ILC Studies Using the Grid

Sometimes 127.0.0.1 is Just Not Enough

Adrian Vogel
DESY FLC

Background Sources at the ILC

e^+e^- pairs are a main source of background

- beams have to be focused very strongly (σ_y = 5 nm)
- beam-beam interaction creates beamstrahlung
- beamstrahlung photons scatter to e^+e^- (10^5 / BX)
- e^+e^- smash into forward calorimeters (BeamCal) and magnets of the beam delivery / extraction line
- lots of photons, neutrons, and charged particles

Other sources are supposed to be negligible (beam dump, synchrotron radiation,. . .) or have to be studied in further detail (beam halo, extraction line losses)

Problems with Background

Inner silicon trackers (VXD, SIT, FTD)

- hits from charged particles (direct / indirect)
- silicon bulk damage from neutron fluence

Main gaseous tracker (TPC)

- Compton scattering, photon conversion
- neutron-proton collisions (recoil) with hydrogen
- additional primary ionisation, field distortions

Calorimeters (ECAL, HCAL)

- more photons from nuclear reactions, neutron capture
- random low-energy hits, radiation damage (?)

Simulation Tools – Guinea Pig

Input

- set of beam parameters (E, $\vec{\sigma}$, $\vec{\beta}$, Q, . . .)

Output

- particles in the disrupted beams
- beamstrahlung photons
- e^+e^- pair particles
- hadronic scattering products (“minijets”)

Existing simulation data

- TESLA beam parameters (500 GeV, 800 GeV)
- various ILC parameter sets (500 GeV, 1 TeV)

Simulation Tools – Mokka

Mokka is a full detector simulation

- based on the Geant 4 framework
- written in C++, modular design
- main development at LLR, France
- now: contributions from many different users
- successor of Brahms (GEANT3, Fortran)

Mokka uses LCIO as a persistency framework

- predefined storage classes (particle, track, hit,. . .)
- lightweight and robust, cross-platform design
- supported by large parts of the ILC community

Simulation Runs

Guinea-Pig is run on the local host

- simulated pairs from 100 BX for various parameter sets
- uploaded output (in chunks) to Storage Elements
  (can be found at `/grid/ilc/vogel/pairs`)

Mokka is run on the Grid

- shell script takes control on the worker node:
- downloads Mokka, MySQL, G4 data, and input files
- spawns a MySQL server and runs Mokka
- uploads the simulation output to SEs

Data extraction with Marlin runs locally again

Pitfalls – Data Storage

Input data may be unavailable

- SEs may not respond, waiting doesn't always help
- keep replicas on several sites, choose one

Output data must go somewhere

- failure to store the output data is annoying
- always try several SEs for uploading
- ship data in the OutputSandbox as a fallback

Persistency is not 100% guaranteed (as always)

- better have a backup replica somewhere

Good news: LFC runs solid as a rock – fortunately!

Simulation Outcome

Simulated various settings (with 100 BX each, for now)

- geometries of the forward region (including B-fields)
- beam parameter sets (TESLA, Nominal, Low P)
- thickness of the absorber in front of the BeamCal

Full simulations are time-consuming

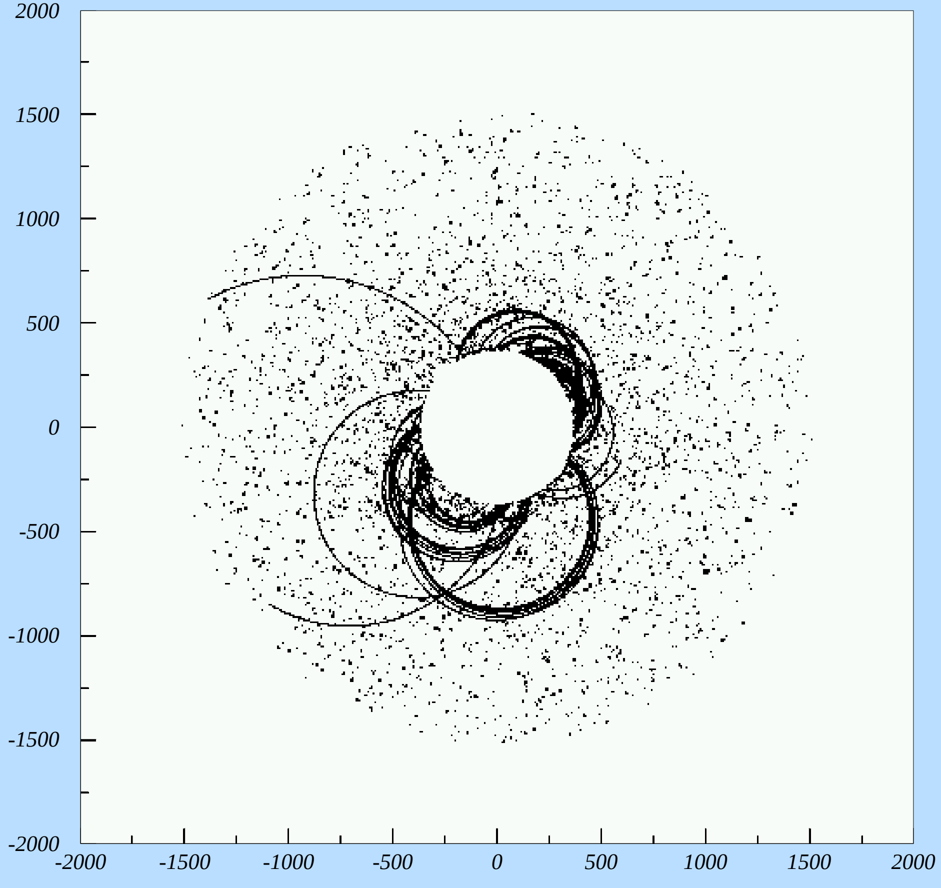
- each setting nedds approx. one half CPU-year
- takes about one to two days on the Grid

Simulated data is available on the Grid

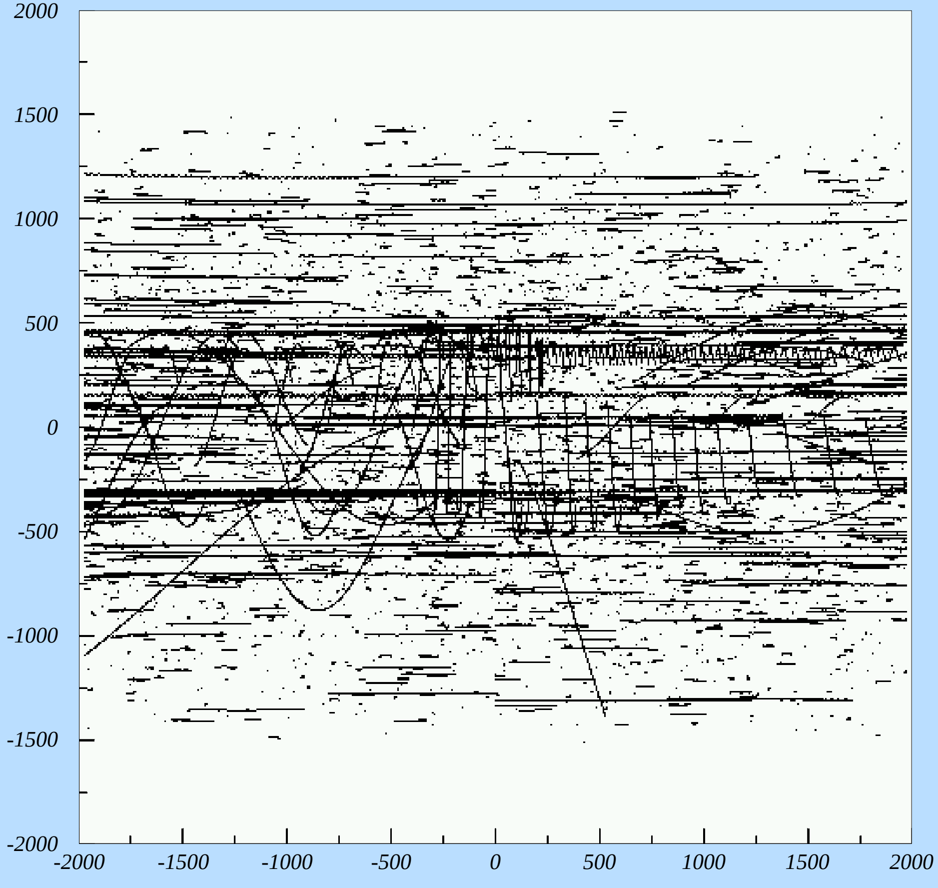
- approximately 5 GB of data per setting
- standard Mokka output and customised information

Example – TPC Hits

Mokka hits in the TPC (overlay of 100 BX)



Front view



Side view

Questions and Tasks

- Which design decisions affect the TPC, and how?
- Can we use a quencher which contains hydrogen?
- How large will the occupancy be at a given time? (with superposition of 160 bunch crossings)
- Provide a “background library” with ready-to-use events to be superimposed on “real” physics for analyses
- Set up a consistent software toolkit for the TPC: digitisation – tracking – reconstruction – analysis
- Will the background signals have an impact on pattern recognition, efficiencies, resolutions?