CLAS12 Software Planning and Preparations

G.P. Gilfoyle, D.P. Weygand, V. Gyurjyan, and K. Hicks

Outline

1. The Goal
2. Challenges - Requirements
3. CLAS12 Software Review - May, 2011
4. Ideas for Organizing Software
5. The CLARA Project
6. Summary and Conclusions

The Goal

> “it is the desire of the laboratory to have all computing systems and software ready, so that the time from beam on target to physics journal articles is as short as possible”.

Final Report

Information Technology for the 12 GeV Era - Internal Review

The Goal - Some History

Latency of run group publications in the 6 GeV era

Table: Number of run groups with $1^{\rm st}$, $2^{\rm nd}$, $3^{\rm rd}$ publication within $\rm X$ months after data taking.

Latency	≤ 12	≤ 18	≤ 24	≤ 36	≤ 48	≤ 60	≤ 84	> 84
$1^{\rm st}$ paper	-	1	3	2	6	2	1	2
$2^{\rm nd}$ paper	-	-	2	3	3	3	3	2
$3^{\rm rd}$ paper	-	-	1	1	1	4	4	1

- Not all due to software.
- Steep learning curve, redundant efforts, modularity, testing, ...
- Desire to operate software more like HEP collaboration.

Challenges - Software Requirements for CLAS12

- DAQ - 10^{11} events/year, 1 PByte/year of storage.
- Calibrations - ONLINE and offline.
- Simulations - 10^{11} events/year, 800 cores.
- Reconstruction - 4 PByte/year, 400 cores.
- Post-Reconstruction Analysis - 500 cores.

	Cores	Disk (TByte)	Tape (TByte/yr)
DAQ	-	-	1100
Calibration	127	-	-
Simulation	828	65	327
Reconstruction	393	370	3700
Analysis	463	370	370
Sum	1811	805	5497

Information Technology for the 12 GeV Era

Charge:

Review ’aspects of Information Technology that impact preparations for and initial running of the 12 GeV science program’.

Findings and Recommendations:

- ‘no critical show-stoppers at this stage’.
- ‘software development groups for all halls appeared somewhat understaffed.’
- ‘Establish a more formal joint effort to ensure that analysis software is ready to meet the experimental requirements.’
- ‘No common process for defining requirements, no common management structure.’
- ‘~~12 GeV computing requirements ... (not funded ...)~~’

Website:`http://wwwold.jlab.org/conferences//IT12GeV/`

Software Organization - General

- Large software projects are

  complicated - hard to understand, but predictable

  complex - harder to understand, not as predictable

  chaotic - not understandable, unpredictable

Software Organization - General

- Large software projects are

  complicated - hard to understand, but predictable

  complex - harder to understand, not as predictable

  chaotic - not understandable, unpredictable
- Over 300,000 lines of code in CLAS repository.

Software Organization - General

- Large software projects are

  complicated - hard to understand, but predictable

  complex - harder to understand, not as predictable

  chaotic - not understandable, unpredictable
- Over 300,000 lines of code in CLAS repository.
- How do we get people to write ‘good’ code?

Software Organization - General

- Large software projects are

  complicated - hard to understand, but predictable

  complex - harder to understand, not as predictable

  chaotic - not understandable, unpredictable

- Over 300,000 lines of code in CLAS repository.
- How do we get people to write ‘good’ code?
  - Divide and conquer - Break the problem into small steps.

Software Organization - General

- Large software projects are

  complicated - hard to understand, but predictable

  complex - harder to understand, not as predictable

  chaotic - not understandable, unpredictable
- Over 300,000 lines of code in CLAS repository.
- How do we get people to write ‘good’ code?
  - Divide and conquer - Break the problem into small steps.
  - Modularity - each ‘function’ is independent of others.

Software Organization - General

- Large software projects are

  complicated - hard to understand, but predictable

  complex - harder to understand, not as predictable

  chaotic - not understandable, unpredictable
- Over 300,000 lines of code in CLAS repository.
- How do we get people to write ‘good’ code?
  - Divide and conquer - Break the problem into small steps.
  - Modularity - each ‘function’ is independent of others.
  - Multi-platform - Use everyone.

Software Organization - General

- Large software projects are

  complicated - hard to understand, but predictable

  complex - harder to understand, not as predictable

  chaotic - not understandable, unpredictable
- Over 300,000 lines of code in CLAS repository.
- How do we get people to write 'good' code?
  - Divide and conquer - Break the problem into small steps.
  - Modularity - each 'function' is independent of others.
  - Multi-platform - Use everyone.
  - Design in maintenance at the start.

Software Organization - General

- Large software projects are

  complicated - hard to understand, but predictable

  complex - harder to understand, not as predictable

  chaotic - not understandable, unpredictable

- Over 300 000 lines of code in CLAS repository

We are at a branch point in the CLAS software story.

steps.

  - Modularity - each ‘function’ is independent of others.
  - Multi-platform - Use everyone.
  - Design in maintenance at the start.

Software Organization - The CLARA project

CLAs12 Analysis and Reconstruction framework
- Vardan Gyurjyan.

Divide and conquer - break the problem into distinct pieces of code or data structures.

Modularity - 1 - each distinct piece has or gets what it needs to accomplish a task.

Modularity - 2 - an interface controls the interaction of the pieces.

Multi-platform - incorporate different languages.

Software Organization - The CLARA project

Service Definition

1. distinct piece of code or data structure that is reusable.
2. has little or no knowledge of the definitions of other components.
3. communicates with other services by passing data in a well-defined, shared format.

The CLARA project - Service-Oriented Architecture

Divide and conquer - break the problem into distinct pieces of code or data structures.

Services SHOULD be written to accomplish a well-defined, limited task, *i.e.*, cluster finding.

Modularity - 1 - each distinct piece has or gets what it needs to accomplish a task.

Services CANNOT access data used by other services.

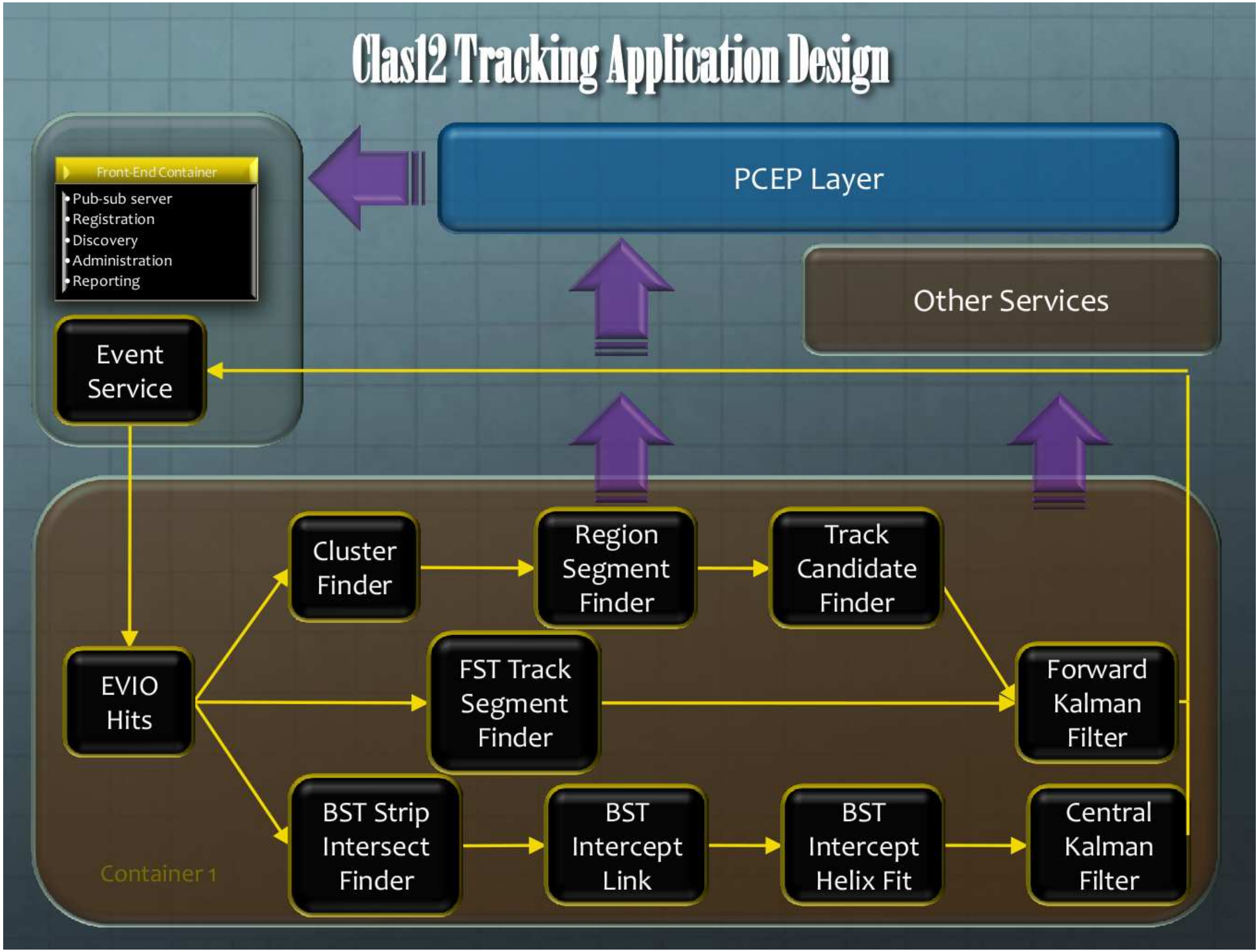
Modularity - 2 - an interface controls the interaction of the pieces.

Services interact/communicate only by passing data.

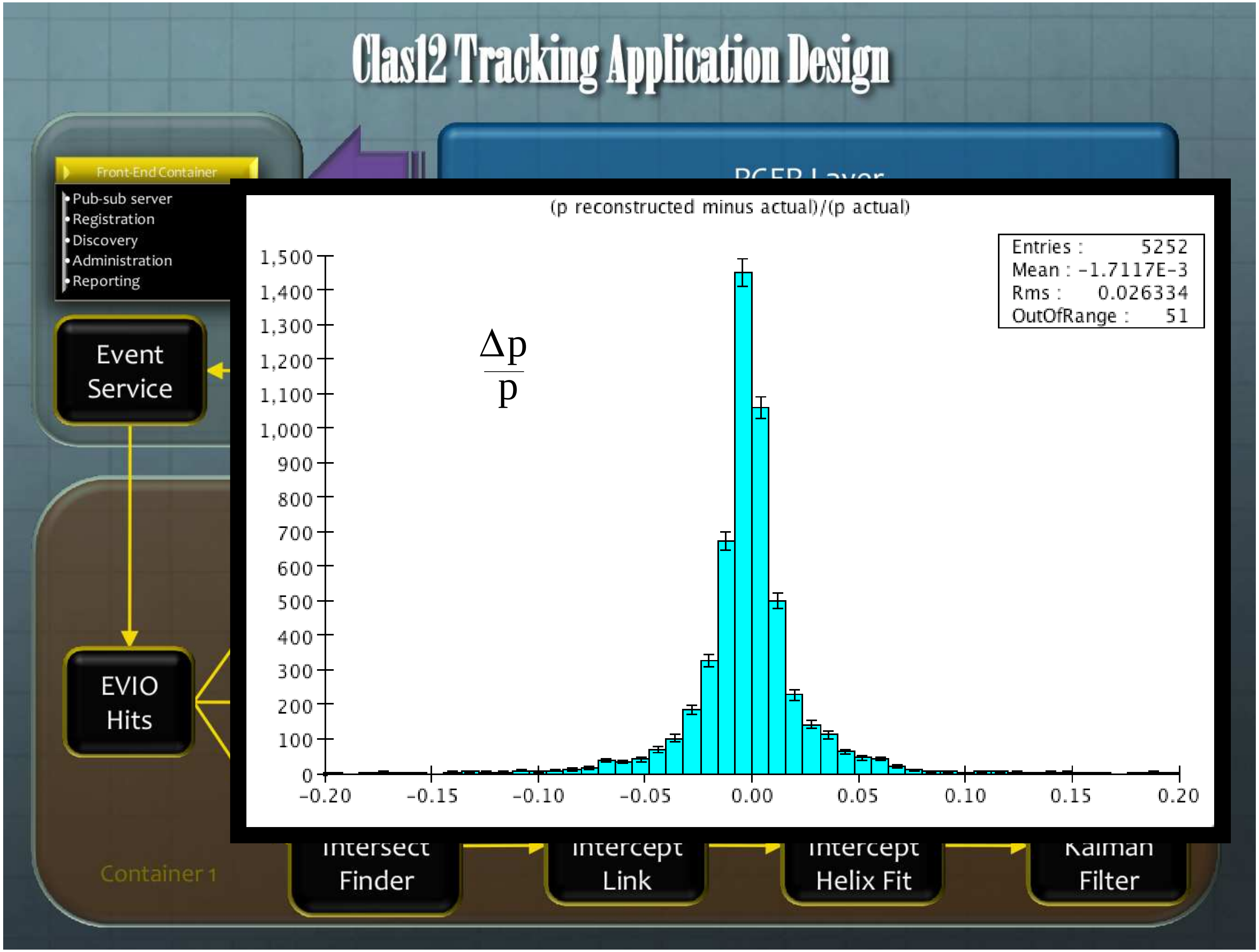
Multi-platform - should run on different machines, incorporate different languages.

CLARA can work with C++, Java, Python, Fortran (indirectly).

The CLARA project - A Tracking Example

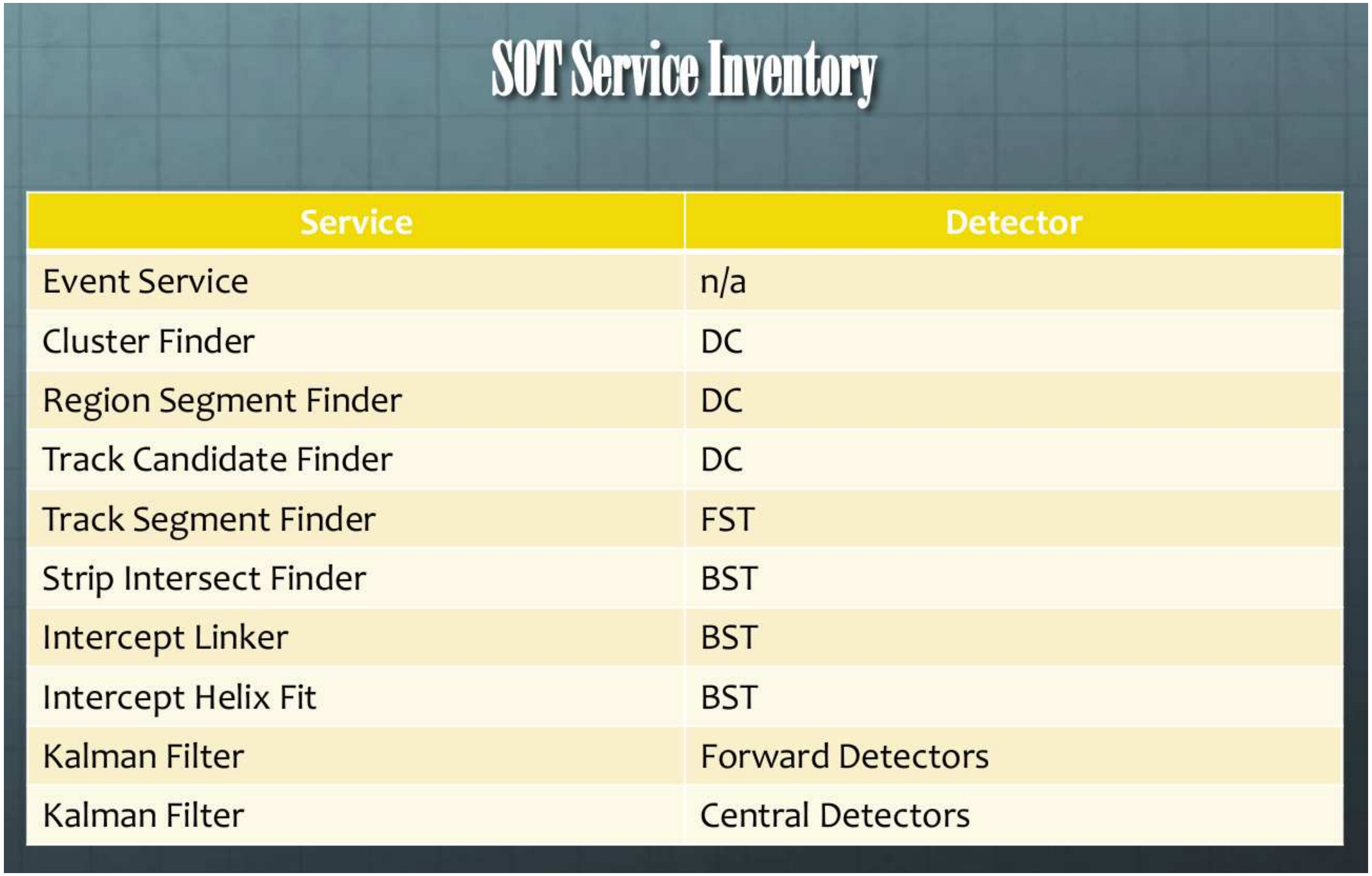


The CLARA project - A Tracking Example



The CLARA project - Status

1. Still under development, but components ready for deployment.
2. Inventory of services (≈ 20 total):



SOT Service Inventory

Service	Detector
Event Service	n/a
Cluster Finder	DC
Region Segment Finder	DC
Track Candidate Finder	DC
Track Segment Finder	FST
Strip Intersect Finder	BST
Intercept Linker	BST
Intercept Helix Fit	BST
Kalman Filter	Forward Detectors
Kalman Filter	Central Detectors

The CLARA project - Why Should You Care?

1. Modularity - 1: Data access only within local service. You can’t hammer someone else’s data structures.
2. Modularity - 2: Communication done only by passing data through a well-defined interface.
3. Modularity - 3: Encourages users to ‘Divide and conquer’.
4. Load balancing: Multithreaded so it can respond to bottlenecks by spawning additional threads or processes.
5. Speed on a modern computer: Easy to scale up and takes full advantage of distributed, multi-core, multi-threaded environment.
6. Technology independent: Multilingual.

Summary and Conclusions

1. Must be able to analyze data on Day 1 of 12 GeV Era.
   - latency issue.
   - computing requirements.
2. Software review
   - no show stoppers
   - concerns about staffing, management.
3. CLARA
   - opportunity to apply what we learned from CLAS6
   - need Collaboration to start production running.

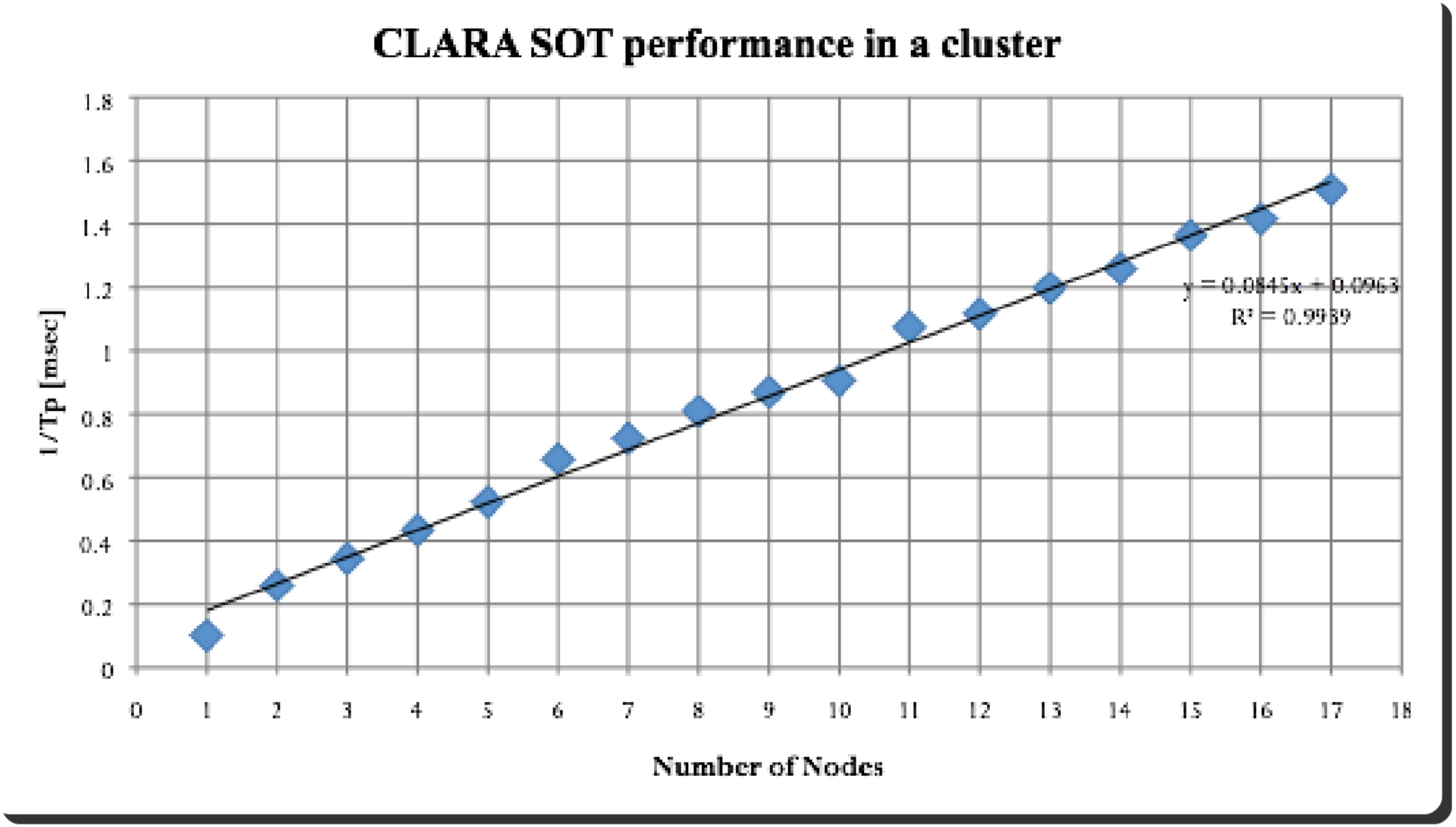
Software planning document in preparation.

svn co `https://clas12svn.jlab.org/repos/docs/Software/Vision/`

Additional Slides.

The CLARA project - Status

Scaling up in a cluster.



Accuracy

- Testing Procedures.
  - Analyze a standard data set and produce standard histograms.
  - Compare with reference histograms and send alarms when necessary.
  - Much of the infrastructure already exists, but is not used.
- Software Reviews
  - Establish a group of subsystem coordinators that will check/vet new code.
  - Coordinators will sign-off on changes to the software for major releases.

Agility

- Multi-lingual, multi-platform.
- Can take advantage of distributed, multi-core, multi-thread technologies.
- Loosely coupled components/services - CLARA
  - Physics applications communicate with the interface.
  - Interface manages processing.
  - Fully multi-threaded and multi-core.
  - Richmond cluster test results.
  - Functioning tracking services.
  - Modular
- List of services?? Production use?

Efficiency - Management - 1

- Software Developers
  - Infrastructure, CLARA, database structure
  - Subsystems developers: tracking, calibration, ...
  - Users: physics reconstruction and final analysis.
- Software Subsystem Coordinators
  - Come from the hardware groups.
  - Leader and mentor to users and developers for a subsystem.
  - Responsible for code quality and documentation.
  - Group of Subsystem Coordinators vetts code, presents results.
  - Regular software presentations at Collaboration meeting.

Efficiency - Management - 2

Other ideas to consider:

- Regular working-group style presentations at Collaboration meetings.
- Rewarding subsystem developers: high-end workstations and monitors.
- Funding from JSA/SURA for software projects.
- Cooking coordinator position - term position that will coordinate all farm-based reconstruction.
- Analysis using unreviewed software must be reviewed by physics working groups before approval to show results.
- Create analysis working group equivalent to physics working groups.

Efficiency - Management - 2

Other ideas to consider:

- Regular working-group style presentations at Collaboration meetings.
- Rewarding subsystem developers: high-end workstations and monitors.
- Fundi
- Cooki at will coordinate all farm-based reconstruction.
- Analysis using unreviewed software must be reviewed by physics working groups before approval to show results.
- Create analysis working group equivalent to physics working groups.

Need feedback from Collaboration!

Software Subsystem Groups

- Infrastrcture
- Calibrations
- Simulations
- Reconstruction
- Corrections
- Analysis
- Service

Summary

- We must be able to calibrate and analyze from turn-on.
- Software will continue to be reviewed and assessed like hardware.
- A broad, as yet incomplete, software plan is being developed.
- Time to pitch in.

CLAS12 Software Review, May, 2011

Information Technology for the 12 GeV Era - Internal Review
May 20, 2011
Thomas Jefferson National Accelerator Facility, Newport News, VA

CIO Message

Colleagues,

The one day internal CIO review of most aspects of Information Technology that impact preparations for and initial running of the 12 GeV science program will take place Friday, May 20. This review is intended to get a good understanding of progress towards IT in the12 GeV era, and discover if there are areas that might need increased effort in the coming year. Thank you in advance for working this important area of the Lab. Please let me know if you have any questions or comments.

Thanks,
Roy

Review Panel:

Chip Watson, Chair (JLab, Deputy CIO)
Graham Drinkwater (ATG)
Richard Jones (UConn, GlueX)
Cortney Carpenter (W&M, CIO)
Brad Sawatzky (Hall C IT)
Karl Slifer (UGBOD IT rep)

Report

Hall B Findings and Recommendations

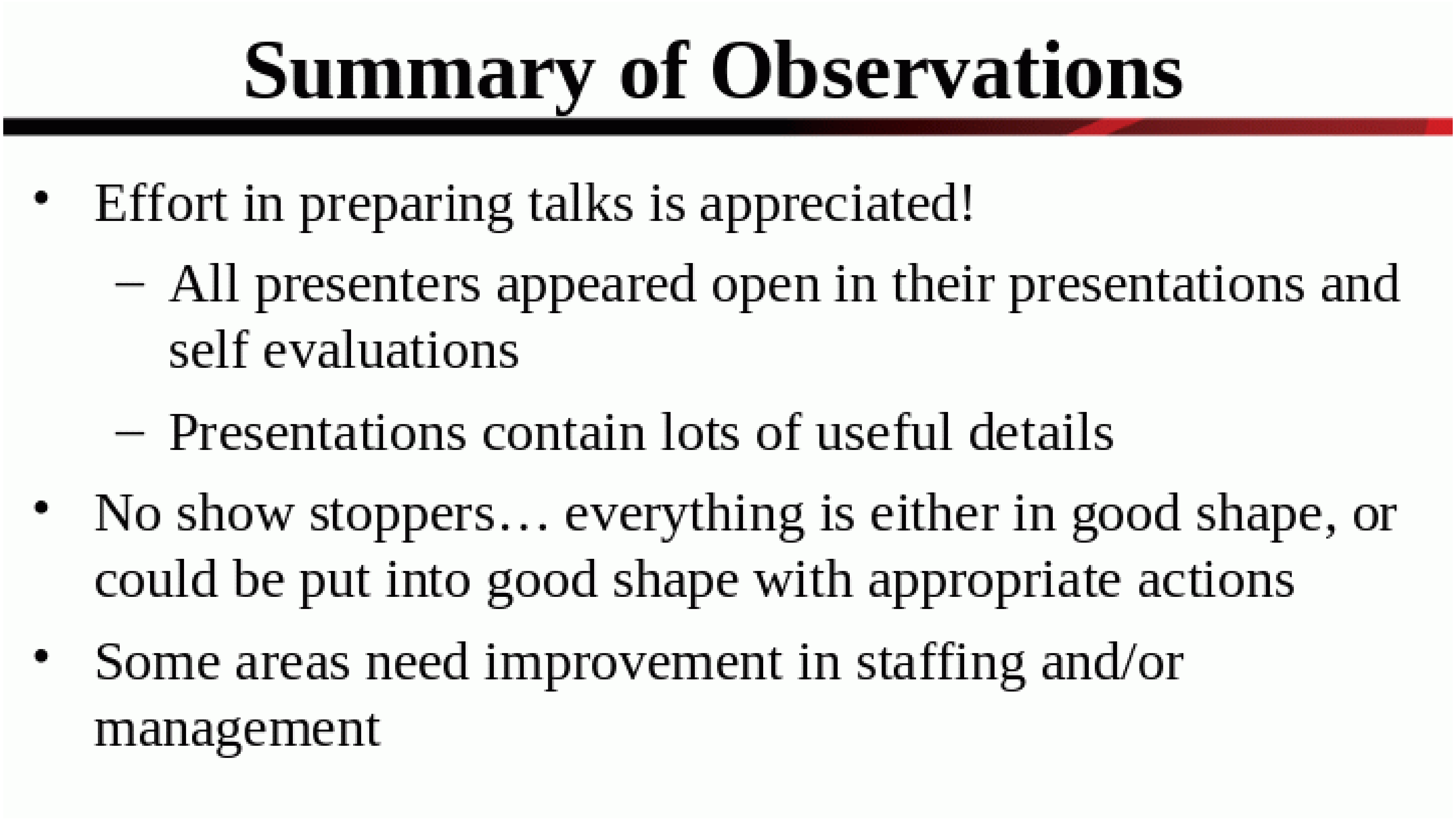
- ‘Their plans and progress generally appear to be in good shape, however manpower was noted as a potential concern (in particular, there is a fairly small group of core software developers).’
- With regard to CLARA the committee had questions about reliability. ‘It was not clear if the security, bandwidth, latency, uptime, and other associated issues have been fully considered.’.

  Response: Redundant resources that users start on their own machines.
- ‘running CLARA with components distributed across a wide area network is not likely to be a particularly high performance choice’.

  Response: Processing will routinely be done on ‘local’ machines though some resources (*i.e,*, geometry service) may be downloaded from a remote source at startup.
- ‘the cost of disk and tape will be non-negligible, and the presented plan does not yet include keeping a duplicate of raw data.’

  Response: Will now include the cost of keeping a copy of the data.

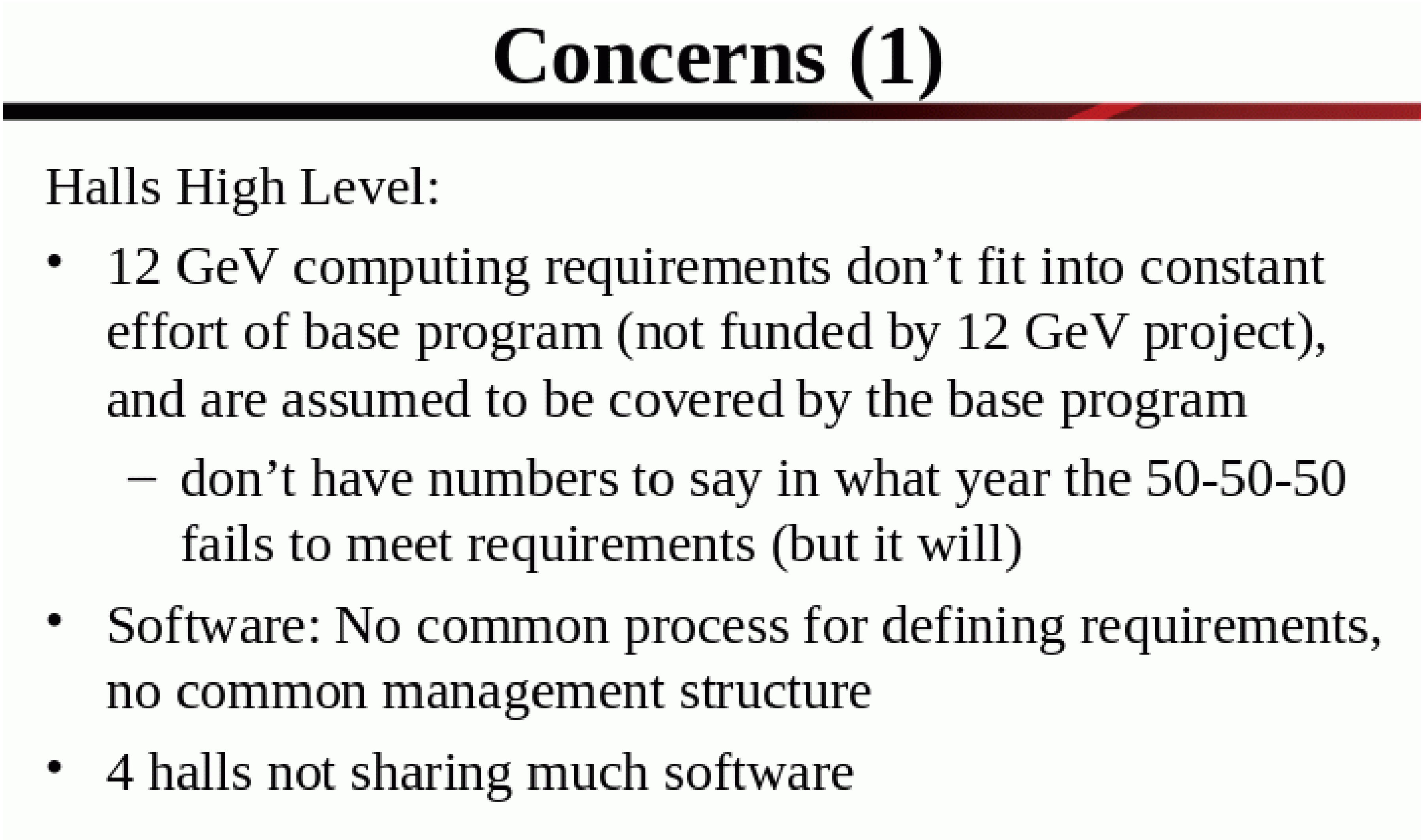
Closeout - 1



Summary of Observations

- Effort in preparing talks is appreciated!
  - All presenters appeared open in their presentations and self evaluations
  - Presentations contain lots of useful details
- No show stoppers… everything is either in good shape, or could be put into good shape with appropriate actions
- Some areas need improvement in staffing and/or management

Closeout - 2

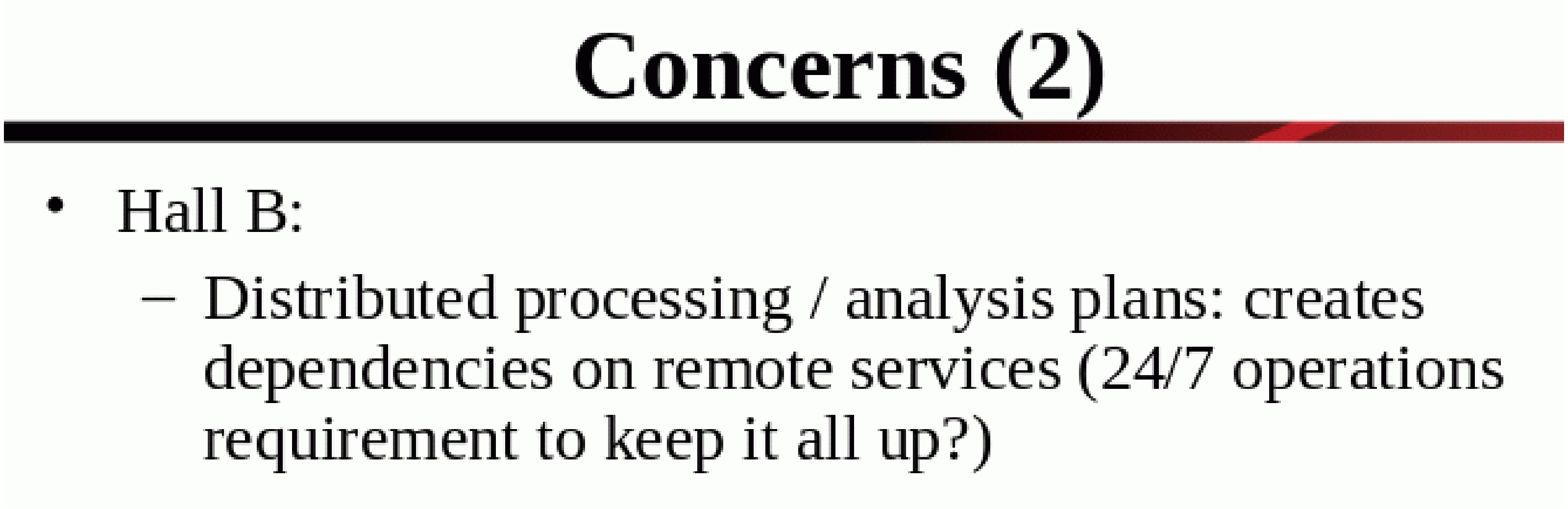


Concerns (1)

Halls High Level:

- 12 GeV computing requirements don’t fit into constant effort of base program (not funded by 12 GeV project), and are assumed to be covered by the base program
  - don’t have numbers to say in what year the 50-50-50 fails to meet requirements (but it will)
- Software: No common process for defining requirements, no common management structure
- 4 halls not sharing much software

Closeout - 3



Concerns (2)

- Hall B:
  - Distributed processing / analysis plans: creates dependencies on remote services (24/7 operations requirement to keep it all up?)