SoLID Software: Responses to Recommendations

Ole Hansen

Jefferson Lab

SoLID Collaboration Meeting
December 3, 2016

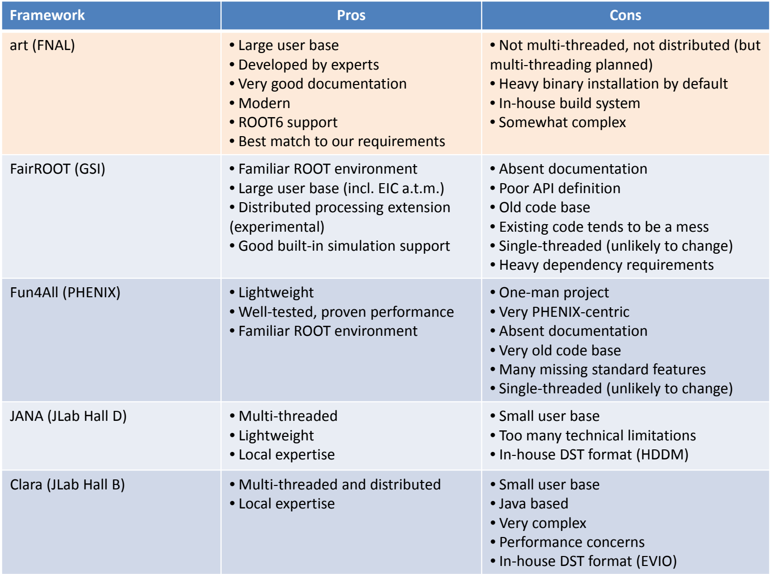
Responses I: “End-to-End” Framework

- Efficient approach: adopt an existing framework
- Developed set of requirements
- Evaluated 6 candidate frameworks. Extensive list of pros/cons
- *art* framework from Fermilab appears most suitable
- Testing and prototyping underway
- High-level task list developed
- Aim to have usable version ready by mid-2017

Framework Requirements

- Consistent environment for simulation, digitization, reconstruction and physics analysis ("end-to-end")
- Must support multi-pass processing (persistent data objects). Strongly prefer standard file format/persistence model (ROOT)
- Should support multiple processing chains per job
- Must have option to output ROOT files directly usable for interactive analysis
- Should support data provenance tracking (metadata generation and passthrough)
- Must be ready for or directly support parallel/distributed processing
- Must be readily available at this time

Frameworks Pros/Cons



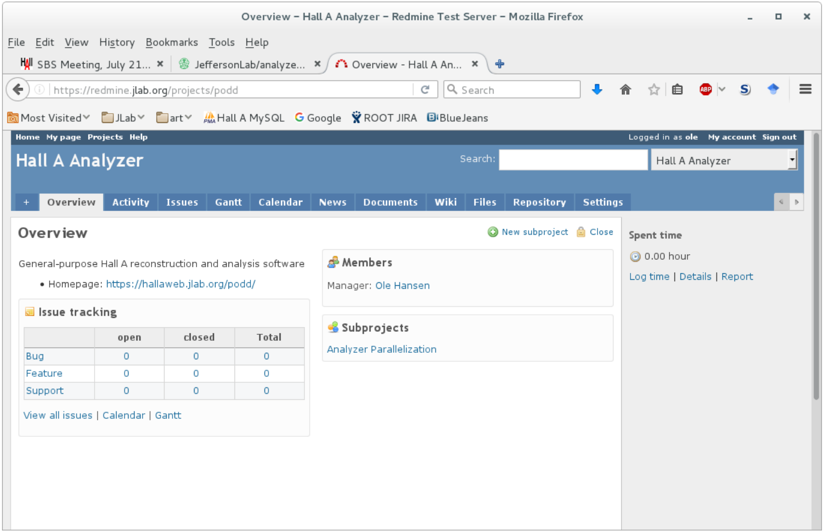
Framework	Pros	Cons
art (FNAL)	• Large user base • Developed by experts • Very good documentation • Modern • ROOT6 support • Best match to our requirements	• Not multi-threaded, not distributed (but multi-threading planned) • Heavy binary installation by default • In-house build system • Somewhat complex
FairROOT (GSI)	• Familiar ROOT environment • Large user base (incl. EIC a.t.m.) • Distributed processing extension (experimental) • Good built-in simulation support	• Absent documentation • Poor API definition • Old code base • Existing code tends to be a mess • Single-threaded (unlikely to change) • Heavy dependency requirements
Fun4All (PHENIX)	• Lightweight • Well-tested, proven performance • Familiar ROOT environment	• One-man project • Very PHENIX-centric • Absent documentation • Very old code base • Many missing standard features • Single-threaded (unlikely to change)
JANA (JLab Hall D)	• Multi-threaded • Lightweight • Local expertise	• Small user base • Too many technical limitations • In-house DST format (HDDM)
Clara (JLab Hall B)	• Multi-threaded and distributed • Local expertise	• Small user base • Java based • Very complex • Performance concerns • In-house DST format (EVIO)

NB: Also evaluated Hall A analyzer (Podd), but rejected due to one-pass-only design

Software Milestones

- Draft software design document (by end of 2016)
- Create documentation wiki to collect numerous existing documents (by end of 2016)
- Set up task/issue tracking system (Redmine?)
- Port existing simulations to *art* (aiming for spring 2017, but big job)
- Start broader adoption by collaboration hopefully by summer 2017. This will obviously be an early, incomplete version of the software. Timing is aggressive.

Improving Project Management



Overview – Hall A Analyzer – Redmine Test Server – Mozilla Firefox
File Edit View History Bookmarks Tools Help
SBS Meeting, July 21...
JeffersonLab/analyze...
Overview - Hall A An...
https://redmine.jlab.org/projects/podd
Search
Most Visited
JLab
art
Hall A MySQL
Google
ROOT JIRA
BlueJeans
Home My page Projects Help
Logged in as ole My account Sign out
Hall A Analyzer
Search:
Hall A Analyzer
+ Overview Activity Issues Gantt Calendar News Documents Wiki Files Repository Settings
New subproject
Close

Overview

General-purpose Hall A reconstruction and analysis software

- Homepage: https://hallaweb.jlab.org/podd/

Issue tracking

	open	closed	Total
Bug	0	0	0
Feature	0	0	0
Support	0	0	0

View all issues | Calendar | Gantt

Members

Manager: Ole Hansen

Subprojects

Analyzer Parallelization

Spent time

0.00 hour

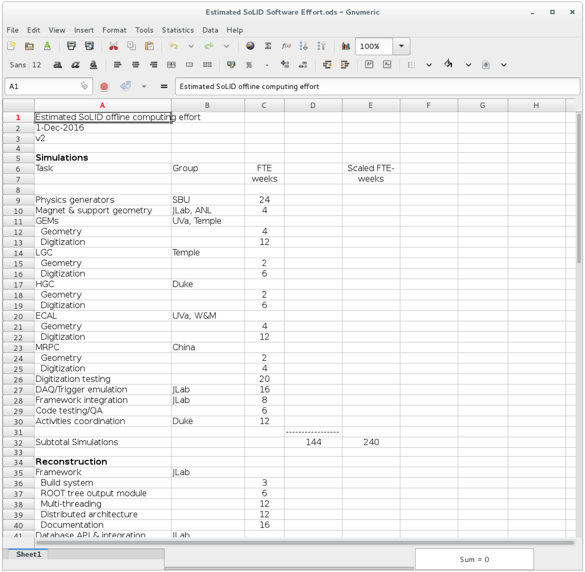
Log time | Details | Report

Responses II: Software Manpower/Resources

- Developed detailed list of software tasks with time estimates
- Compared estimates with those published by GlueX (in 2013)
- SoLID estimate is roughly half of that of GlueX: 22 vs. 42 FTE-years
- Differences largely understood

SoLID Software Manpower Estimate

`https://hallaweb.jlab.org/12GeV/SoLID/download/doc/Estimated_SoLID_Offline_Effort.xlsx`


Estimated SoLID Software Effort.ods – Gnumeric
File Edit View Insert Format Tools Statistics Data Help
100%
Sans 12
Sheet1
Sum = 0

A1 = Estimated SoLID offline computing effort

	A	B	C	D	E	F	G	H
1	Estimated SoLID offline computing effort							
2	1-Dec-2016							
3	v2							
4								
5	**Simulations**							
6	Task	Group	FTE		Scaled FTE-			
7			weeks		weeks			
8								
9	Physics generators	SBU	24					
10	Magnet & support geometry	JLab, ANL	4					
11	GEMs	UVa, Temple						
12	Geometry		4					
13	Digitization		12					
14	LGC	Temple						
15	Geometry		2					
16	Digitization		6					
17	HGC	Duke						
18	Geometry		2					
19	Digitization		6					
20	ECAL	UVa, W&M						
21	Geometry		4					
22	Digitization		12					
23	MRPC	China						
24	Geometry		2					
25	Digitization		4					
26	Digitization testing		20					
27	DAQ/Trigger emulation	JLab	16					
28	Framework integration	JLab	8					
29	Code testing/QA		6					
30	Activities coordination	Duke	12					
31				----------------				
32	Subtotal Simulations			144	240			
33								
34	**Reconstruction**							
35	Framework	JLab						
36	Build system		3					
37	ROOT tree output module		6					
38	Multi-threading		12					
39	Distributed architecture		12					
40	Documentation		16					
41	Database API & integration	JLab						

Software Manpower: Comparison with GlueX

Task Group	Labor estimate (FTE-weeks)		Main reasons for difference
	GlueX[1]	SoLID[2]	
Simulation	192	240	Simulation to be integrated into framework.
Reconstruction	787	355	Adoption of existing framework. Re-use of algorithms. Smaller number of subsystems.
Calibration	275	103	Smaller number of subsystems.
Production	275	155	Standard data format. Re-use of workflow tools.
Analysis	275	100	No PWA analysis and no grid implementation of analysis.
Data Challenges	62	23	No PWA data challenge.
Totals	1866	976	

[1] https://halldsvn.jlab.org/repos/trunk/docs/offline/ProjectProgress/OfflineComputingActivities2013.xlsx

[2] https://hallaweb.jlab.org/12GeV/SoLID/download/doc/Estimated_SoLID_Offline_Effort.xlsx

Director's Review Recommendations III: Data Handling

- **Finding:** "Early exploration of the tools available at Jefferson Lab that can handle the data at the expected scale of SoLID will be crucial in minimizing false starts in software development."
- **Recommendation:** "Closer communication with the other JLab experiments and the JLab computing center is strongly encouraged."

Responses III: Data Handling

- We have been in active communication with the JLab computer center regarding future computing needs for SoLID. Based on current trends, handling of data volumes at the expected scale of SoLID, *viz.* 5-10 PB/year, is already fully managable at JLab today and will likely be routine at the time SoLID runs.
- We are investigating the suitability of the existing JLab workflow management tools (SWIF) for SoLID computing.
- Substantial data for GlueX have just begun to arrive. CLAS12 is expected to go into production mode in 2018. Further, the Hall A SBS program, which will also produce multi-PB data sets, will commence in 2019. The experiences of these groups, as they emerge, will inform future decisions we may have to make for SoLID software development.
- In the long run, it would be beneficial if SoLID software supported distributed and/or grid computing. We will keep this option in mind. Any advanced data processing capabilities would be developed in close collaboration with the computer center and the other halls, who are already exploring massively parallel approaches.