

Computing Meeting Summary

Takanori Hara
Thomas Kuhr

SuperBelle
General
Meeting

19.03.2009

Agenda

- Requirements to Software Framework and Comparison of Possible Candidates Ryosuke Itoh
- ILC Software Framework for sBelle Zbynek Drasal
- Current Status and Plan on Grid at KEK/CRC Go Iwai
- Belle MC Production on Grid Hideyuki Nakazawa
- Production of MC events on the Amazon EC2 Cloud Tom Fifield
- Prototyping a Distributed Computing Environment Tom Fifield
- Discussions on Code Management Thomas Kuhr

Requirements:
















- Katayama-san (and I) are now trying to collect requirements to the framework starting from the use-case study.
- It is on the wiki:
<http://wiki.kek.jp/display/sbelle/Use+cases>

Use cases

作成者: [KATAYAMA Nobuhiko](#): 最終更新者: [KATAYAMA Nobuhiko](#): 最終更新日: Feb 24, 2009

Here we would like to write down use cases of the framework.

子ページ (15)

-  Japanese version (incomplete and stopped as of Feb, 23)
-  Use case 1 - Itoh san
-  Use case 2 - MacNaughton san (shifter)
-  Use case 3 - Adachi san
-  Use case 4 - Mike Jones san
-  Use case 5 - Ozaki san
-  Use case 6 - Iwasaki san
-  Use case 7 - Nishida san
-  Use case 8 - Hara san
-  Use case 9 - Nakamura san
-  Use case 10 - Schumann san
-  Use case 11 - Inami san
-  Use case 12 - Wicht san
-  Use case 13 - Kinoshita san
-  Use case 14 - Uehara san

Use case 1 - Itoh san

作成者: [KATAYAMA Nobuhiko](#): 最終更新者: [KATAYAMA Nobuhiko](#): 最終更新日: Feb 24, 2009

Itoh san is responsible for the DAQ system for Belle. We use Basf for online data taking system. Just like tiny streams become big river, data from hundreds of thousands of sensors are gathered event by event and events are build and high level software triggering is performed. Some events are thrown away and others are kept and written to storage system. The daq system is almost all automated.

Functional requirements:

1. parallel processing through out the daq system
2. synchronization of events during the event building processes
3. software trigger to select events should be run
4. results of software trigger should be sent to colleagues(not just downstream)
5. data should be corrected for calibration due to electronics and other things
6. must deal with wave form sampling data somewhere (upstream/hardware?)
7. zero suppression and compression
8. need to monitor status of processes and sensors
9. begin run/ end run processing: system resets, reload constants
10. output file size, change file
11. run changes should be instantaneous
12. the order of events are not too important. There could be more than one output streams
13. multiple streams for different types of events?

1. Requirements to SB Framework

a) Modular structure = Software Bus

- * Dynamic-linkable user-supplied analysis modules
- * Flexible execution control of modules

b) Object I/O

- * Capability to read/write C++ object.
- * Independent of specific data models

c) Interface to distributed computing

- * Parallel processing
- * Data file access over WAN (i.e. GRID)

d) Input/Output file management

- * Database interface

e) Integrated histogram/N-tuple management

f) Programmable script parser for the control

g) Dynamic configuration of framework components

h) Target Use: DAQ(readout/HLT), data production, MC production, user analysis

- * Unified framework for all cases is desired.

i) Backward compatibility

- * Compatibility with Belle software/data

3. Future direction and plan

- Framework provides the basis for various software development at SuperKEKB
 - > Need to decide as early as possible
- Practically speaking, realistic candidates are two:
 1. “Marlin” offered by German group
 2. “roobasf” being jointly developed by Belle and HSC
 - * GAUDI – no experts around here
 - * BASF – taken over by roobasf
- Decision should be made by considering the actual requirements to the framework.
- We will focus on the discussion on the requirements and make the decision by autumn at the latest.

→ Same applies to data format!

Personal pros and cons to candidates

a) Marlin

pros:

- It is actually being used for the SuperKEKB detector simulation.
- Basic required functions as a framework are already built-in.
- Nice scripting system based on XML.

cons:

- Too heavy dependence on LC data model. If we decide to use Marlin, that means SuperKEKB data model has to be constructed based on LC data model.
 - > big issue which affects on the whole SuperKEKB software design.
- Marlin is basically the “old generation” framework to which BASF belongs as well.
 - * Object I/O (LCIO) is old style (just like Panther/BASF).
- Worries on future support.
 - * Support for SIO (SLAC product) is already terminated.
 - * Development team is outside SuperKEKB.
 - * LC data model is not combat proven. It is basically developed for MC study.
 - * Not established even in ILD community. Merging with JSF is discussed.
- No parallel processing support
 - > External parallel processing framework is necessary to be developed for massive data production.

Personal pros and cons to candidates

a) Marlin

pros:

- It is actually being used for the SuperKEKB detector simulation.
- Basic required functions as a framework are already built-in.
- Nice scripting system based on XML.

cons:

- Too heavy dependence on LC data model. If we decide to use Marlin, that means SuperKEKB data model has to be constructed based on LC data model.
 - > big issue which affects on the whole SuperKEKB software design.
- Marlin is basically the “old generation” framework to which BASF belongs as well.
 - * Object I/O (LCIO) is old style (just like Panther/BASF).
- Worries on future support.
 - * Support for SIO (SLAC product) is already terminated.
 - * Development team is outside SuperKEKB.
 - * LC data model is not combat proven. It is basically developed for MC study.
 - * Not established even in ILD community. Merging with JSF is discussed.
- No parallel processing support
 - > External parallel processing framework is necessary to be developed for massive data production.

Data model can be changes, have to write streamer code

Personal pros and cons to candidates

a) Marlin

pros:

- It is actually being used for the SuperKEKB detector simulation.
- Basic required functions as a framework are already built-in.
- Nice scripting system based on XML.

cons:

- Too heavy dependence on LC data model. If we decide to use Marlin, that means SuperKEKB data model has to be constructed based on LC data model.
 - > big issue which affects on the whole SuperKEKB software design.
- Marlin is basically the “old generation” framework to which BASF belongs as well.
 - * Object I/O (LCIO) is old style (just like Panther/BASF).
- Worries on future support.
 - * Support for SIO (SLAC product) is already terminated.
 - * Development team is outside SuperKEKB.
 - * LC data model is not combat proven. It is basically developed for MC study.
 - * Not established even in ILD community. Merging with JSF is discussed.
- No parallel processing support
 - > External parallel processing framework is necessary to be developed for massive data production.

Support taken
over by LCIO
team

Personal pros and cons to candidates

a) Marlin

pros:

- It is actually being used for the SuperKEKB detector simulation.
- Basic required functions as a framework are already built-in.
- Nice scripting system based on XML.

cons:

- Too heavy dependence on LC data model. If we decide to use Marlin, that means SuperKEKB data model has to be constructed based on LC data model.
 - > big issue which affects on the whole SuperKEKB software design.
- Marlin is basically the “old generation” framework to which BASF belongs as well.
 - * Object I/O (LCIO) is old style (just like Panther/BASF).
- Worries on future support.
 - * Support for SIO (SLAC product) is already terminated.
 - * Development team is outside SuperKEKB.
 - * LC data model is not combat proven. It is basically developed for MC study.
 - * Not established even in ILD community. Merging with JSF is discussed.
- No parallel processing support
 - > External parallel processing framework is necessary to be developed for massive data production.

Used by Calice
(40TB), EUDET,
LCTPC

Personal pros and cons to candidates

a) Marlin

pros:

- It is actually being used for the SuperKEKB detector simulation.
- Basic required functions as a framework are already built-in.
- Nice scripting system based on XML.

cons:

- Too heavy dependence on LC data model. If we decide to use Marlin, that means SuperKEKB data model has to be constructed based on LC data model.
 - > big issue which affects on the whole SuperKEKB software design.
- Marlin is basically the “old generation” framework to which BASF belongs as well.
 - * Object I/O (LCIO) is old style (just like Panther/BASF).
- Worries on future support.
 - * Support for SIO (SLAC product) is already terminated.
 - * Development team is outside SuperKEKB.
 - * LC data model is not combat proven. It is basically developed for MC study.
 - * Not established even in ILD community. Merging with JSF is discussed.
- No parallel processing support
 - > External parallel processing framework is necessary to be developed for massive data production.

LCIO used by JSF, merge into Marlin

Personal pros and cons to candidates

a) Marlin

pros:

- It is actually being used for the SuperKEKB detector simulation.
- Basic required functions as a framework are already built-in.
- Nice scripting system based on XML.

cons:

- Too heavy dependence on LC data model. If we decide to use Marlin, that means SuperKEKB data model has to be constructed based on LC data model.
 - > big issue which affects on the whole SuperKEKB software design.
- Marlin is basically the “old generation” framework to which BASF belongs as well.
 - * Object I/O (LCIO) is old style (just like Panther/BASF).
- Worries on future support.
 - * Support for SIO (SLAC product) is already terminated.
 - * Development team is outside SuperKEKB.
 - * LC data model is not combat proven. It is basically developed for MC study.
 - * Not established even in ILD community. Merging with JSF is discussed.
- No parallel processing support
 - > External parallel processing framework is necessary to be developed for massive data production.

Open for
collaboration

b) roobasf

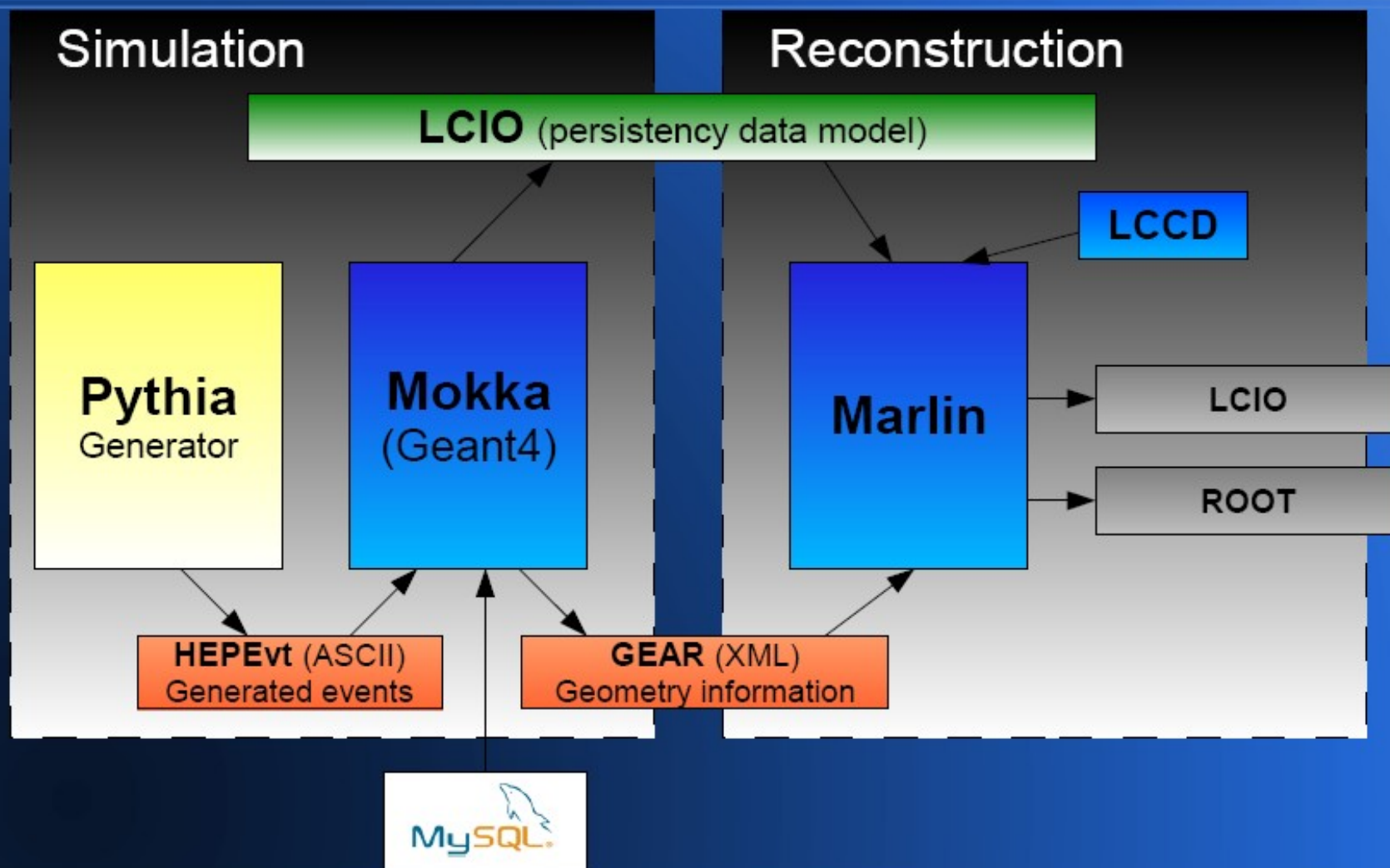
pros:

- All basic functions are built in the prototype already
- Smooth extrapolation from existing BASF/Belle software
- Independent of the choice of data model
- New generation framework with true object I/O support by ROOT IO, to which GAUDI belongs as well.
- Built-in parallel processing support
 - > unified framework for DAQ, production and user analyses (as well as BASF).
- Being developed/supported inside SuperKEKB collab.

cons:

- Not yet used in SuperKEKB study although prototype is ready and being used for HSC.
- Scripting system of the prototype is poor. It will be replaced with Python-based system.

ILC Software Framework – Scheme



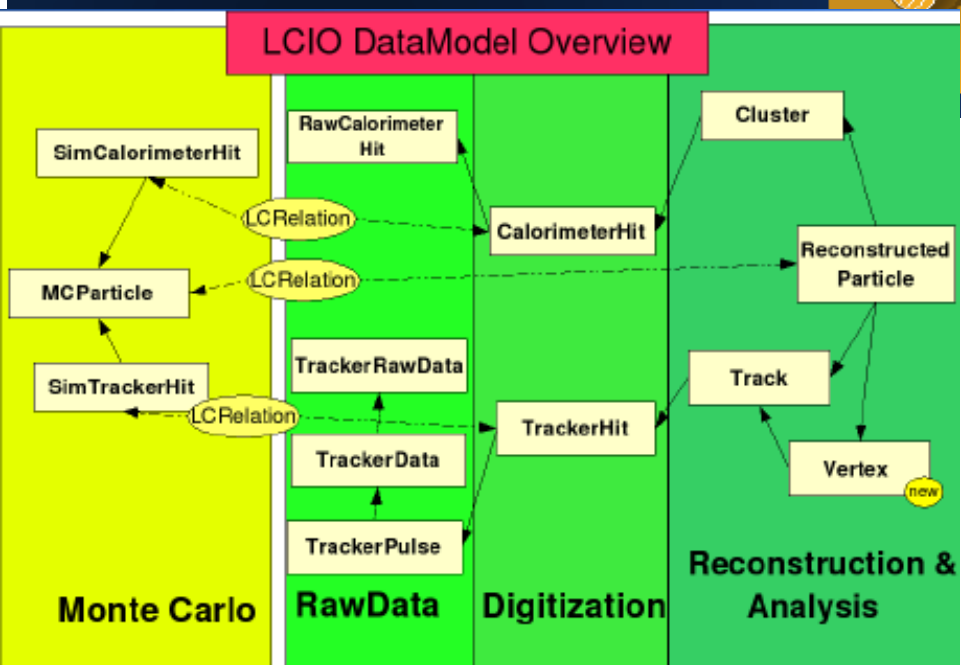
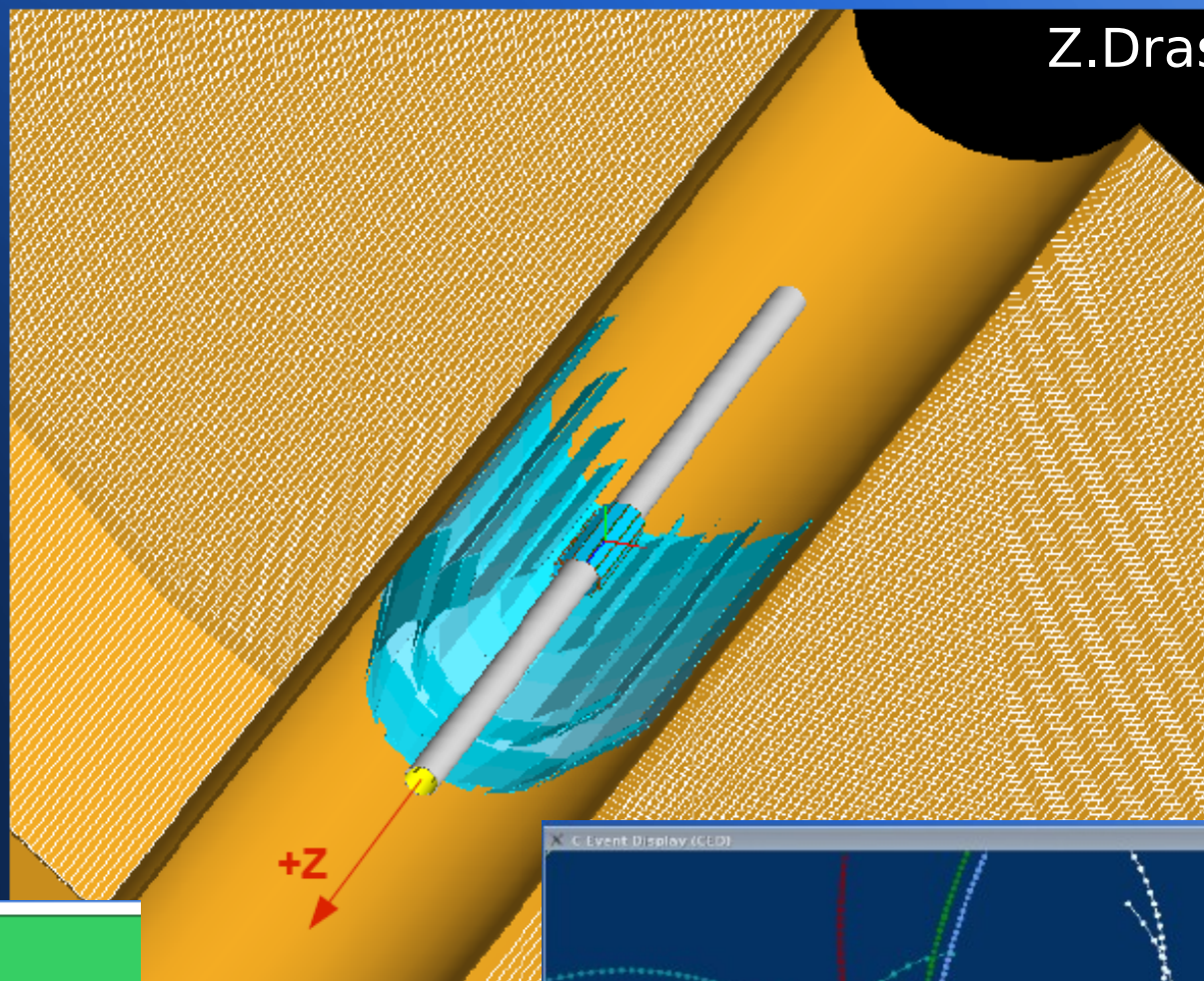
- *Implemented tracker:*

- Pixel layers (VXD)
- Strip layers (VXD)
- CDC

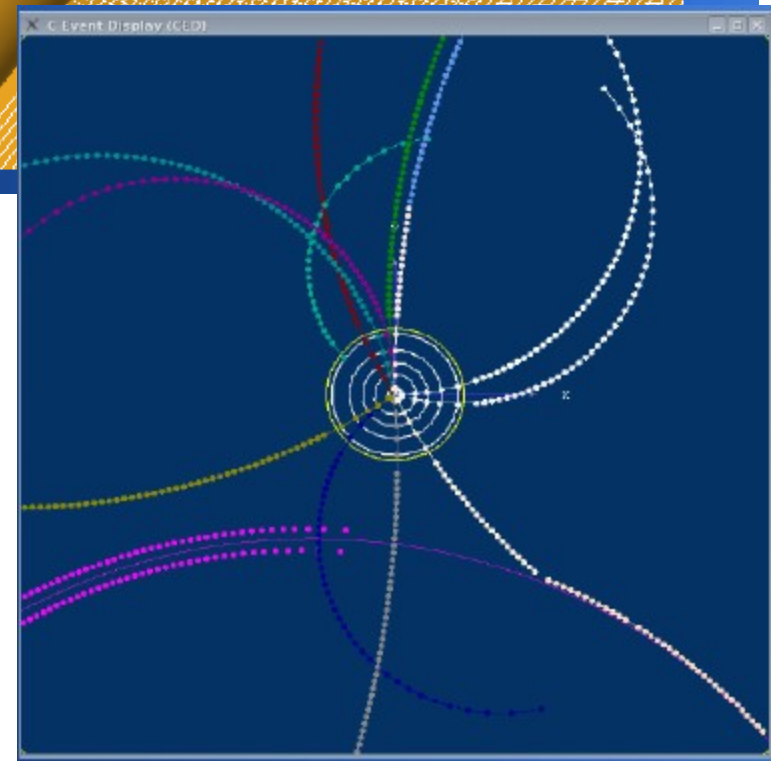
- Clustering

- Tracking

Z.Drasal



Bidirectional relations

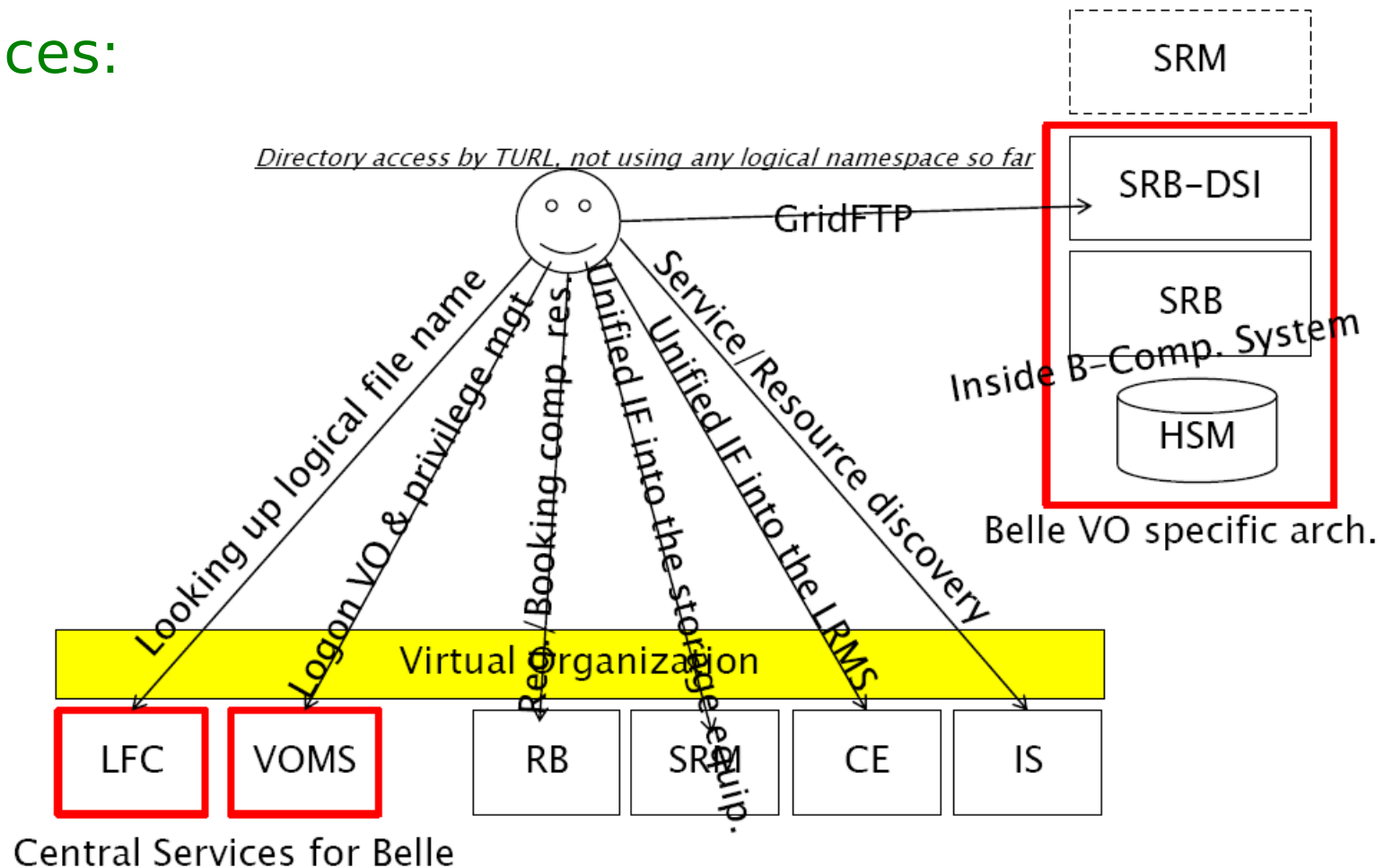


Framework Discussion

- Continue to collect/refine requirements
 - Easy to use
 - Fast
 - ...
- ➔ “Competition” of frameworks beneficial
 - Can check different approaches against requirements
 - Take best solution for each feature
- *Aim for convergence on framework in autumn*

Grid at KEK

- Services:



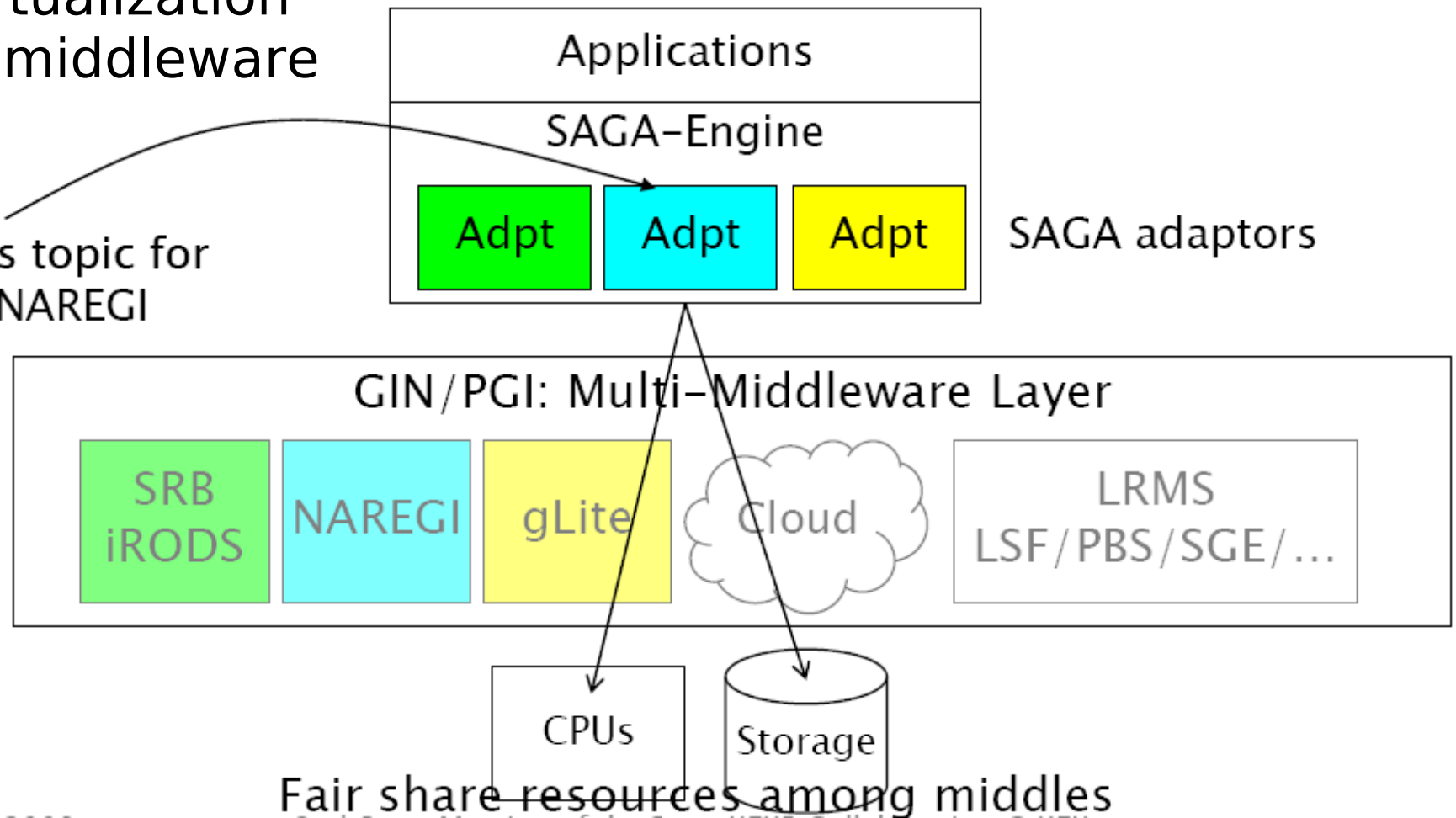
- Resources: 62 CPUs, 8 TB disk, tape

SAGA

- A Simple API for Grid Applications

- Virtualization of middleware

Today's topic for
SAGA-NAREGI

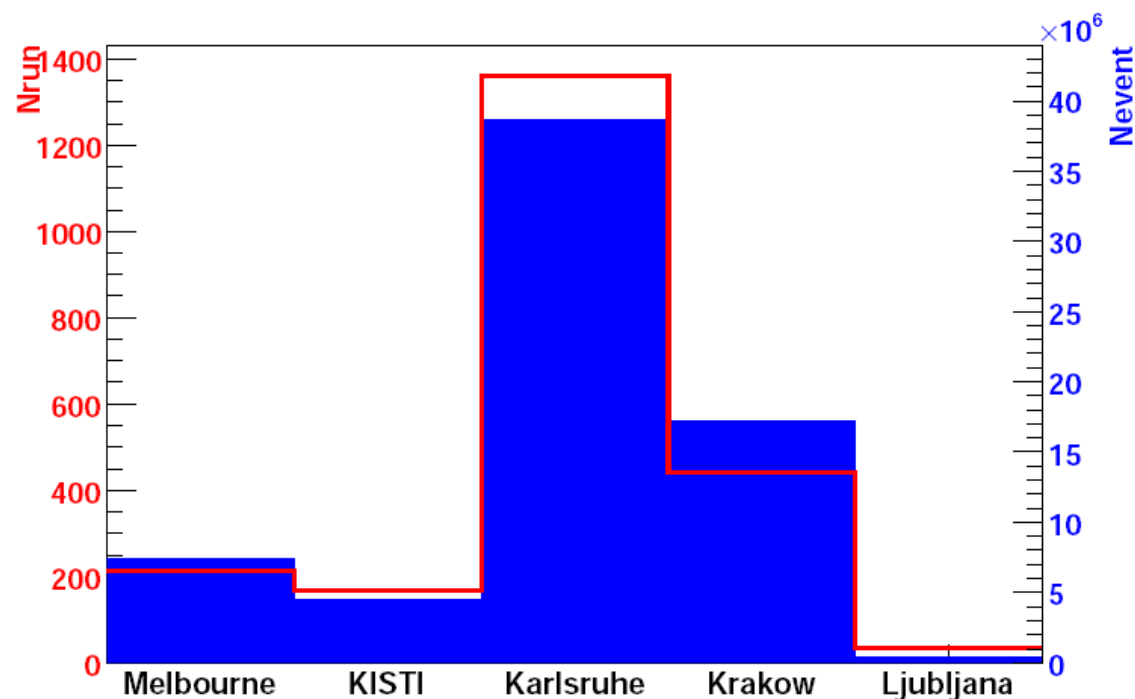


Belle MC Production on the Grid

Belle VO sites

Institute	CPU	Status
KISTI	60	new
Merbourne	60	new
Karlsruhe	400	new
Ljubljana	80	new
ASGC	120	Stop due
KEK01	12	Unavailal
KEK02	32	Under up
Krakov	90	

One Week Statistics



Summary

```

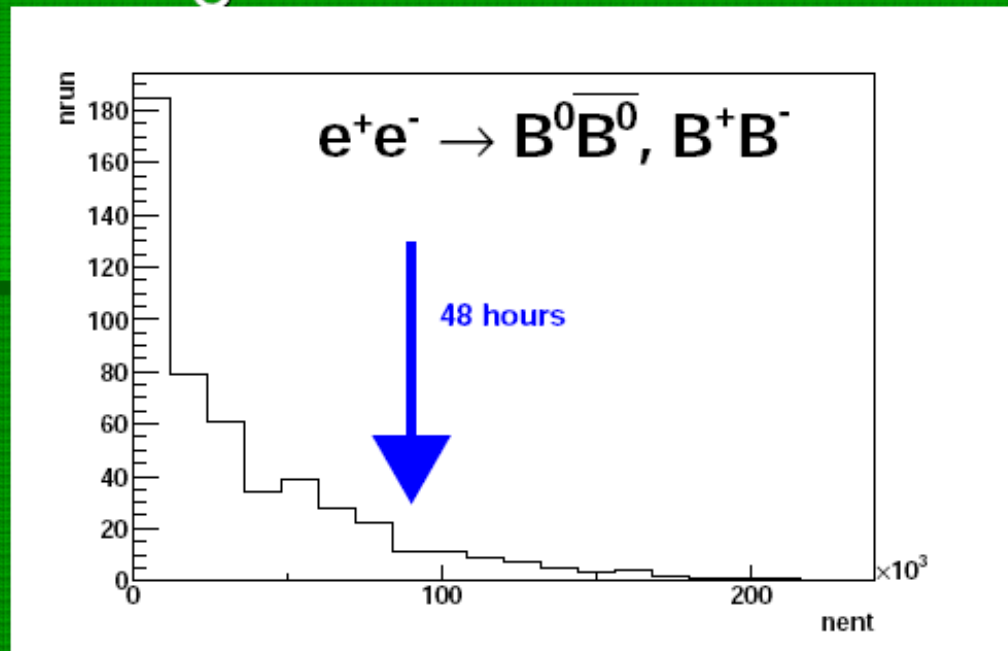
ファイル(E) 編集(E) 表示(V) 端末(T) タブ(T) ヘルプ(H)
Every 120.0s: jobstatus.rb Sun Mar 15 23:00:27 2009
agh2.atlas.unimelb.edu.au
  Done(Success) 9
  Running 37
ce-alice.sdfarm.kr
  Done(Success) 1
  Running 23
ce-1-fzk.gridka.de
  Done(Success) 14
  Scheduled 3
  Running 233
lcgce.ijs.si
  Running 45
ce.cyt-kr.edu.pl
  Scheduled 2
  Running 68

```

Problems

- Storage, HSM busy
 - Stays inside B system and shared with Belle users
 - Access to HSM sometimes concentrate on one of four server
- Job processing time exceed limits

Problem not caused by grid jobs



Keep
concept
of one file
per run?

Ganga MC Production

H.Nakazawa

ファイル(F) 編集(E) 表示(V) 端末(T) タブ(T) ヘルプ(H)

```
[nkzw@dg14 work]$ ganga
```

```
*** Welcome to Ganga ***
```

```
Version: Ganga-5-1-8
```

```
Documentation and support: http://cern.ch/ganga
```

```
Type help() or help('index') for online help.
```

```
This is free software (GPL), and you are welcome to redistribute it  
under certain conditions; type license() for details.
```

```
Ganga.GPIDev.Lib.JobRegistry : INFO Found 26 jobs in "jobs", completed in 0 seconds
```

```
Ganga.GPIDev.Lib.JobRegistry : INFO Found 0 jobs in "templates", completed in 0 seconds
```

Backend=LCG

```
In [1]: j=Job(backend=LCG(), application=Executable(exe='/bin/sh', args=['evtgen-mixed-04-all-e000063r000056-b20090127_0910.sh']))
```

```
In [2]: j.inputsandbox = ['/home/nkzw/Mixed.e63s04r00/script/evtgen-mixed-04-all-e000063r000056-b20090127_0910.sh']
```

```
In [3]: j.backend.CE = 'ce-1-fzk.gridka.de:2119/jobmanager-pbspro-belleXL'
```

```
In [4]: j.submit() → edg-job-submit
```

```
Ganga.GPIDev.Lib.Job : INFO submitting job 27
```

```
Ganga.GPIDev.Adapters : INFO submitting job 27 to LCG backend
```

```
Ganga.GPIDev.Lib.Job : INFO job 27 status changed to "submitted"
```

```
Out[4]: 1
```

```
In [5]: j.status
```

```
Out[5]: submitted
```

```
In [6]:
```

Grid job management
frontend used by
Atlas and LHCb

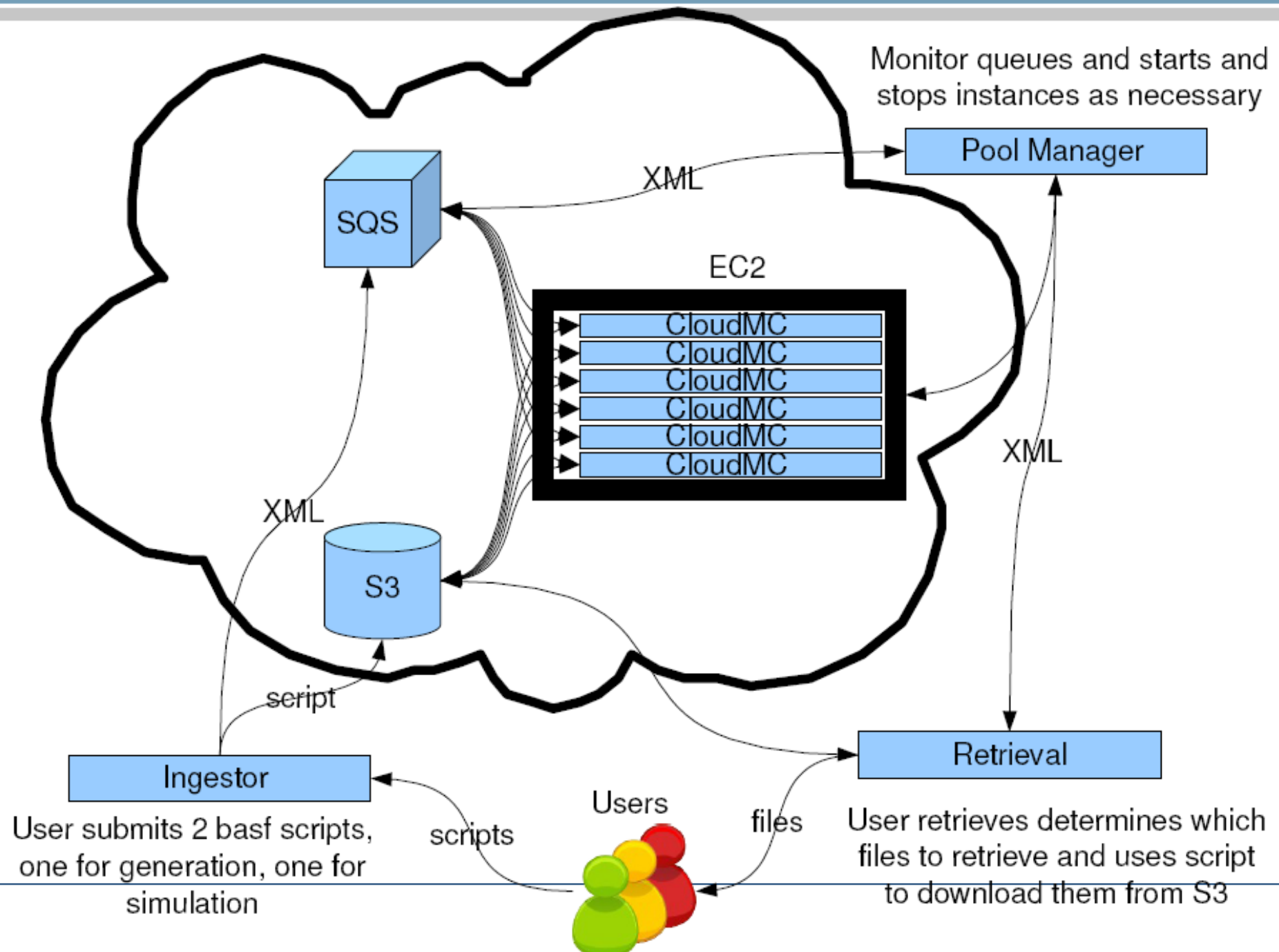
- Configuration and Job submission tool "Ganga" tried.

- Easy to see what we're doing
- Good to unified interface
- No convenience about functionality

We must write script for scripts, anyway.

- Have Belle MC Analysis working on EC2
- “cloud computing ... will provide extra resources on top of the baseline (grid) resources in case we are in urgent need of CPU power.”
- Automating it was the next step.
- Followed by a production MC run

Amazon Elastic Cloud Computing:
Commercial Resource Provider



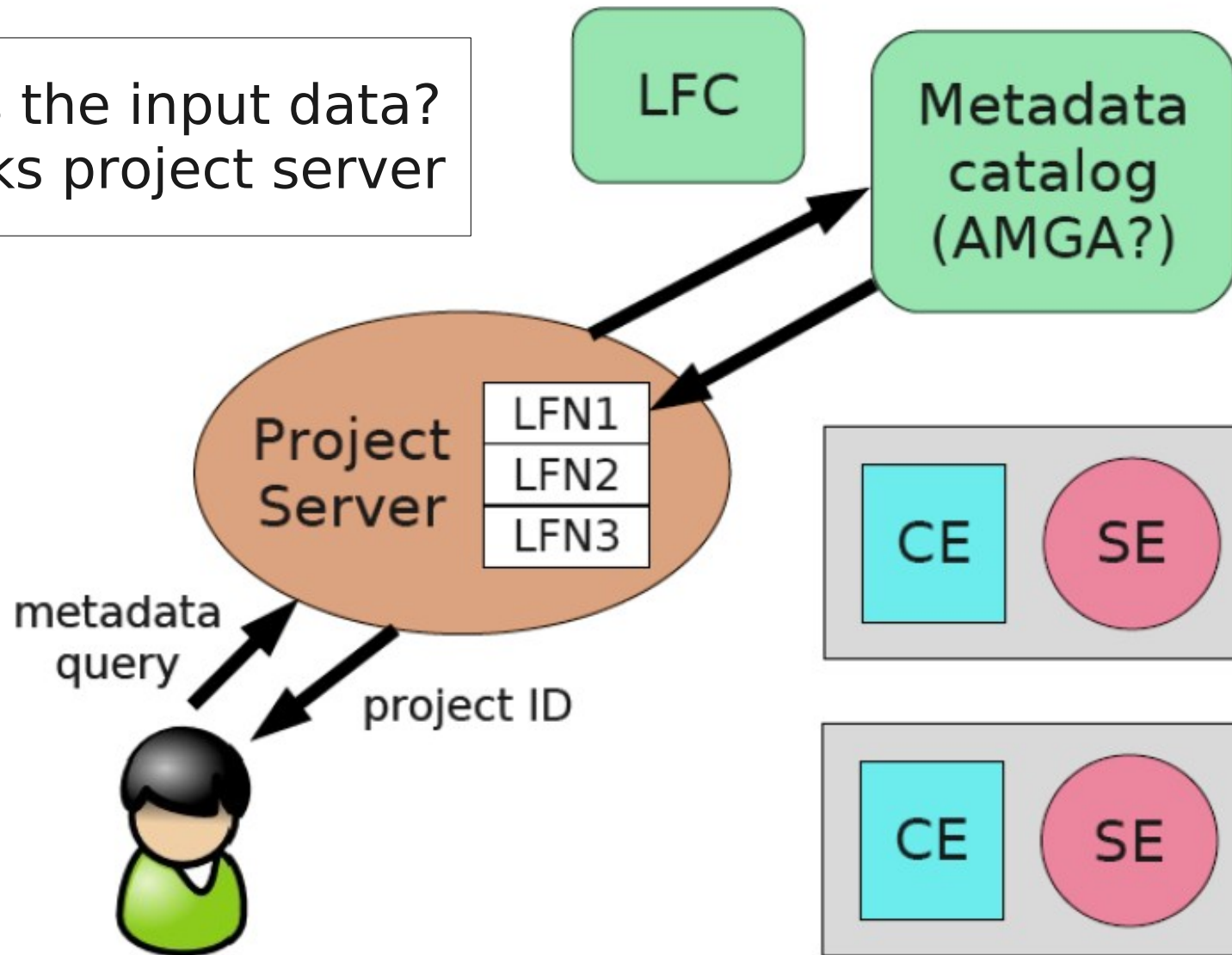


Reservation ID	Instance ID	AMI	State	Public DNS	Private DNS	Key	Type	Local Launch Time	Tag	
-fd46c194	i-ae71efc7	ami-7735d...	terminated			fifieldt	c1.xlarge	2009-03-15 14:22:32		
-fd46c194	i-a171efc8	ami-7735d...	terminated			fifieldt	c1.xlarge	2009-03-15 14:22:32		
-e940c780	i-1169f778	ami-7735d...	terminated			fifieldt	c1.xlarge	2009-03-15 16:16:03		
-aa40c7c3	i-d669f7bf	ami-7735d...	running	ec2-174-129-116-137.compu...	ip-10-250-163-223.ec2.internal	fifieldt	c1.xlarge	2009-03-15 16:19:23		
-aa40c7c3	i-a969f7c0	ami-7735d...	running	ec2-75-101-243-21.compute...	ip-10-250-43-159.ec2.internal	fifieldt	c1.xlarge	2009-03-15 16:19:23		
-aa40c7c3	i-a869f7c1	ami-7735d...	running	ec2-75-101-226-223.comput...	ip-10-250-94-207.ec2.internal	fifieldt	c1.xlarge	2009-03-15 16:19:23		
-aa40c7c3	i-ab69f7c2	ami-7735d...	running	ec2-67-202-38-36.compute-1...	ip-10-250-222-239.ec2.internal	fifieldt	c1.xlarge	2009-03-15 16:19:23		
-aa40c7c3	i-aa69f7c3	ami-7735d...	running	ec2-174-129-114-89.comput...	ip-10-251-63-239.ec2.internal	fifieldt	c1.xlarge	2009-03-15 16:19:23		
-aa40c7c3	i-ac69f7c5	ami-7735d...	running	ec2-72-44-33-190.compute-1...	ip-10-250-230-255.ec2.internal	fifieldt	c1.xlarge	2009-03-15 16:19:23		
-aa40c7c3	i-af69f7c6	ami-7735d...	running	ec2-174-129-153-242.compu...	ip-10-250-71-63.ec2.internal	fifieldt	c1.xlarge	2009-03-15 16:19:23		
-aa40c7c3	i-ae69f7c7	ami-7735d...	running	ec2-75-101-205-44.compute...	ip-10-250-198-239.ec2.internal	fifieldt	c1.xlarge	2009-03-15 16:19:23		
-aa40c7c3	i-a169f7c8	ami-7735d...	running	ec2-67-202-1-207.compute-1...	ip-10-250-146-223.ec2.internal	fifieldt	c1.xlarge	2009-03-15 16:19:23		
-aa40c7c3	i-ad69f7c4	ami-7735d...	terminated			fifieldt	c1.xlarge	2009-03-15 16:19:23		
-574fc83e	i-2766f84e	ami-7735d...	running	ec2-174-129-156-115.compu...	ip-10-250-211-79.ec2.internal	fifieldt	c1.xlarge	2009-03-15 16:29:30		
-164fc87f	i-dc66f8b5	ami-7735d...	pending			fifieldt	c1.xlarge	2009-03-15 16:35:35		
-164fc87f	i-c566f8ac	ami-7735d...	running	ec2-75-101-216-251.comput...	ip-10-250-210-175.ec2.internal	fifieldt	c1.xlarge	2009-03-15 16:35:35		
-164fc87f	i-c466f8ad	ami-7735d...	running	ec2-75-101-217-173.comput...	ip-10-250-35-175.ec2.internal	fifieldt	c1.xlarge	2009-03-15 16:35:35		
-164fc87f	i-c766f8ae	ami-7735d...	running	ec2-72-44-36-91.compute-1...	ip-10-250-34-47.ec2.internal	fifieldt	c1.xlarge	2009-03-15 16:35:35		
-164fc87f	i-d966f8b0	ami-7735d...	running	ec2-174-129-182-249.compu...	ip-10-250-187-207.ec2.internal	fifieldt	c1.xlarge	2009-03-15 16:35:35		
-164fc87f	i-d866f8b1	ami-7735d...	running	ec2-75-101-243-99.compute...	ip-10-250-30-159.ec2.internal	fifieldt	c1.xlarge	2009-03-15 16:35:35		
-164fc87f	i-db66f8b2	ami-7735d...	running	ec2-75-101-181-136.comput...	ip-10-250-127-143.ec2.internal	fifieldt	c1.xlarge	2009-03-15 16:35:35		
-164fc87f	i-da66f8b3	ami-7735d...	running	ec2-174-129-178-75.comput...	ip-10-250-63-239.ec2.internal	fifieldt	c1.xlarge	2009-03-15 16:35:35		
-164fc87f	i-dd66f8b4	ami-7735d...	running	ec2-67-202-16-241.compute...	ip-10-250-218-95.ec2.internal	fifieldt	c1.xlarge	2009-03-15 16:35:35		
-164fc87f	i-c666f8af	ami-7735d...	terminated			fifieldt	c1.xlarge	2009-03-15 16:35:35		

- Sorry, we only managed to do 752,233 events in time for the presentation
- CPU cost: \$80
 - 20 Instances, 4 hours 57minutes
- Storage cost: \$0.20
 - Storage on S3: Addbg 3.1Gb, pgen 0.5Gb, results 37Gb, \$6.08/month or \$0.20/day
- Transfer cost: \$6.65
 - Addbg, pgen in: \$0.36, mdst out: \$6.29
- **Total Cost: \$86.85**

M.Sevior (previous estimate):
Comparable to cost of cluster

Where's the input data?
→ job asks project server



Thomas Kuhr, 2009-02-17

“Tier 0” KEK

Possible to simulate in ~12 **gLite** services across 2 virtual sites (12 CPUs, 12GB Ram, 200Gb disk) [production needs a lot more!]

– Use virtual machines = low to zero cost

Central LFC

AGMA

VOMS

TopBDII

SE

Site BDII

DSs

Have to
know name of
collaboration
to request
new VO

“Tier 1” Germany, Korea, Aus etc

UI

TopBDII

SE

Site BDII

DS

- Working on defining requirements
- Use cases from Katayama san
- Data Handling
 - The system must allow individual users access to raw data (Use Case #4)
 - The system must support returning histograms and other information (Use Case #4, Use Case #9)
 - The system must facilitate the access of another users data, within permission restrictions (Use Case #7)
 - The system must interface with the QAM System (Use Case #13)
- Job Handling
 - The system must support masses of parallel Jobs to a fine grain (Use Case #3)
 - The system must support different run lengths for jobs (Use Case #3)
 - The system must minimize idle CPU time between jobs (Use Case #3)
 - The system should support the ability to re-run with certain modules only (Use Case #3)
 - The system must allow for priority to be given to certain short-running jobs (Use Case #4)
 - The system must allow for automatic determination of CPU and Storage resources to use, where the user does not specify them (Use Case #4)
 - The system must allow similar jobs to be run with minimal changes (i.e. parameterised jobs) (Use Case #5)
 - The system must provide an appropriate level of debugging information (Use Case #5)
 - The system must provide an interface that allows users to manage their jobs (Use Case #7)
 - The system must allow computation to be run on a selected number of events (Use Case #7)
 - The system must run on an entire data set with the shortest elapsed time possible (Use Case #7)
 - The system must allow the use of geant4 (Use Case #8)

4 students
from Krakow
will join

Code Management

Code repository

- Decided to use subversion (svn)
- Need volunteer to take care of code repository

Discussion on language

- Agreement on C++ as first language (although very complicated, but HEP standard)
- Want to keep complexity of software as low as possible
- *Always use C++ by default for official code, except for specific cases where second language is obviously better*
- Have only one second language

Code Management

Coding recommendations:

- first draft
- **Try to keep your code as simple as possible!**

Wiki (confluence) at KEK

- <http://wiki.kek.jp/display/sbelle/SuperBelle+Wiki+Home>
- World readable, individual registration via paper form

twiki at EKP

- <http://www-ekp.physik.uni-karlsruhe.de/~twiki/bin/view/SuperBelle>
- Accessible by group account
- Want to have twiki at KEK
- **Need volunteer for web/twiki page**

Summary

- Evaluation of framework options progressing
 - Uses cases/requirements help to focus discussion
- Support of grid services from KEK is essential
- Belle MC production on the grid works well
- Cloud computing proved to be a realistic option in case supplemental resources are needed
- Concept for data access in distributed environment exists
 - Work on implementation is starting
- Code repository and first language decided
- Still many possibilities to contribute:
code management, web/twiki page, database