conclusions

Conclusion

- WHIZARD is a great tool for cross sections evaluations: does proper diagram interference computations
- Simple to define a decay process
- Some channels cannot be done ($t\bar{t}$) because final states have no width, and integration time very large (several days)
- PYTHIA is straight-forward
- Can generate the $t\bar{t}$, W^+W^-, and Z^0Z^0 with width
- Very fast
- Not configurable during runtime
- Cannot generate $W^+W^-Z^0$, $Z^0Z^0Z^0$

Combining both programs, most channels are covered and can be generated.