

Controls

EPICS Archiver Appliance Spring 2013

Our top 5 Objectives

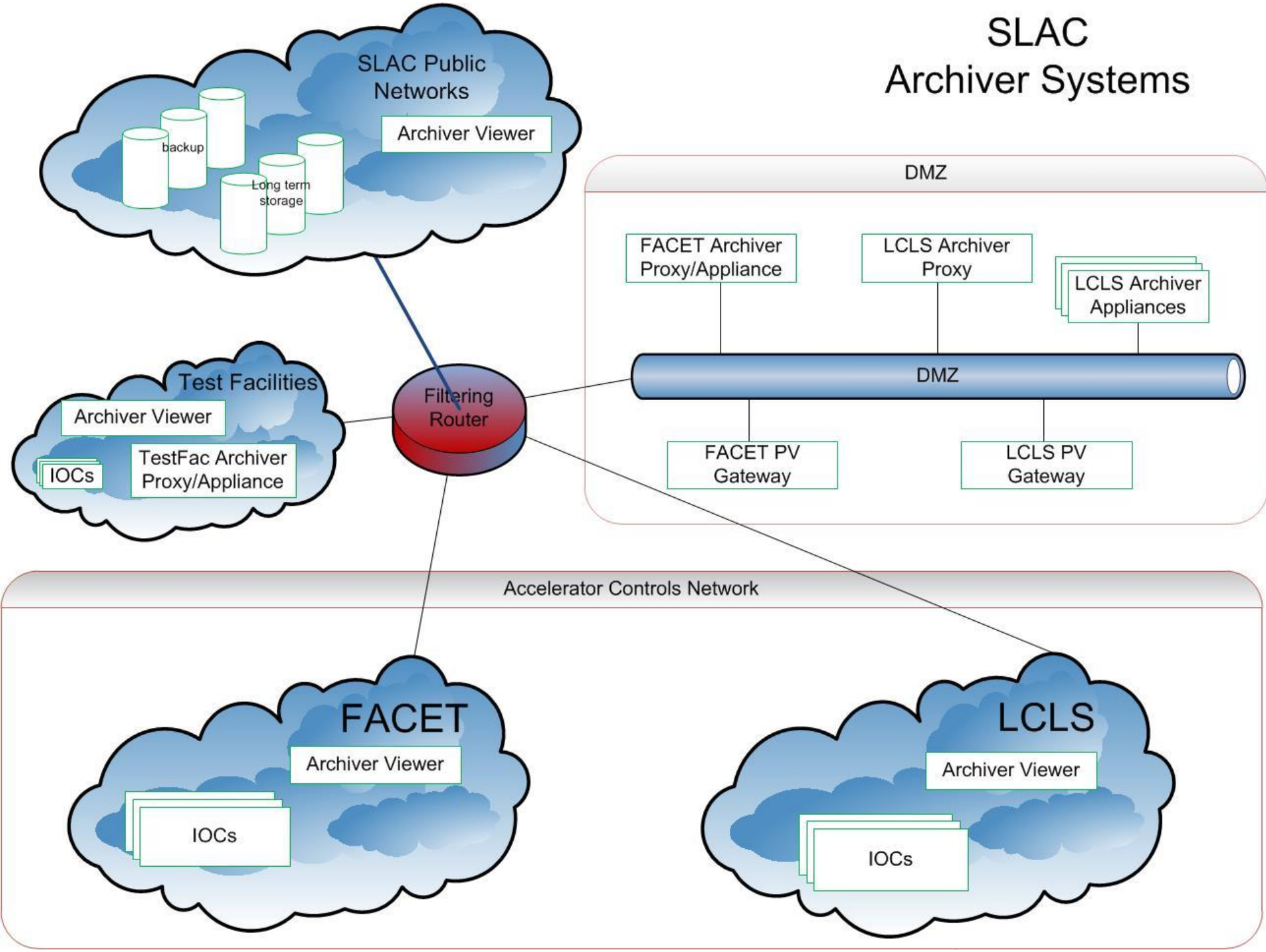
- Scale to 1-2 millions PV's
- Fast data retrieval
- Users add PV's to archiver
- Zero oversight
- Flexible configurations on a per PV basis

Status

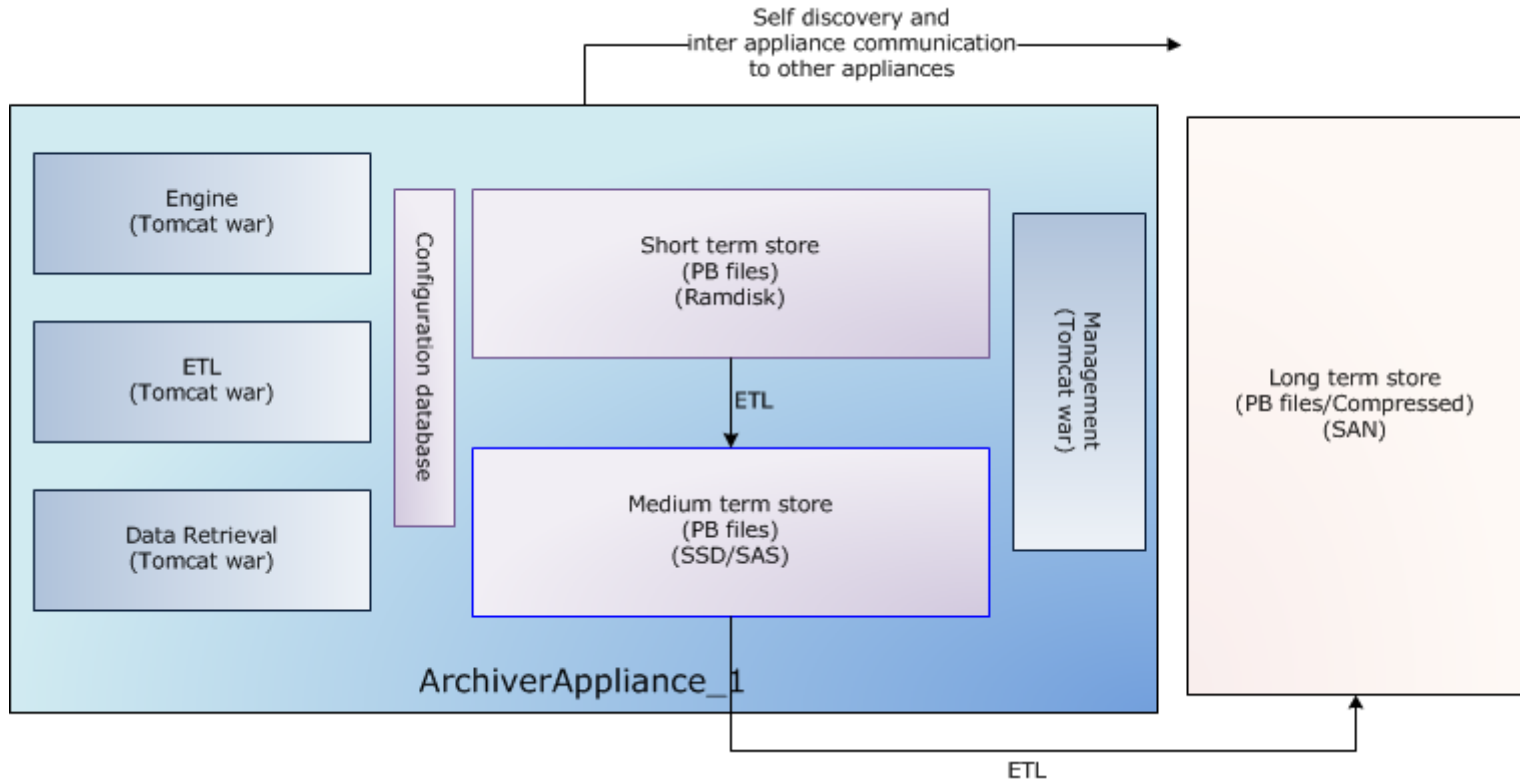
Controls

- In production
 - Small accelerator program called TestFac
 - Accelerator R&D like Xband
 - Running for a month
 - 24903 PVs
 - 0.2GB/day
 - 25.34 bytes per event
- FACET/LCLS/LCLS2 next
 - LCLS on our test stands – 16G/day
- Targeted at machine physicists and operators

SLAC Archiver Systems

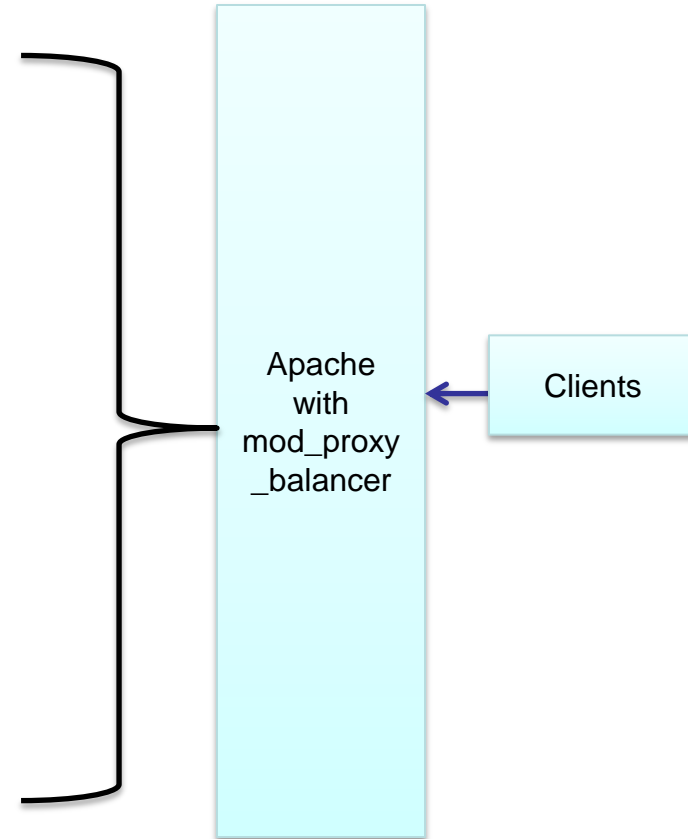
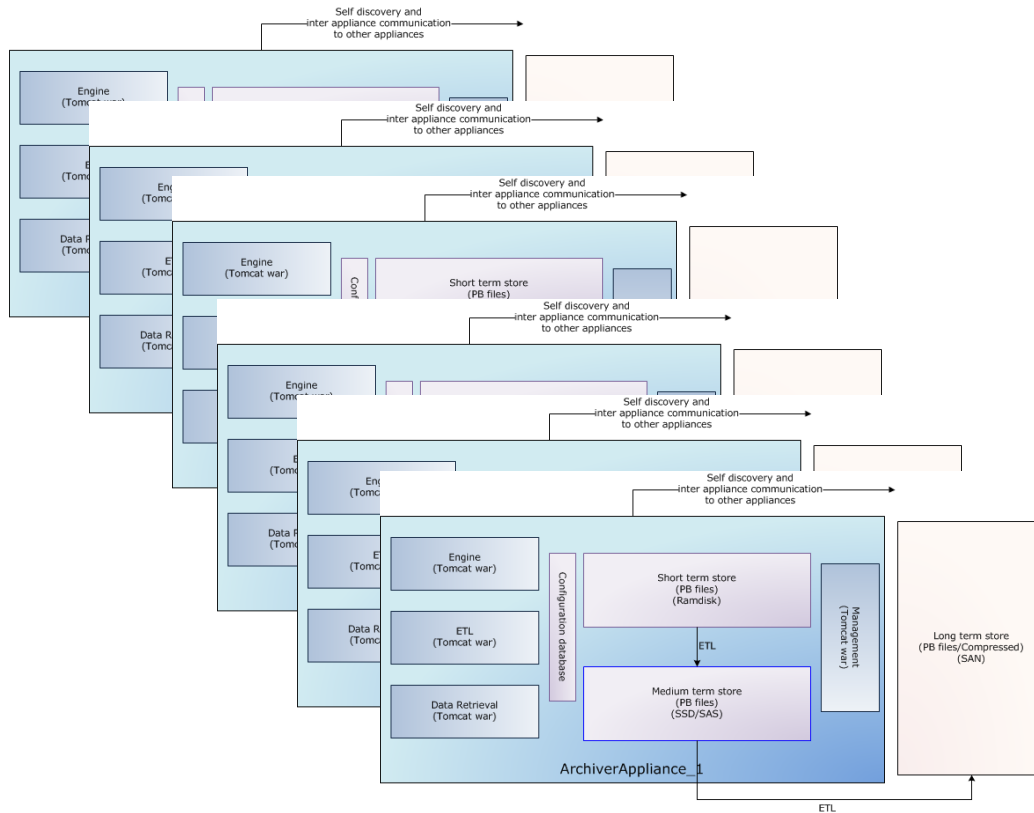


Components



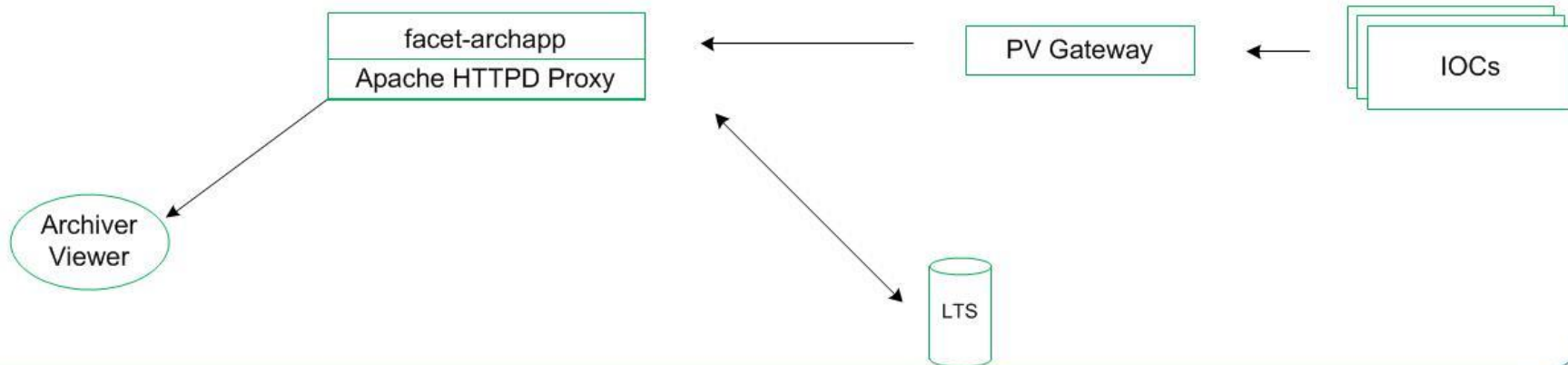
Scale by clustering appliances

Controls

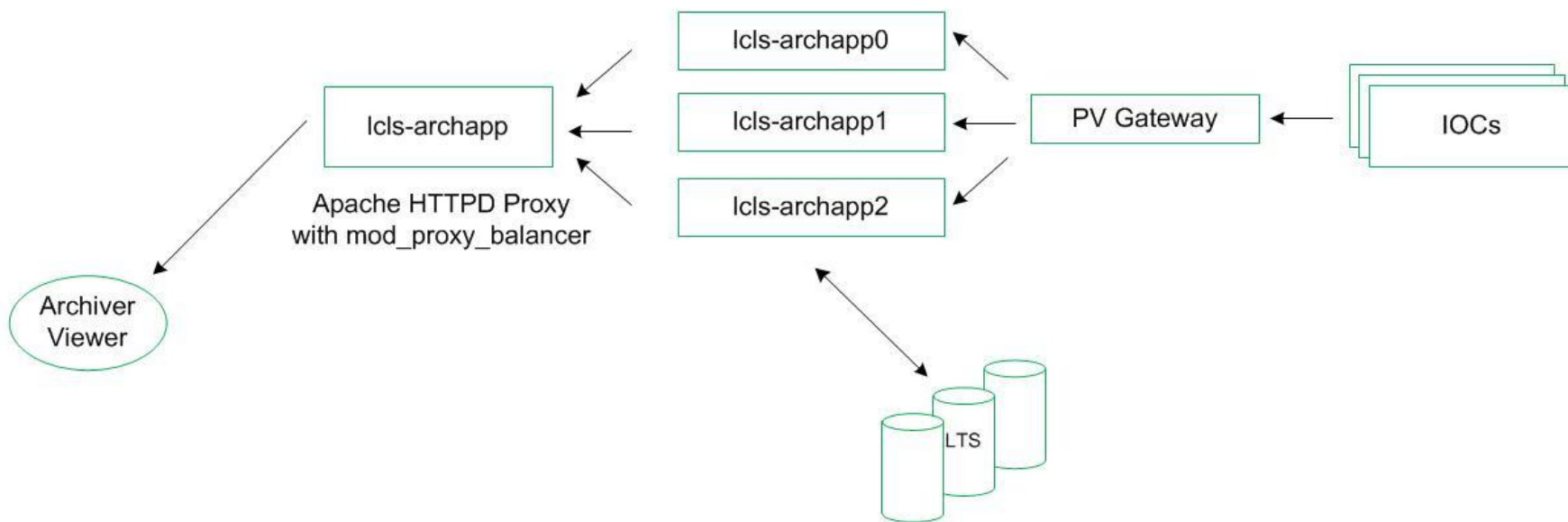


Archiver System Deployment in SLAC

Archiver System with Single Applinace



Archiver System in a Cluster Mode



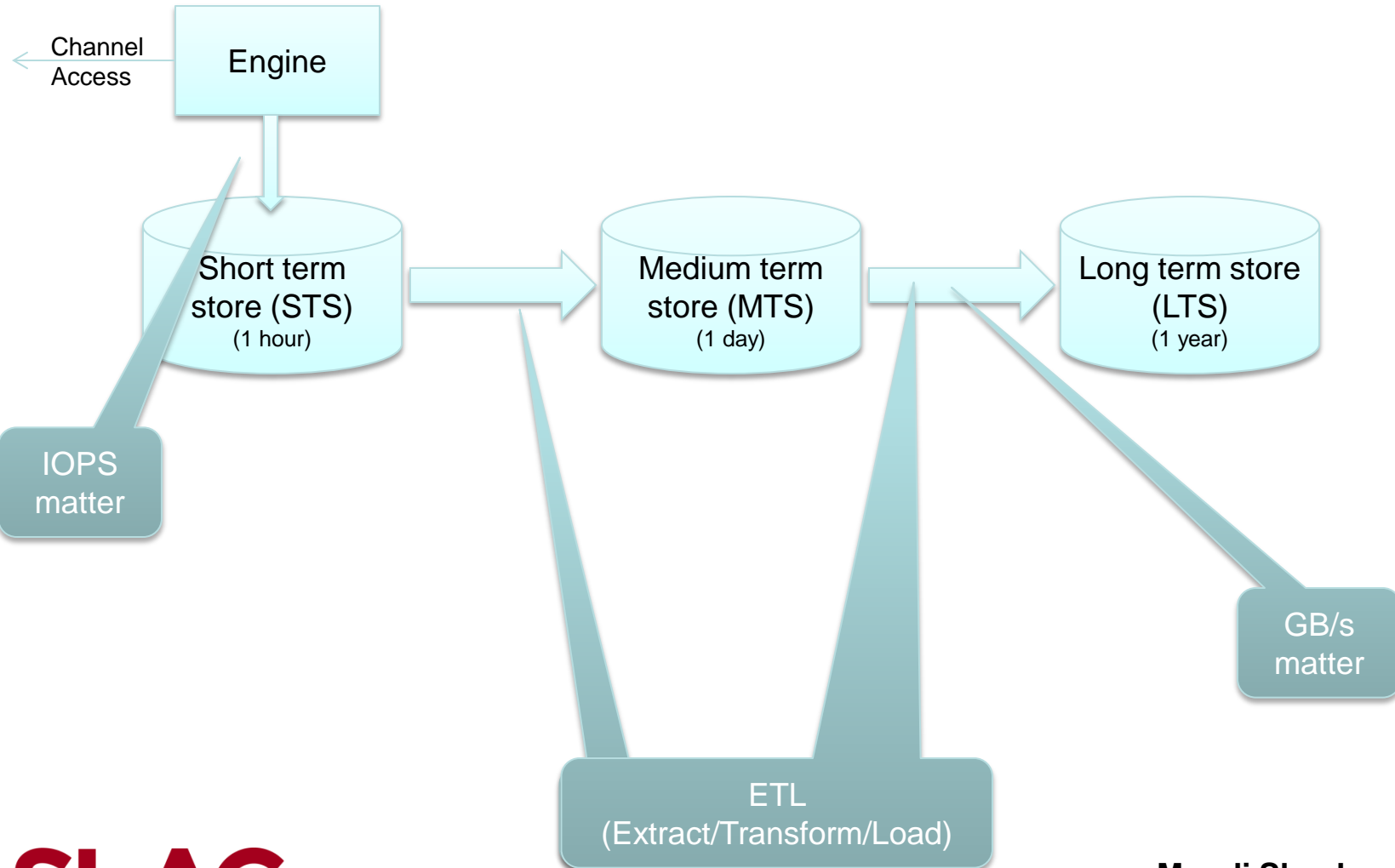
SLAC specific constraints

Controls

- We “buy” storage from IT
 - RAIDed for redundancy
 - Tape backup
 - Lots of support/infrastructure
 - Located on their network

Controls

Multi stage



Controls

Keep track of capacity

25	Page 1 of 1			
Instance Name	Status	PV Count	Event Rate	Data Rate (GB/year)
test-arch	Working	116878	3,541.82	2,366.43

Write thread
secondsToBuffer=10
==> writing at a
rate of 70K/sec

Time consumed by Writing sampleBuffer to PB (in seconds)	0.49
Total number of ETL(0->1) runs so far	23
Time spent in ETL(0->1) (s)	122
Percentage of time spent in ETL(0->1)	3.5

Each stage
is measured

Total number of ETL(1->2) runs so far	3
Time spent in ETL(1->2) (s)	125
Percentage of time spent in ETL(1->2)	0.47

Capacity planning

- Use measured performance of each stage to compute capacity consumed
- Use a minimax algorithm to automatically assign PVs to appliances
 - Find resource (stage) with most utilization
 - Use appliance with most available capacity in this resource.

Users add PVs

EPICS Archiver Appliance for Test Facilities

[Home](#) [Reports](#) [Metrics](#) [Storage](#) [Appliances](#) [Integration](#)

This is the archiver for Test Facilities. If you have any questions, please contact Murali Shankar or Jingchen Zhou.
To check the status of or to archive some PV's, please type in some PV names here.

```
BPMS:XT01:461:X  
BPMS:XT01:461:Y
```

Check Status

Archive

Archive (specify sampling period)

Archive PV workflow

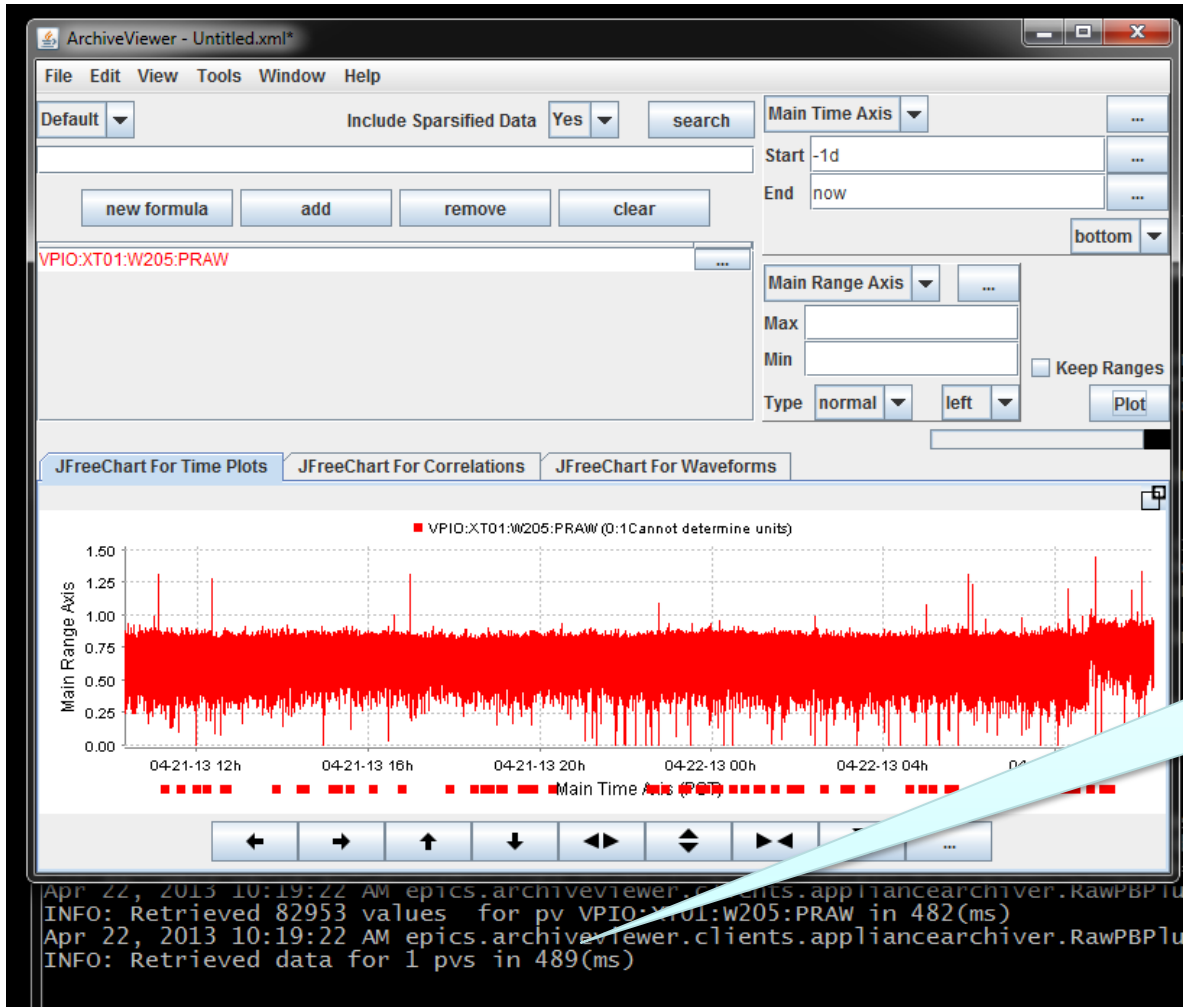
- Gather information about PV
 - Measure event rate and storage rate
 - Use ca_monitor for a minute
 - Additional information
 - Is this an alias?
 - What's the RTYP?
 - Alarm limits, ADEL and much more
- Use this info to make “intelligent” configuration decisions.

- Use policies to decide
 - Monitor/scan?
 - Increase sampling period for large waveforms
 - Skip a stage?
 - Keep a sparsified version of data using avg.
 - Fields (HIHI, LOLO etc) to automatically archive.
- All of this is on a per PV basis.
- Zero oversight

- Clients get data over HTTP
 - PB/RAW – Binary format over HTTP
 - Serialized using Protocol Buffer
 - ArchiveViewer
 - DataBrowser
 - JSON over HTTP
 - Python
 - Matlab over HTTP
 - CSV/XML/SVG/Others

Controls

ArchiveViewer



1 days worth
of 1Hz
DBR_DBL <
500ms

CSS Databrowser

The screenshot displays the CSS Databrowser application window. The interface includes a menu bar (File, Edit, CSS, Window, Help), a toolbar, and a main workspace divided into several panels.

Data Browser Panel:

- URL: `pbraw://testfac-archapp.slac.stanford.edu/retrieval`
- Table:

Name	Description	Key
<code>pbraw://testfac-archapp.slac...</code>	EPICS Archive app	1

Search Panel:

- Pattern: `VPIO:XT01:W205:PRAW`
- Buttons: Search, Add..., Replace search results, Reg.Exp.

Search Results Table:

PV Name	Name
<code>VPIO:XT01:W205:PRAW</code>	<code>pbraw://testfac-archapp.slac.s...</code>

Plot Panel:

- Y-axis: Value 1 (0 to 10)
- X-axis: Time (2013-04-22 10:25:45 to 10:56)
- Trace: `VPIO:XT01:W...` (blue line)

Properties Panel:

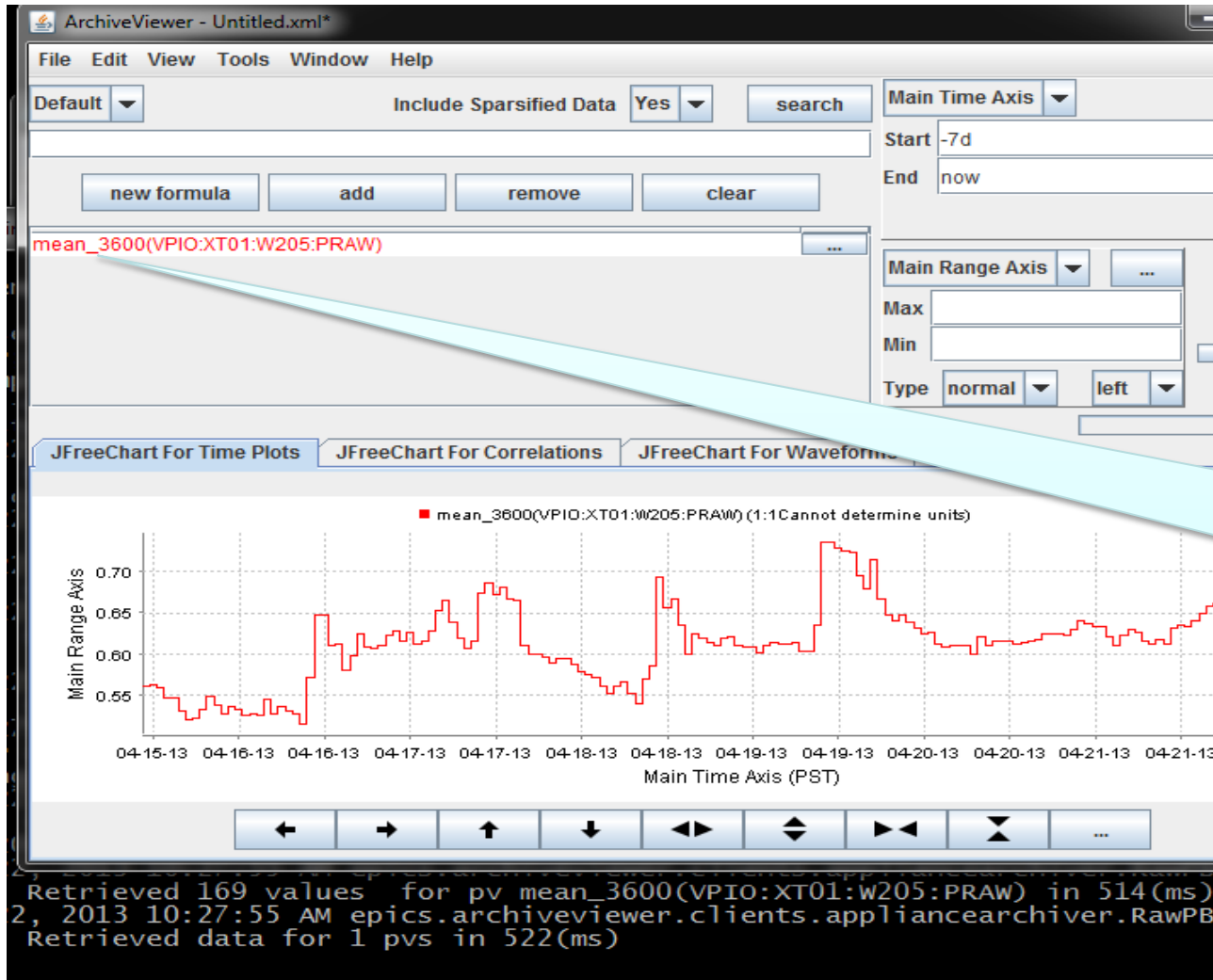
- Buttons: Properties, Export Samples
- Traces: Time Axis, Value Axes, Misc.

Traces Table:

Show	Item (PV, Formula)	Display Name	Col...	St
<input checked="" type="checkbox"/>	<code>VPIO:XT01:W205:PRAW</code>	<code>VPIO:XT01:W205:PRAW</code>		0.

Controls

Sparsification



Compute avg with a bin size of 3600 seconds

- Can be precomputed during ETL and stored
 - If present, we'll use it
 - If not, we'll compute at runtime
 - Precomputation be specified on a per PV basis
 - One can always go back and specify later
- Using Apache Commons Math
 - Average, RMS, Stdz...

- Add PVs using the web based UI
 - Let the archiver compute sampling period+mode
 - IOC engineer specifies this
- UI for monitoring appliances, storage
- UI for ChannelArchiver integration.
- Lots of reports

- Every call that is available to the UI is available for scripting
 - Plus some more.
 - Many examples in Python.
- Serves many purposes
 - Diagnostics/Nagios integration
 - Changes to multiple PVs
 - Specify archiving in IOC INFO fields

Scripting example

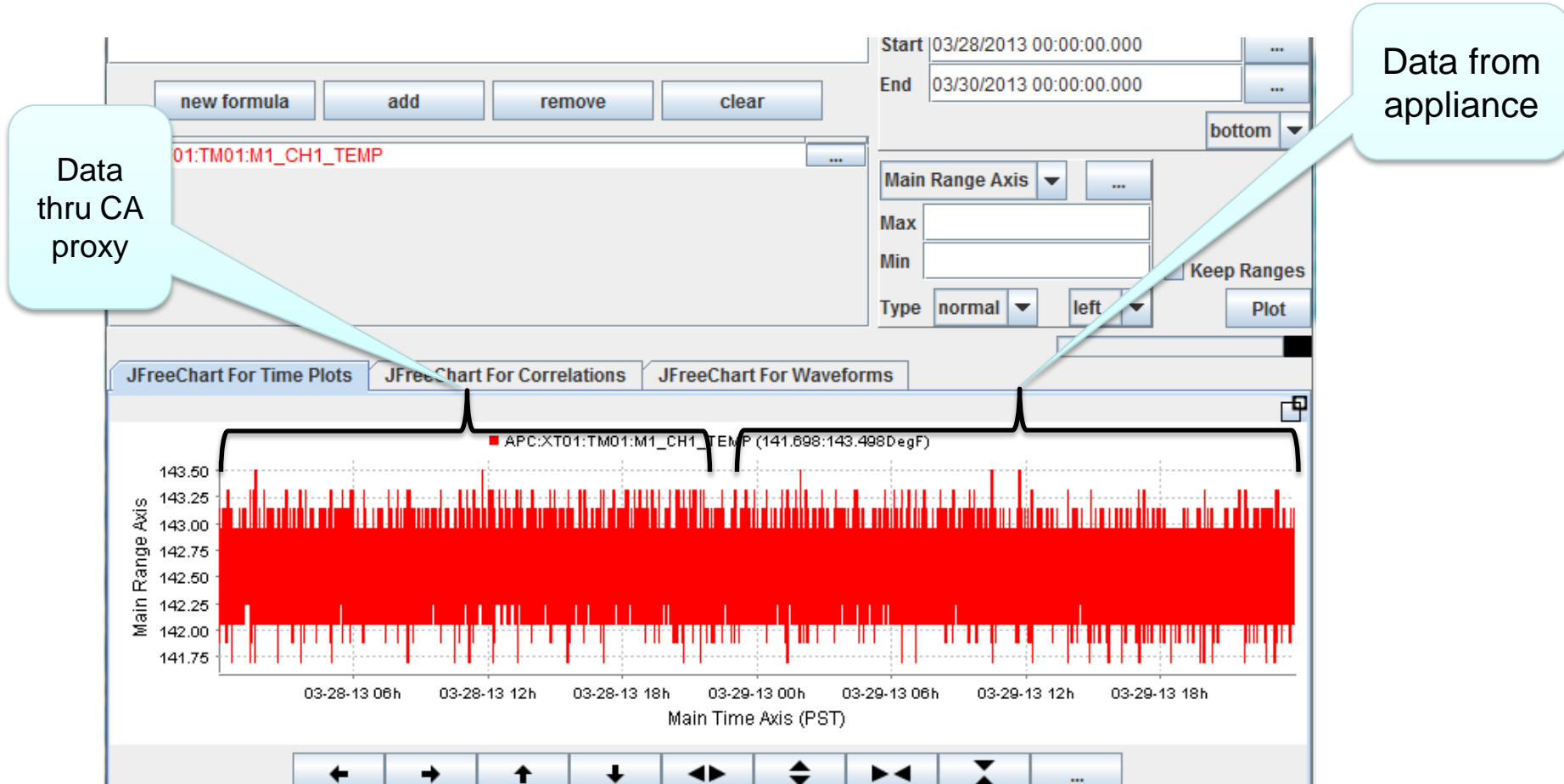
```
#!/usr/bin/env python
import urllib
import json
from lookupIOCEgrForPV import lookupIOCEngineerForPV

httpclient=urllib.HTTPConnection('archiver.appliance.server')
req=httpclient.request('GET', '/mgmt/bpl/getCurrentlyDisconnectedPVs')
resp=httpclient.getresponse()
resptext=resp.read()
allPVs=json.loads(resptext)
for pv in allPVs:
    pvName=pv['pvName']
    lookupIOCEngineerForPV(pvName)
```

- Proxy ChannelArchiver XMLRPC data server
 - No “import” task at installation
- Upload ChannelArchiver XML files
- Facilitates transition from ChannelArchiver.

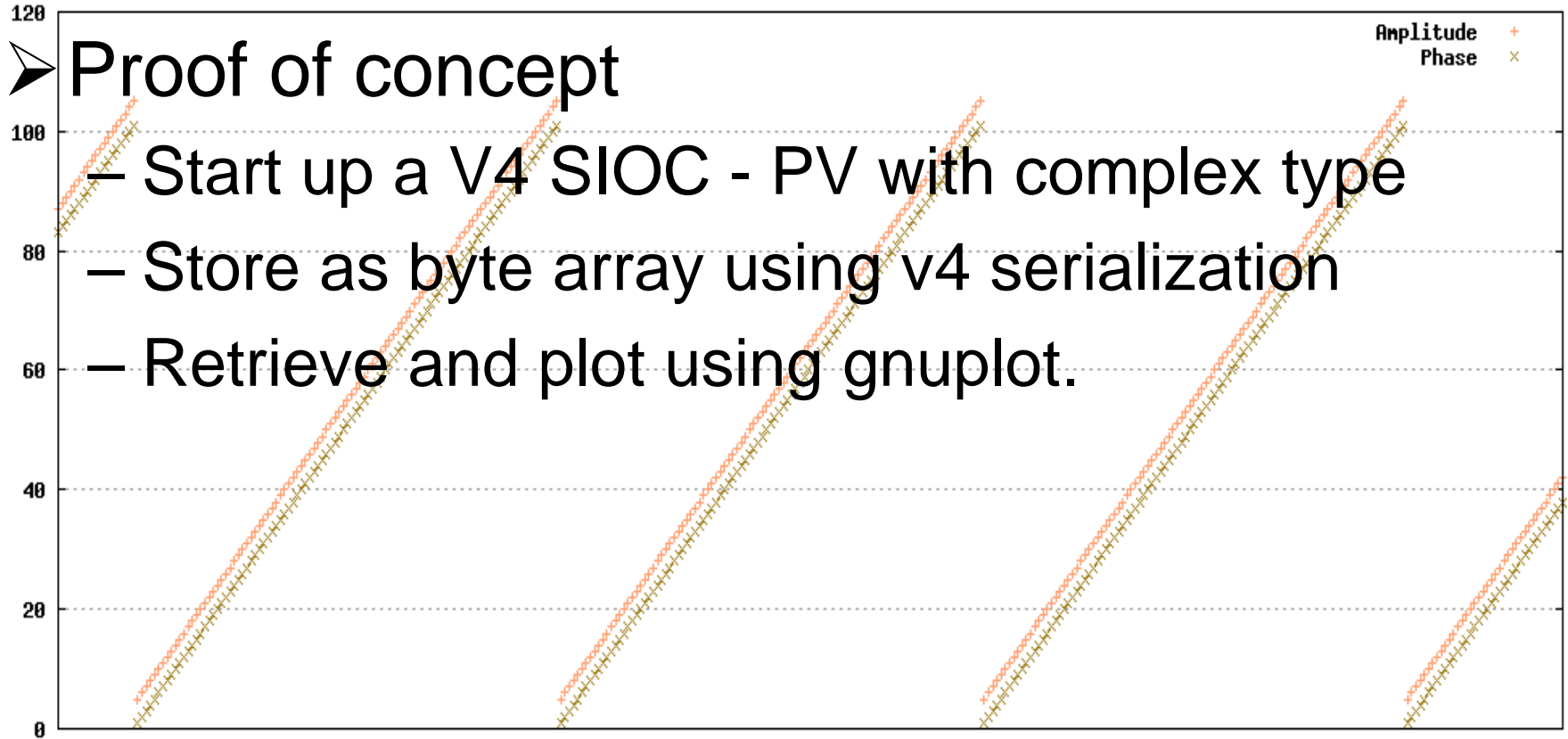
Transparent CA proxy

Controls



- Will definitely not meet everyone's requirements
 - Site specific builds
 - Persistence of configuration
 - Storage
 - Serialization using Google's Protocol Buffers - chunk/PV/time_partition
 - Written on top of Java NIO2
 - Simple interface for other implementations.
 - Alternate type systems

Archiving EPICS V4



Quickstart

Controls

- Easy to try it out
 - Setup EPICS environment variables.
 - Download tar.gz and untar
 - Download tomcat
 - Run shell script

```
sh-4.1$  
sh-4.1$ ./quickstart.sh apache-tomcat-7.0.27.tar.gz
```

- Create appliances.xml
- Create policies.py
- Install script that installs on one appliance
 - Expect most people will develop their own install/upgrade scripts.
 - We have our own install/upgrade scripts.

```
sh-4.1$  
sh-4.1$ ./install_scripts/single_machine_install.sh
```

- Centralized, shared repository
 - Archiver appliance software
 - Required packages (Tomcat, Java etc)
 - Tools/scripts
 - appliances.xml and policies.py for all accelerator programs
- On each appliance
 - Tomcat containers hosting appliance components
 - One container per component
 - `-XX:MaxPermSize=128M -XX:+UseG1GC -Xmx4G -Xms4G -ea`
 - MySQL hosted locally on appliance.

Machines/Storage @ SLAC

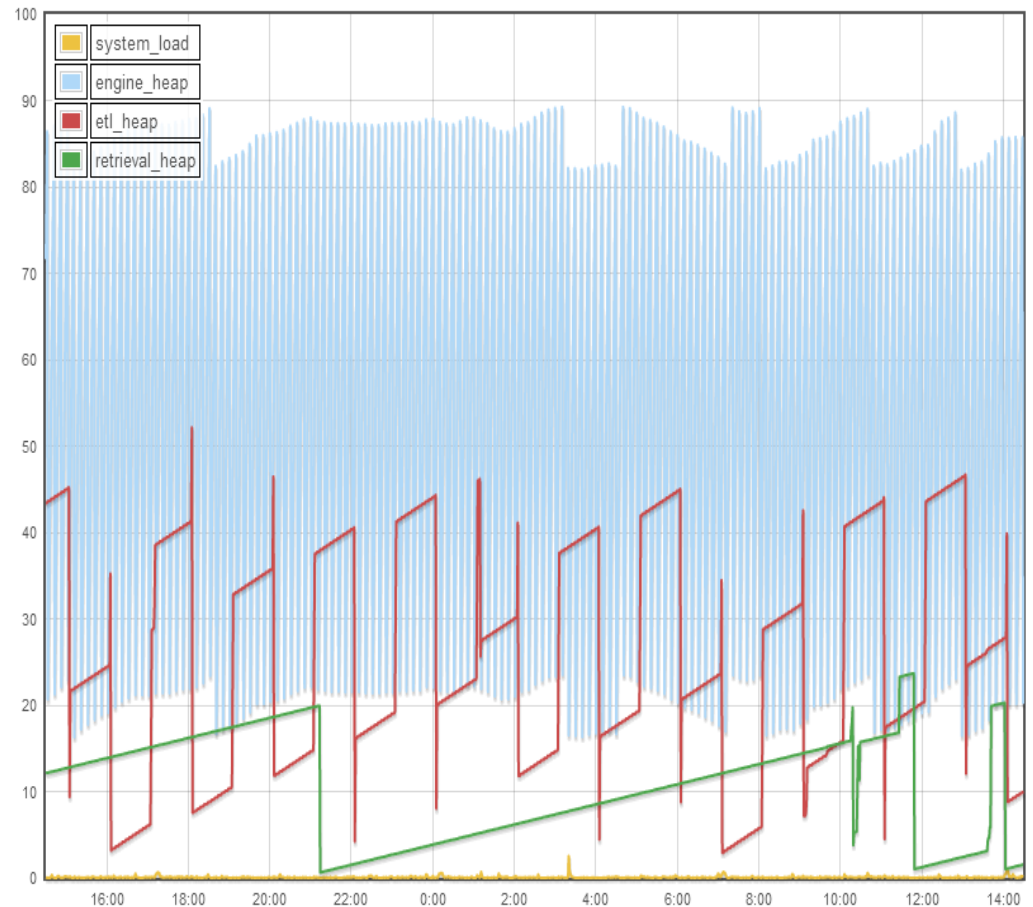
Controls

- Appliances are Dell servers with
 - 128G memory
 - Two hard drives for system OS
 - RAIDed 10k RPM SAS (for MTS)
 - A integrated RAID controller
- Storage
 - STS – System memory – Hourly partitions
 - MTS – Mirrored SAS drives – Daily partitions
 - LTS – ZFS from IT mounted using NFS – Yearly partitions

VM stats

Controls

- `-XX:+UseG1GC`
 - CPU is steady
 - Engine 10%
 - Others 5%



- CSS Databrowser integration
- Support for resharding
 - Move PV between appliances
- Embed JCA/CAJ
- Handle out of space issues gracefully
- Get DESC as part of retrieval request
- Hooks for site specific modifications

- Support for new storage technologies
 - HDF5 – Nikolay Mailtsky (BNL)
 - Object Storage - Amazon Glacier
 - Matlab storage plugin
- Integration with clients.
 - More mime formats
- EPICS V4 support
 - Add support for V4 normative types

- Hosted on sourceforge
- Google EPICS Archiver Appliance
- Try it out with the quickstart

Controls

Questions
