SiD Software and Computing: Status / Plans

Jan Strube
Tohoku University

TOHOKU
UNIVERSITY

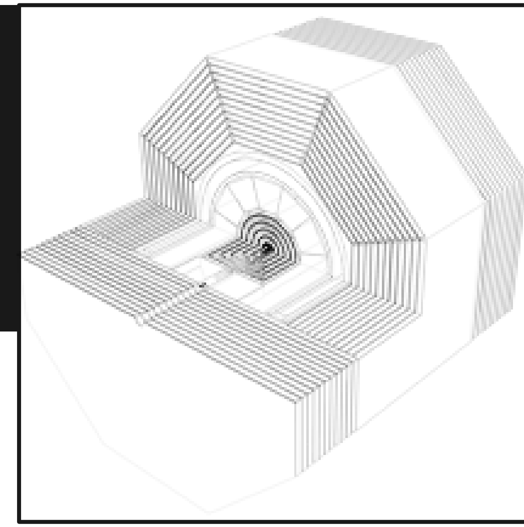
Thanks to Norman Graf (SLAC) for material

Software List / SimReco chain

- Event Generator -- Common Generators working group
- **Detector Simulation -- SLIC**
- **Digitization and Track reconstruction**
- **Event Reconstruction -- slicPandora**
- Vertex Reconstruction and Jet Flavor Tagging -- LCFIPlus
- Data Analysis -- any LCIO tool

Largely Common Software - shared by ILD/SiD
Analysis code works for both concepts

Simulation



SLIC - developed at SLAC

- Geometry interface to GEANT4
- Geometry definable at runtime
- Defines two xml formats
  - compact format readable by humans
  - expanded to lcdd - same format for use in simulation **and** reconstruction

Centerpiece for the DD4HEP project to unify detector description for simulation and reconstruction

Digitization and Pattern Recognition

- Digitization of charge depositions from simulation
- Pattern recognition with automated generation of strategies for seeding and extending tracks for different detector variants
- helical track fitter

Stable code base, with unit tests

http://www.slac.stanford.edu/~sinev/chronodoc/pixsim_help.html

Interlude - PixSim (Nick Sinev, UofO)

Specialized code for detailed studies of charge carrier movement in silicon sensors

- can read TCAD files
- study effects of doping, mobility, diffusion length, …
- study effects of electric and magnetic fields

Not run in default reconstruction chain

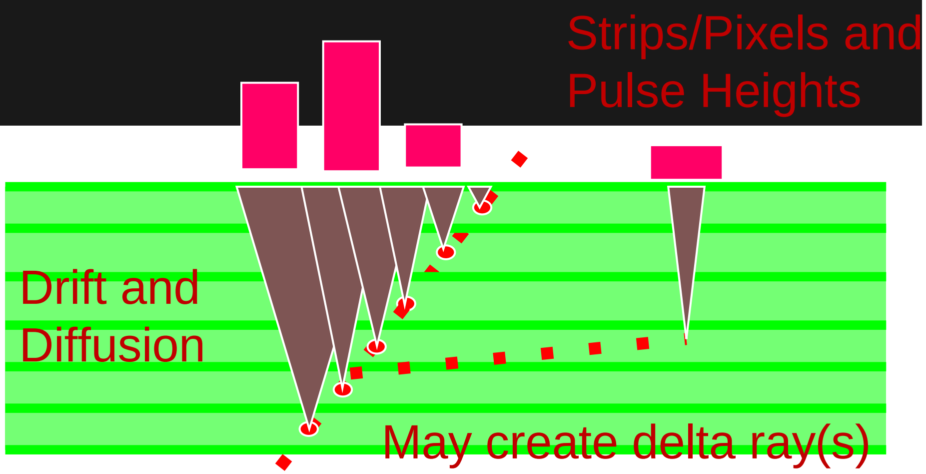
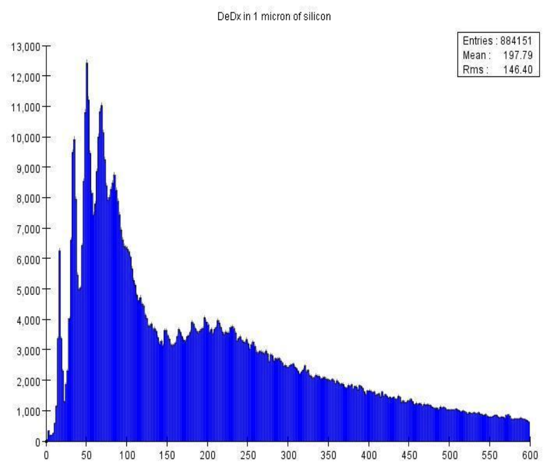
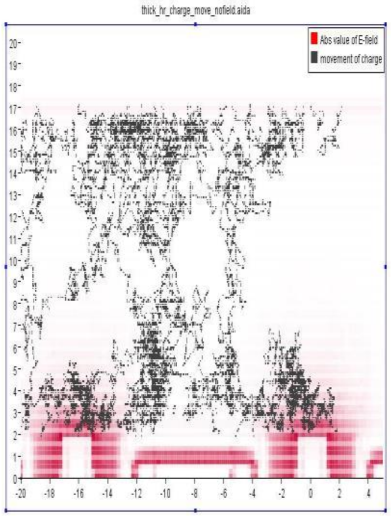


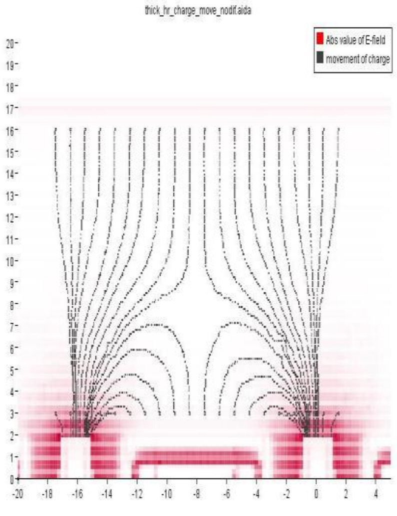
Illustration of the electric field effect on the charge collection in silicon



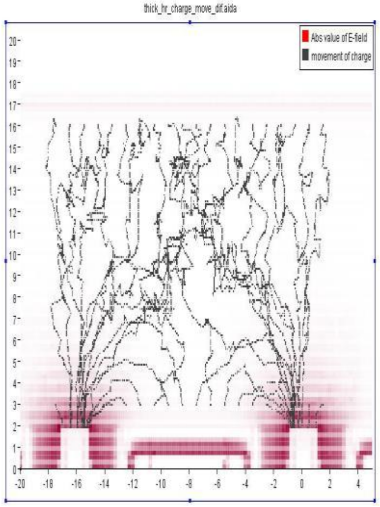
Example energy loss distribution for 1μm Si



only by diffusion



only by electric forces



combined

Event Reconstruction

- PandoraPFA used since DBD
- Almost out of the box
  - ScaleHotHadrons turned off for DHCAL
  - Worked reasonably well, even with DHCAL
  - Some work on tuning would be welcome

- Accessed with slicPandora wrapper to expose the SiD geometry to PandoraPFA API https://confluence.slac.stanford.edu/display/ilc/LCFIPlus

Vertex Finding / Flavor Tagging

- LCFIPlus used since DBD
- Vertex finding based on SLD algorithm ZVTOP
  - Uses all tracks in the event (not only in jet) to find vertices
- Comes with own jet finding that does not break up the vertices
- Same software as ILD, but needs dedicated tuning for each detector variant / analysis

Introduction to LCIO

Linear Collider Input / Output

- Two pieces:
  - Event Data Model
  - Default file format: SIO (Serial Input / Output)
- EDM is defined with Particle Flow in mind
- ReconstructedParticle object
  - Container of Tracks, Clusters, Vertices
  - For everything that fits a particle hypothesis
  - electrons, muons, V0, B mesons, jets, …
- (Currently still) largely detector agnostic
  - B_z is only external input flavor tagging

Software Summary

- ILD / SiD share a large part of the reconstruction software and LCIO
  - Allows analyses to compare results
  - Allows people with prior ILC exposure to get started quickly
- The way geometry is being handled in ILC detectors is changing
  - Expertise needs to be re-learned
  - There is never a bad time to join SiD. Now is a good time.

Opportunities For Getting Involved

- LCFIPlus (see also Marcel Demarteau’s talk)
  - currently two jet finding steps necessary
    - one to remove background, one for flavor tagging
    - vertex finding needs to be re-done after first step
  - vertex finding still basically same as SLD Detector
- Tracking
  - Kalman filter for detailed tracker layout studies
- PandoraPFA
  - studies of reconstruction performance in high-field, compact, highly granular calorimeter
  - studies of performance with digital calorimetry (ECAL / HCAL)

Computing

SiD Resources on the Grid

- Resources on the grid are shared within the ILC virtual organization (VO)
- VO members are (main sites)
  - ILD (DESY and KEK)
  - Clicdp (CERN / IN2P3)
  - SiD (PNNL, SLAC, everything else opportunistically)
- All three are currently running detector optimization campaigns
- Storage: 150 TB at RAL (full), 150 TB PNNL, 20 TB at SLAC
- CPU: ILC VO Total: ~10k CPU + PNNL

SiD on the Grid (during the DBD)

dedicated resources: PNNL, SLAC, CERN
temporary quota increase: FNAL, RAL Tier 1
opportunistic use: all others

Worldwide LHC Computing Grid (WLCG) resources have been established during LOI and CLIC CDR efforts

SiD takes advantage of the international computing grid infrastructure

Existing Samples

Stdhep (generator level):
https://confluence.slac.stanford.edu/display/ilc/Standard+Model+Data+Samples
On the grid:
https://confluence.slac.stanford.edu/pages/viewpage.action?pageId=138785074
https://confluence.slac.stanford.edu/display/ilc/DBD+Data+Samples
Currently: ~ 10 TB DST @ 1 TeV ⇒ 51 MEvents
~ 250 GB @ 500 GeV ⇒ 6.5 MEvents
~ 200 GB @ 250 GeV ⇒ 12 MEvents

ILCDirac Overview

- Submit scripts written in Python
  - Choose input files
  - reco steps mix and match
  - (semi-optionally) select sites for running
  - (semi-optionally) select site for output
- Web interface http://ilcdirac.cern.ch/DIRAC/
  - bookkeeping, restart failed jobs
- File Catalog
  - meta data search
  - find physical location of files
  - upload / download files

SiD Plans for Computing

Main computing resources for DBD at PNNL, RAL, SLAC

- Used the grid since the Letter of Intent
- Moved to ILCDirac for grid computing / file catalog since DBD
- Planning broad support for distributed computing in collaboration between KEK / PNNL

Planning for Site Computing

- A group has been established by LCC to work on campus planning
- Understanding the requirements for CPU / storage / networking will be important for planning the computing center
  - online vs. offline computing
  - prompt reco vs. distributed computing
  - manpower / user support
- ILD has started this effort and has already some rather detailed drafts
  - Next slide by Akiya Miyamoto

Summary of Data size and CPU requirements

EM Energy	GeV	250	350	500	
Number of bunches	1/Train	1312	1312	1312	
Number of Train	1/sec	5	5	5	
Total number of BX per year	MBxs/10^7sec	65600	65600	65600	
Total number of events/fb^1	kevents	2380.3	1927.5	2436.8	eL80.pR30
Int Lumi for 10^7	fb^-1	75	100	180	
Number of signal events/10^7	k	173100	189237	435000	eL80.pR30 only
# of Signal events per BX		2.64E-03	2.88E-03	6.63E-03	(0.2 to 0.7 % of BXs)
Bhabha cross section	pb	2319	1184	581	CosTheta<0.996(EndCap CAL まで)
# of Events/fb^1	kevents	2319000	1184000	581000	
# of Events/10^7 sec	MEvents/year	173.925	118.4	104.58	
Bhabha / BX		2.65E-03	1.80E-03	1.59E-03	
(Bhabha+Signal)/BX		5.29E-03	4.69E-03	8.23E-03	
gamma-gamma-hadron/Bx		0.2	0.33	1.7	
Monte Calro					
Sim Data size/fb	GB	261.5	228.0	297.5	
Rec +DST size/fb (x2.03)	GB	530.8	462.8	603.9	
(SIM+REC+DST) for 10xLumi	PB	0.398	0.463	1.087	x10 of raw data statistics
x2 for +bhabha+eemumu	PB	0.796	0.926	2.174	
Data size (for one set)					
Raw Data (RD)	PB/10^7 sec	5.5	6.2	8.9	
Fast Physics Data(FPD)		0	0	0	
Online Processed Data (OPD)	PB/10^7 sec	0.223	0.252	0.361	(2%xRD+REC&DST(x2.03))
Offline Reconstructed Data (ORD)	PB/10^7 sec	0.335	0.378	0.542	OPDx1.5
DST		0	0	0	
MC Data (Sim+REC+DST)	PB/10^7	0.796	0.926	2.174	
Data size (including copy)					
Raw Data (RD)	PB/10^7 sec	16.5	18.6	26.7	x3 (ILC site, EU, USA)
Fast Physics Data(FPD)		0	0	0	
Online Processed Data (OPD)	PB/10^7 sec	2.233	2.517	3.613	x10 copies
Offline Reconstructed Data (ORD)	PB/10^7 sec	3.350	3.776	5.420	x10 copies
DST		0.012	0.014	0.032	x0.1 of MC DST (10 copies)
MC Data (DST)	PB/10^7	0.118	0.137	0.321	x10 copies of MC DSTs
				0.000	
Total Data size	PB/10^7	22.212	25.043	36.087	
CPU					
MC CPU time(fb-1) for Sim	CPU days	228.3	237.2	364.4	
MC CPU days for 10xLumi with 1k cor	CPU daysx1k core	205.5	284.6	787.1	REC CPU/SIM CPU ~ 0.2 assumed
CPU for OPD and ORD	CPU daysx1k core	13.7	19.0	52.5	x2 of MC REC, x2 for OPD & ORD
Total CPU	CPU daysx1k core	219.2	303.6	839.6	

Collaborative Tools

- Code repositories are all open
  - public read access, no account needed
- Documentation lives at https://confluence.slac.stanford.edu/display/ilc/Home
  - Pages are backed up / versioned. No need to create duplicates to change something.
  - Pages are very easy to move from private (hidden) personal space into the public space. Hyperlinks will be automatically updated.
- Help is available at http://forum.linearcollider.org/
  - Public history -- better than mailing lists
  - Experts from both ILD and SiD are subscribed

Getting Started

Starting point for newcomers:
https://confluence.slac.stanford.edu/display/ilc/Home

Installation instructions for software + examples
https://confluence.slac.stanford.edu/display/~stanitz/From+Zero+to+SiD-Installation
https://confluence.slac.stanford.edu/display/~stanitz/From+Zero+to+SiD+-+Running+Sim+Reco

Please contact me, Norman Graf or Marcel Stanitzki if you want to get involved. We will help you get set up.

Summary

SiD software and computing are a solid foundation for your studies of detector performance / reconstruction capabilities /
→ tested in large-scale production
→ uses a test suite to maintain correctness

It's easy to get started
→ common tools with ILD for reconstruction / analysis
→ forum and online documentation sites for questions
→ a friendly community

There are enough resources for sophisticated studies
→ increasing CPU and storage capacity at various sites

Participate with your ideas for SiD!

Thank you for your attention.

ご清聴ありがとうございました

Backup

Introduction to ILCDirac

- The Grid is a heterogeneous set of computing sites
  - Different architectures, configurations, limitations
- Any tool that claims it can hide this heterogeneity from you is lying
- Dirac is a service to submit computing jobs to grid sites. Similar to your local batch farm
- ILCDirac wraps several ILC applications and executes them with your credentials on grid sites

Example ILCDirac Script (snippet)

See https://confluence.slac.stanford.edu/display/~jstrube/RecoChain.py for complete example

```
from DIRAC.Core.Base import Script
Script.parseCommandLine()from ILCDIRAC.Interfaces.API.DiracILC import Dir
dirac = DiracILC(True,"some_job_repository.rep")
from ILCDIRAC.Interfaces.API.NewInterface.UserJob import UserJob
job = UserJob()from ILCDIRAC.Interfaces.API.NewInterface.Applicati  import SLIC
slic = SLIC()
slic.setVersion('v2r9p8')
slic.setInputFile("LFN:/ilc/prod/ilc/some/file.stdhep")
slic.setSteeringFile('MyMacro.mac')
slic.setDetectorModel('sidloi3')
slic.setOutputFile("out.slcio")
res = job.append(slic)
job.setName("MyJobName")
job.setJobGroup("Agroup")
job.setCPUTime(86400)
job.setInputSandbox(["file1","file2"])
job.setDestination("LCG.CERN.ch")
job.setBannedSites(['LCG.DESY-HH.de', 'LCG.DESYZN.de', 'LCG.KEK.jp'])
job.setCPUTime(50000)
job.setSystemConfig('x86_64-slc5-gcc43-opt')
job.setOutputData("out.slcio", "sidloi3/analysis", "PNNL-SRM")
job.setOutputSandbox(['*.log', '*.xml', '*.lcsim', '*.steer'])
job.submit()
```

Initialization

Set up the application (SLIC)

from input sandbox

from file catalog

Deal with Grid specifics:
- Don't interfere with ILD
- block broken sites
- Make sure binaries run
- require CPU time

log files (web frontend)

grid output data

How to access the grid

1. Obtain grid certificate from your local authority
2. Register with the ILC VO (Do not use the same certificate for ILC and LHC)
3. Register with ILCDirac and follow tutorials
   a. https://twiki.cern.ch/twiki/bin/view/CLIC/DiracForUsers
   b. https://confluence.slac.stanford.edu/display/ilc/Running+LCSim+Analysis+Jobs+on+the+Grid+with+DIRAC