

Hall-D Online Status



David Lawrence JLab

June 22, 2018

RootSpy Refactored to Use xMsg

cMsg

- designed for lighter weight messaging, not the volume and message sizes we use in RootSpy
- all messages go through central server

xMsg

- Google Protocol Buffers + zeroMQ
- messages are direct connections (central server just provides connection info)

Thomas Jefferson National Privacy and Security Notice

Data Acquisition Support

Main Menu

- [Welcome](#)
- [About Us](#)
- [Login](#)

Documentation and downloads

- [Introduction](#)
- [Front end hardware](#)
- [Software subsystems](#)
- ▾ [Core Packages](#)
 - [Event IO \(evio\)](#)
 - [Event Transfer \(ET\)](#)
 - [Experiment Control System \(Afecs\)](#)
 - ▾ [Messaging \(cMsg\)](#)
 - [Doxygen for C](#)
 - [Doxygen for C++](#)
 - [javadoc](#)
 - [CODA 2.6 release](#)
 - [CODA 3 release](#)
 - [Downloads](#)
 - [Documents](#)

[◀ Go to previous page](#) [Go to next page ▶](#)

Messaging (cMsg)

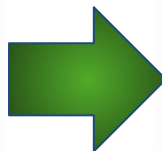
Page summary:
Documentation for the cMsg publish and subscribe messaging system

Current version - 3.6

cMsg is a publish-subscribe, interprocess messaging system developed at Jefferson Lab. At the most basic level it is an API for sending and receiving messages. This API is used to wrap a variety of communication protocols, and is most often used to access a popular and useful, full publish-subscribe messaging system. The package is supported in C, C++ and java on Linux. All online CODA components use cMsg to communicate control information (not data) to each other.

Files associated with this page:

Attachment	Size
Readme file	7.96 KB
Change log	1.15 KB
cMsg User's Guide	751.25 KB
cMsg Developer's Guide	537.85 KB



GlueX-Collaborat... 20180622_lawre... GlueX Monitoring... Main page | xMsg... David


Not Secure <https://claraweb.jlab.org/xmsg/>

Documentation Download About

xMsg


Multilingual publish-subscribe messaging system

[Get Started](#)




xMsg represents many-to-many communication model

xMsg creates an environment where various information producers and consumers can communicate all at the same time. Each piece of information can be delivered to various consumers concurrently, while each consumer can receive information from different producers.



Space decoupling

xMsg actors, i.e. information producers and consumers do not need to know each other. Message addressing is based on the message topic. Information is published to a topic or named logical channel. Consumers will receive all messages published to the topic to which they subscribe, and all subscribers to a topic will receive the same message. The producer is responsible for defining classes of message topics to which consumers can subscribe.



Time decoupling

xMsg actors do not have to be actively participating in the information handling at the same time. Information delivery is mediated through the xMsg proxy. Even though synchronization among information processing actors in most cases is not required, xMsg presents synchronized communication channels as well.

© Copyright 2017, Vardan Gyurjyan. Generated using [mkdpl](#).

Time Series Database

(Grafana + InfluxDB)



Mass fit results



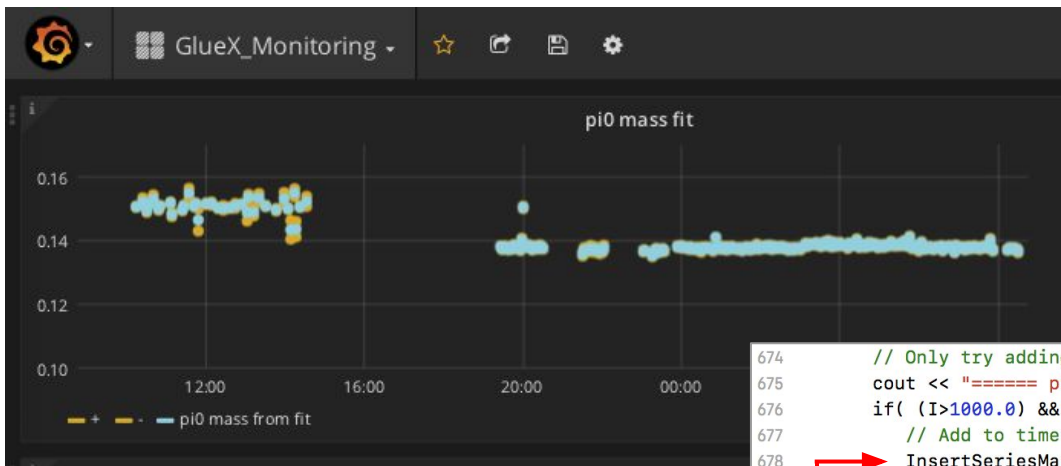
Number of particles per 1k L1 triggers
(π^0 , ρ , ω , ϕ)



Number of particles per 1k PS
coincidences $> 7\text{GeV}$
(π^0 , ρ , ω , ϕ)



Hits per event by detector



code from root macro used by monitoring

```

674 // Only try adding to time series if we have more than 20 particles in peak
675 cout << "==== pi0: I="<<I<< " mean: " << pars[1] << " +/- " << errs[1] << " sigma: "<< p
676 if( (I>1000.0) && (errs[1]<0.1*pars[1]) && (errs[2]<0.2*pars[2]) ){
677 // Add to time series
678 InsertSeriesMassFit("pi0", pars[1], pars[2], errs[1], errs[2], unix_time);
679
680 // per 1k triggers
681 if(Ntrig>0.0){
682 stringstream ss;
683 ss << "fit_stats,ptype=pi0 ";
684 ss << "rate_per_1ktrig="<<rate_per_1ktrig;
685 ss << ",rate_per_1kps="<<rate_per_ps;
686 ss << ",counts="<<I;
687 ss << ",Ntrig_phys="<<Ntrig_phys;
688 ss << ",Ntrig_ps="<<Ntrig_ps;
689 ss << ",Nps="<<Nmy_ps;
690 if(unix_time!=0.0) ss<<" "<<(uint64_t)(unix_time*1.0E9); // time is in units of ns
691 InsertSeriesData( ss.str() );
692 }
693
694 // Optionally reset the histogram so next fit is independent of this one
695 if(rs_GetFlag("RESET_AFTER_FIT")) {
696 rs_ResetHisto("/highlevel/TwoGammaMass");
697 PIDNorms->SetBinContent(NORM_pi0_trig, Ntrig_tot);
698 PIDNorms->SetBinContent(NORM_pi0_ps , Nps);
699 }
700 }

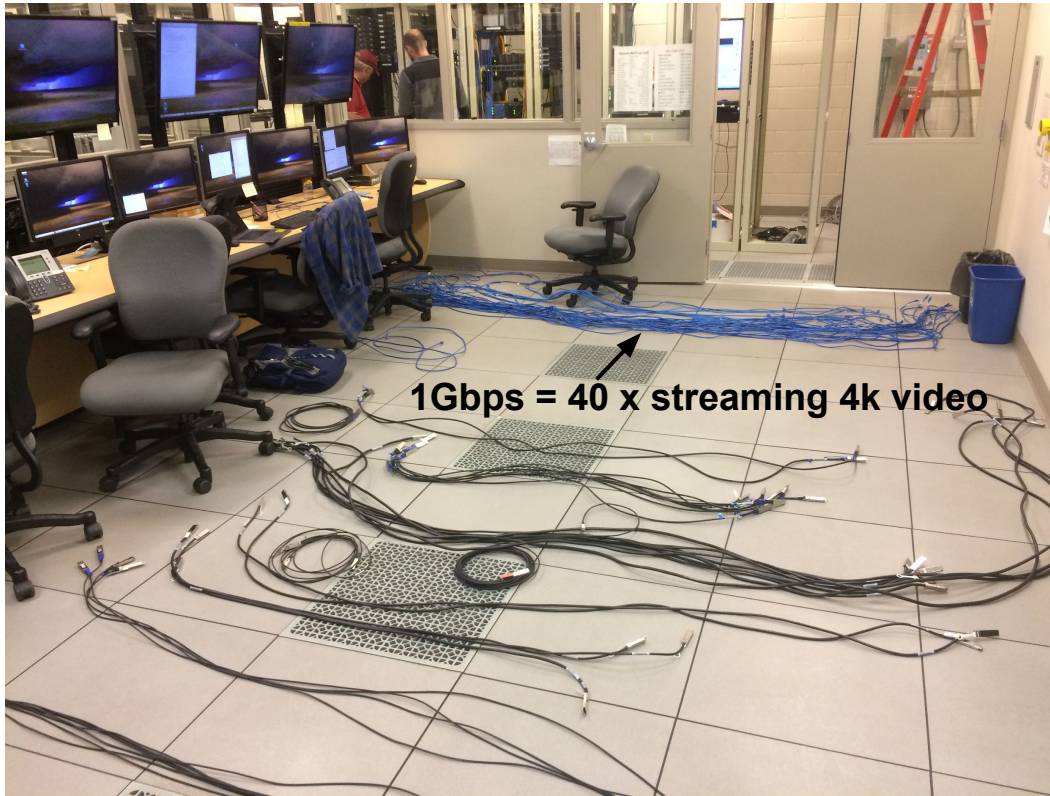
```

- Values may be added to time series DB from online macros using special RootSpy calls
- No special preparation of tables in DB is needed a priori
- Special RSTimeSeries program is run in background that handles this independent of RootSpy GUI

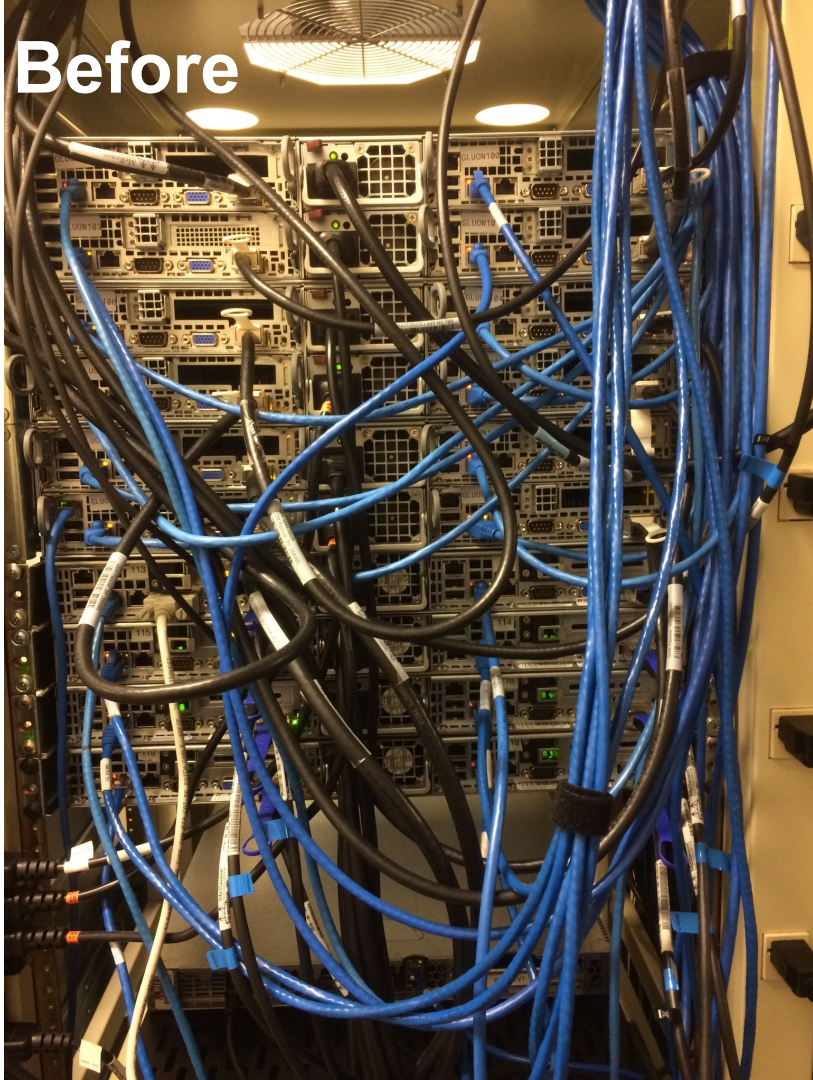
RootSpy Family of Programs

hd_ana	Farm monitoring processes that produce histograms
RootSpy	GUI program shift-takers use to view live histograms/macros
RSelog	Program launched from GUI to make e-log entry into HDMONITOR
RSArchiver	Started by DAQ to continuously rewrite ROOT file with cumulative histograms. File is source of <i>ver00 RootSpy</i> in Plot Browser
RSTimeSeries	Started by DAQ to continuously gather statistics and run macros in background to write to time series DB

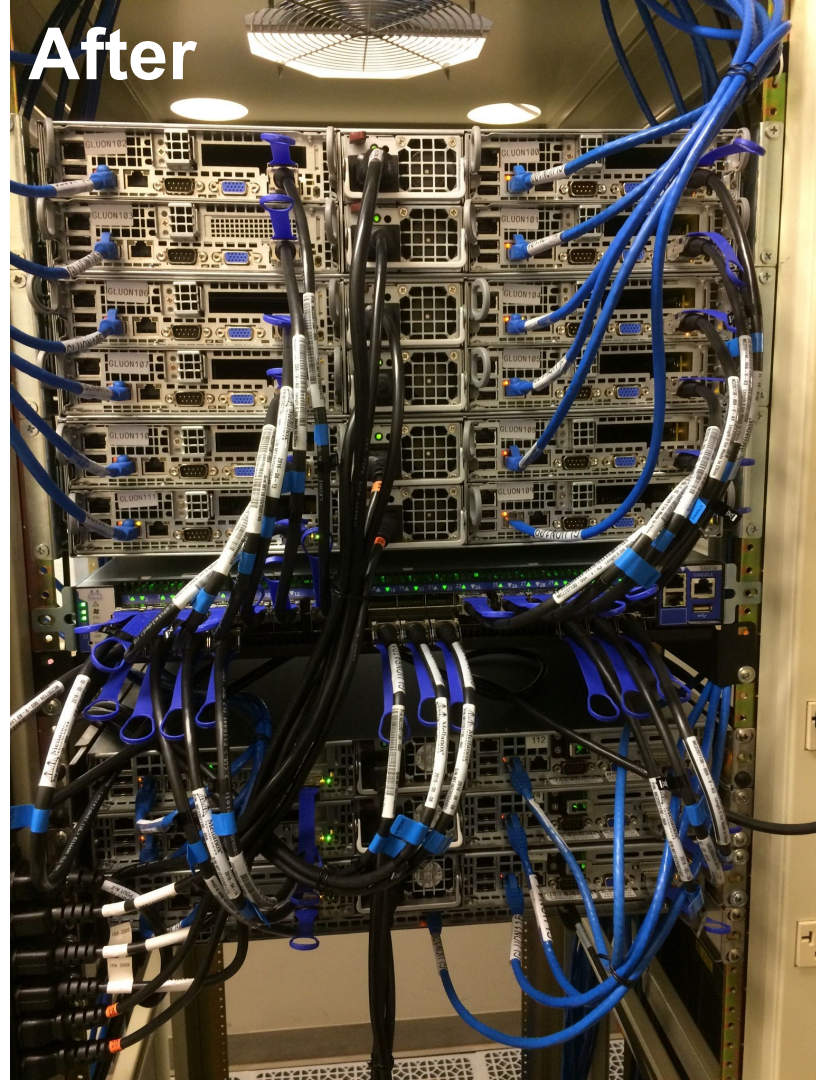
Hall-D Control Room Networking/Gluon Cleanup and DAQ Hardening



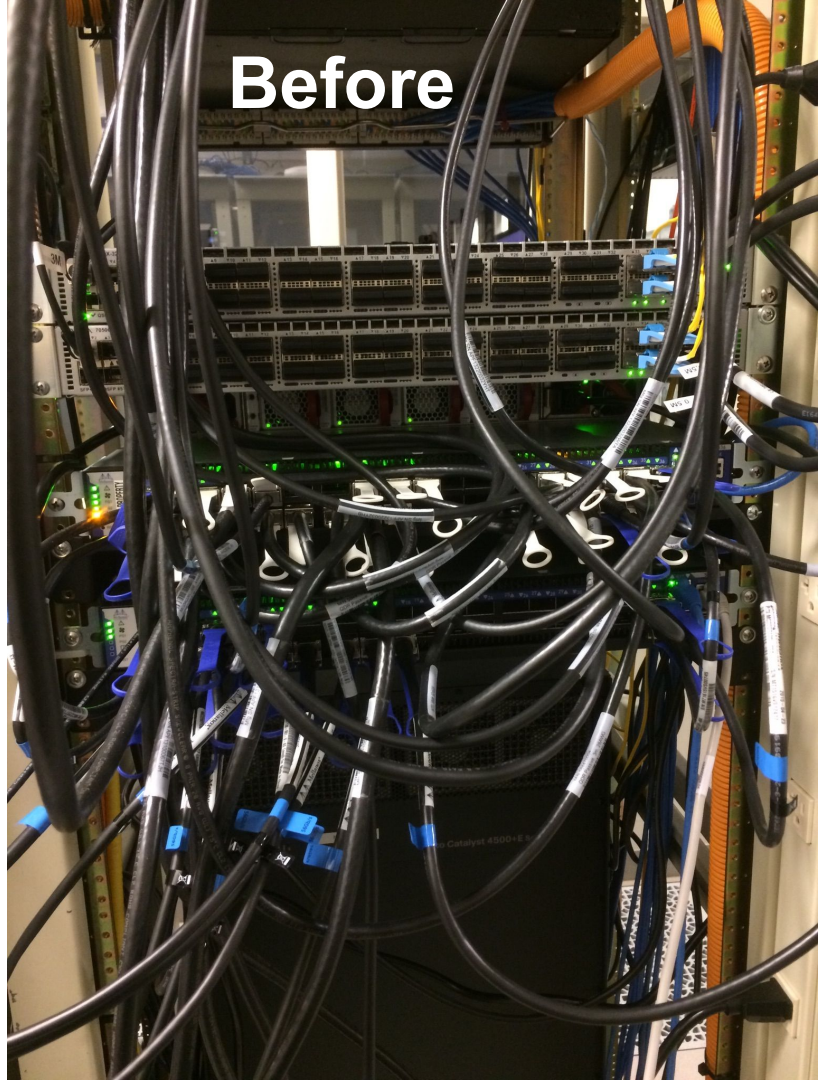
Before



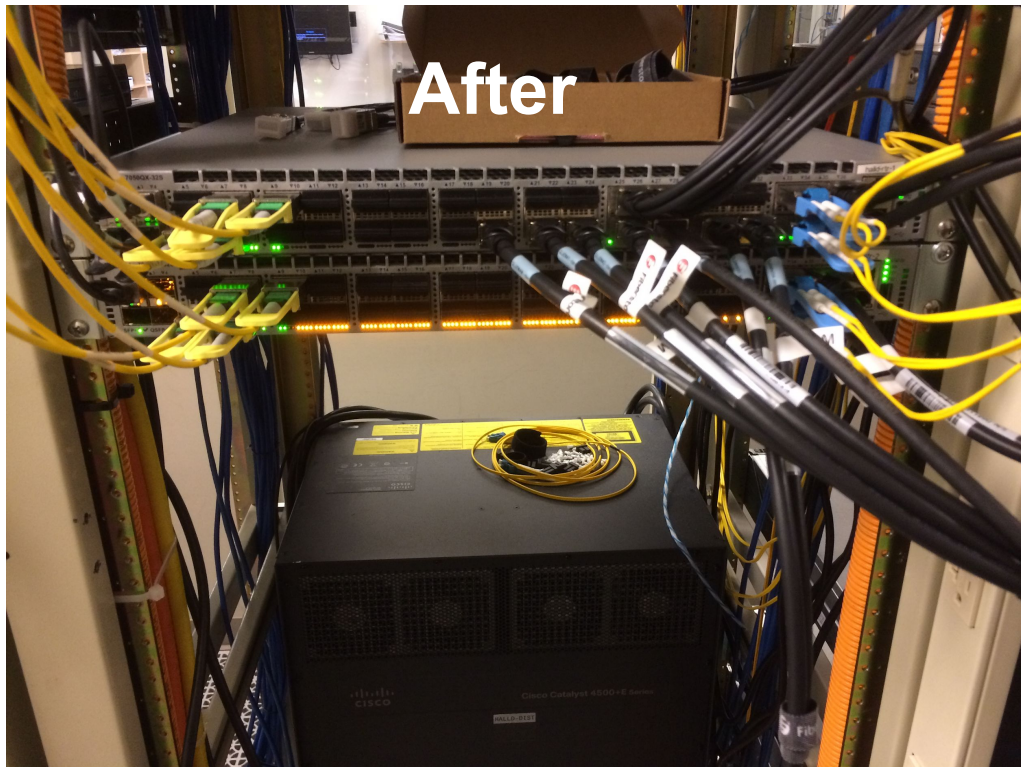
After



Before



After





Labels!



Labels!



Welcome

7

**RED HAT
ENTERPRISE LINUX**

Counting
House
Desktops
upgraded to
RHEL7

Specifications for DAQ and Data Recording

(High Intensity Running)

	Proposal Avg.	Proposal Peak	Current Plan
Beam	5×10^7 γ /s	1×10^8 γ /s	5×10^7 γ /s
Trigger	100 kHz	200 kHz	90 kHz
Front End	1.5 GB/s	3 GB/s	1.2 GB/s
Disk	150* MB/s	300* MB/s	600 GB/s
Tape	1.5* PB (GlueX II) 1.7* PB (GlueX + DIRC) 3.2* PB (Total)	3.0* PB (GlueX II) 3.3* PB (GlueX + DIRC) 6.3* PB (Total)	25 PB (GlueX II) 27 PB (GlueX + DIRC) 52 PB (Total)

** Assumes L3 trigger reduces data by factor of 10*

GlueX II : PR12-13-003 200 PAC Days
 GlueX + DIRC : C12-12-002 220 PAC Days