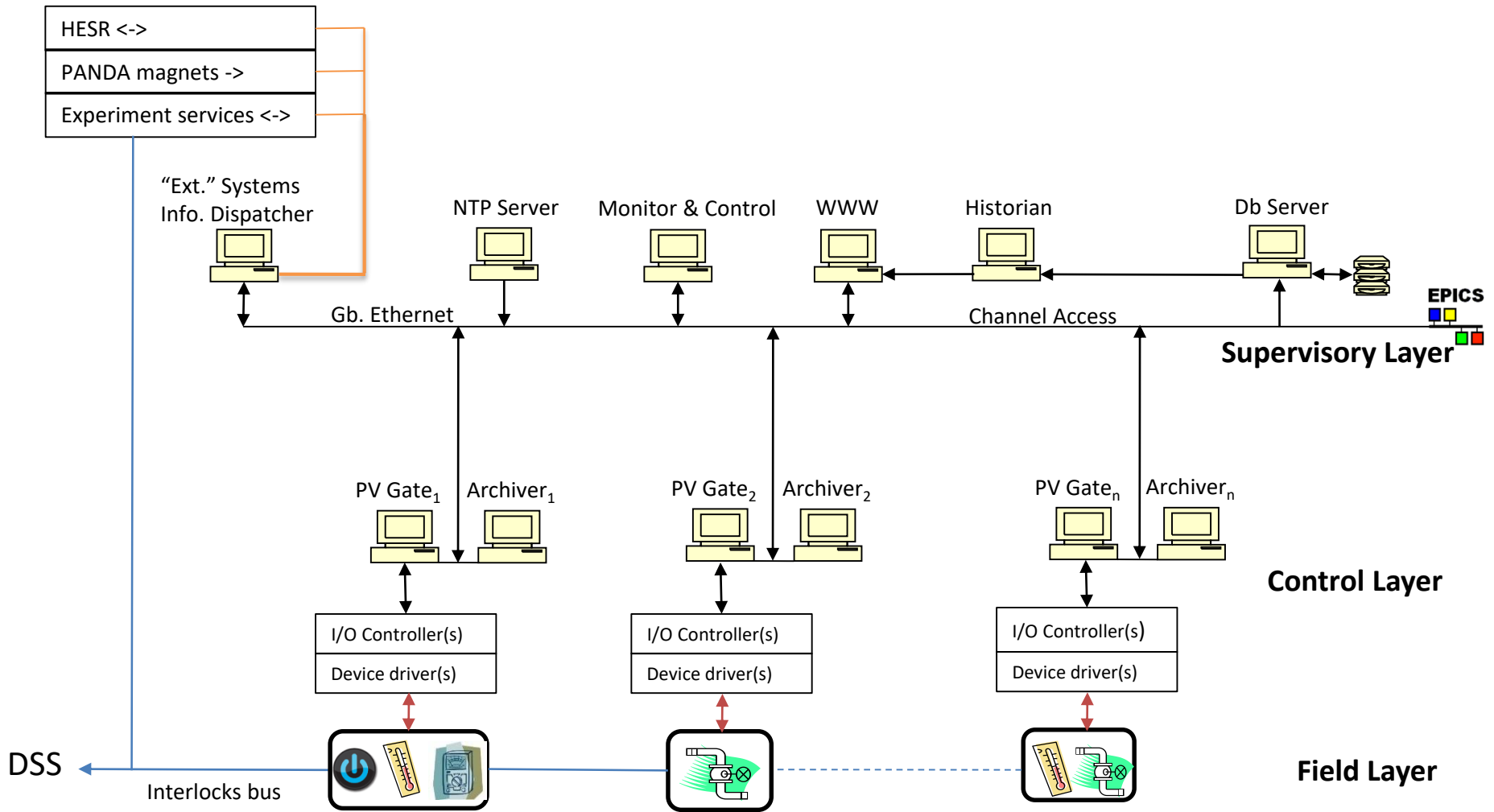
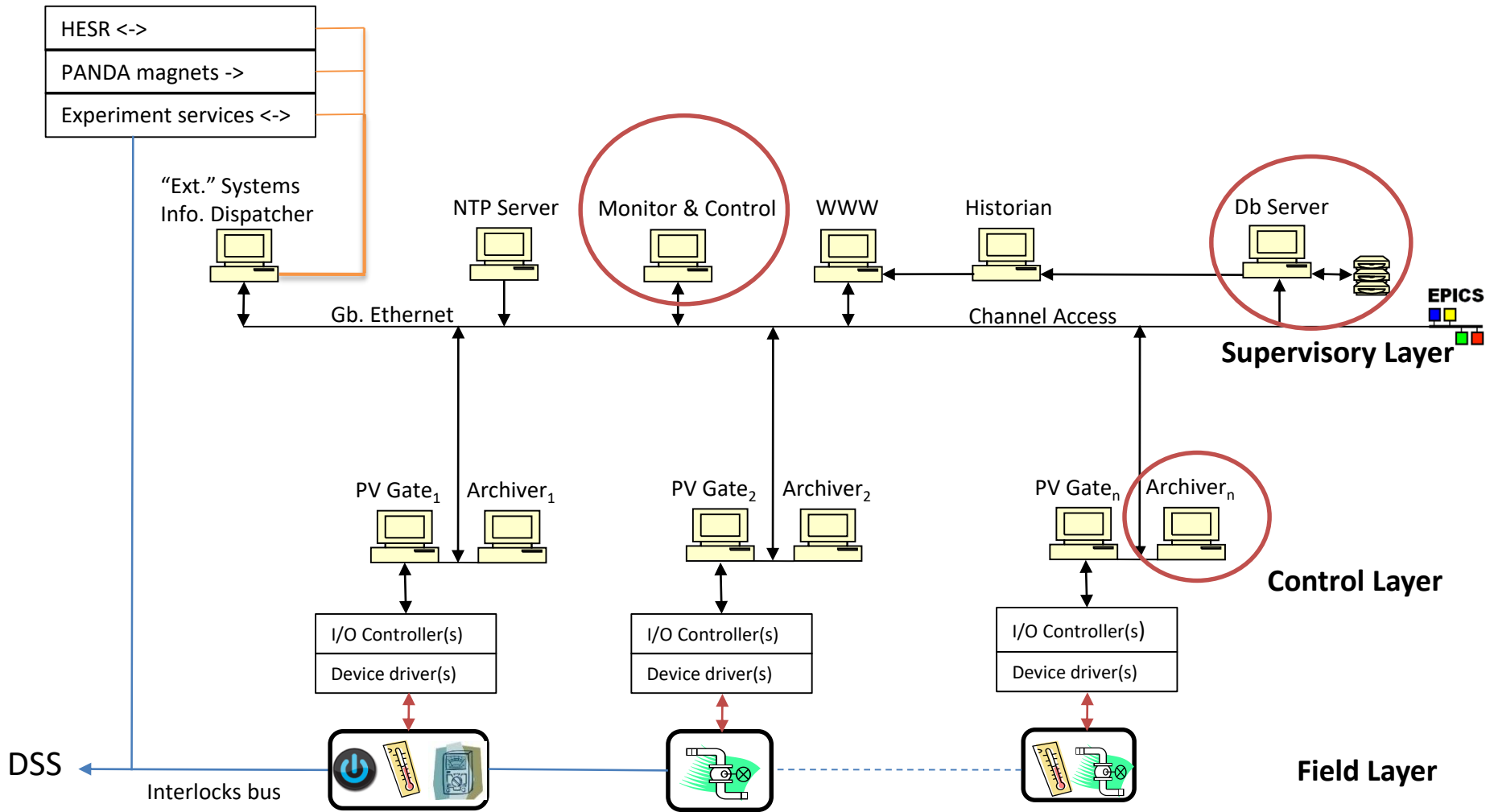


# PANDA PV archiving

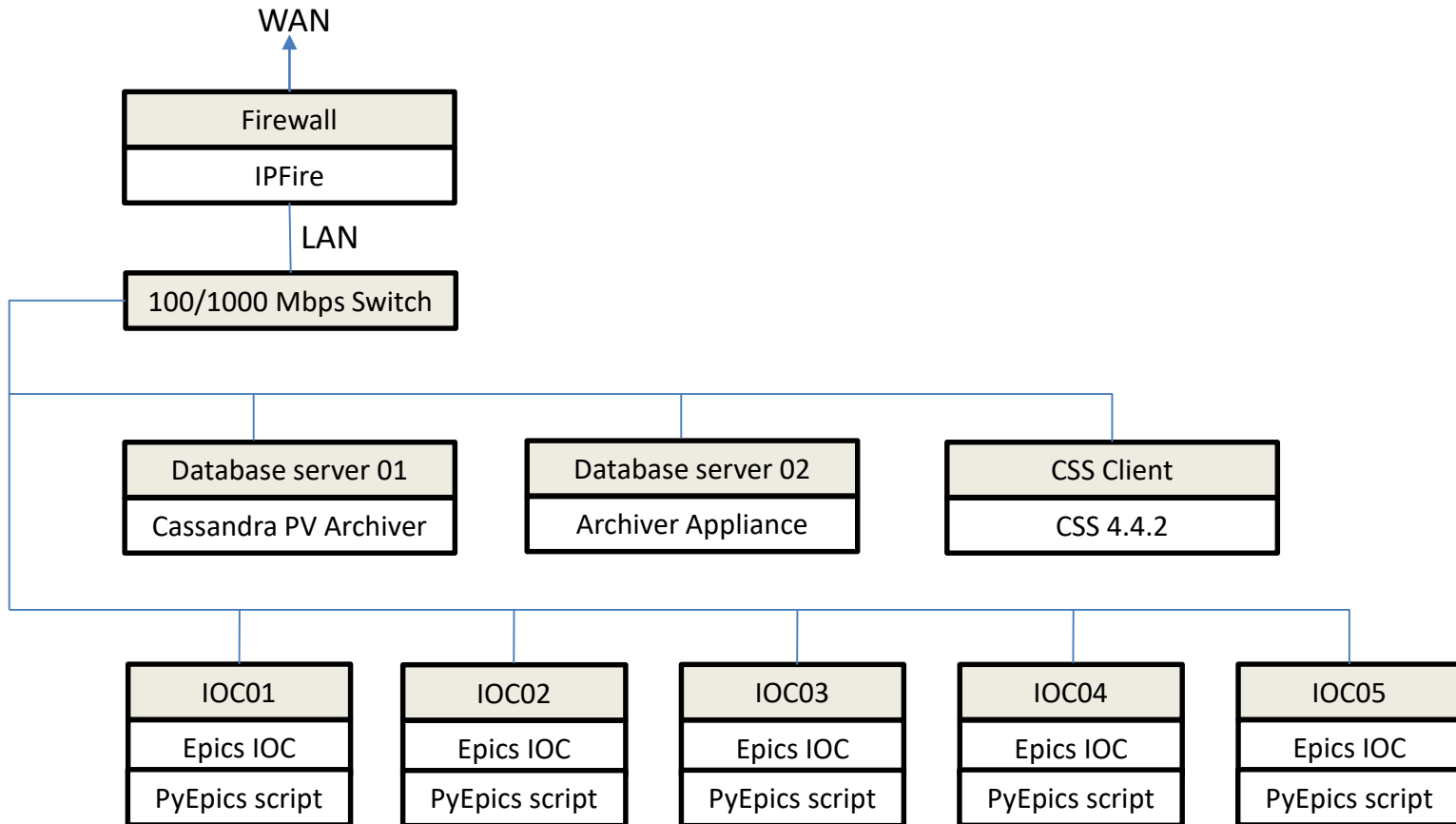
# PANDA DCS Architecture



# PANDA DCS Architecture



# IFIN-HH database testbed



**Db Servers, CSS Client, & IOC01- 03** - 2x Dual-Core AMD Opteron 2216, 8 GB RAM (Dell SC1435)  
**Firewall, IOC03-04:** Intel Xeon CPU 3.00GHz, 4 GB RAM (Dell SC1425)

# IOC

IOC....
Epics IOC
PyEpics script

- EPICS 3.14.12.7

## **Records:**

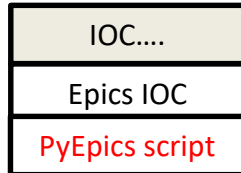
Prefixes: S- sub-system, HVCh- HV channel, LVCh –LV channel

- record(ao,"\$(S):SET\_HV\_\$(HVCh)")
- record(ao,"\$(S):SET\_CURR\_HV\_\$(HVCh)")
- record(ai,"\$(S):MON\_HV\_\$(HVCh)")
- record(ai,"\$(S):MON\_CURR\_HV\_\$(HVCh)")
- record(stringin,"\$(S):STAT\_HV\_\$(HVCh)")
- record(bo,"\$(S):SWCH\_HV\_\$(HVCh)")
- record(ao,"\$(S):SET\_LV\_\$(LVCh)")
- record(ao,"\$(S):SET\_CURR\_LV\_\$(LVCh)")
- record(ai,"\$(S):MON\_LV\_\$(LVCh)")
- record(ai,"\$(S):MON\_CURR\_LV\_\$(LVCh)")
- record(stringin,"\$(S):STAT\_LV\_\$(LVCh)")
- record(bo,"\$(S):SWCH\_LV\_\$(LVCh)")
- record(bo,"\$(S):SWCH\_ALL\_HV")
- record(bo,"\$(S):SWCH\_ALL\_LV")

- All ai records have ADEL, Alarm thresholds defined

**Substitute** file: - generated with a python script where the Prefixes are set for each sub-system (sub-system name, no. of channels)

# PV randomization



- PyEpics 3 – python module to interact with EPICS Channel Access;
- Basically the script is performing three tasks in a loop:

- 1) **read** (all) process variables generated by the Epics IOC;
- 2) **randomize** the ai records (voltage, current, ....)

Eq.

```
rand_val = default_HV -11 + 2*11*random.random()
```

```
default_HV = 1800 V ( record(ao,"$(S):SET_HV_$(HVCh)") )
```

```
field(HIHI,1810)
```

```
field(HIGH,1805)
```

```
field(LOW,1795)
```

```
field(LOLO,1790)
```

```
field(HHSV, "MAJOR")
```

```
field(HSV, "MINOR")
```

```
field(LSV, "MINOR")
```

```
field(LLSV, "MAJOR")
```

```
field(ADEL,5)
```

- 3) **writes** the new process variables;

# IOC summary

IOC01
Epics IOC
PyEpics script

STT

IOC02
Epics IOC
PyEpics script

FTRK

IOC03
Epics IOC
PyEpics script

ECAL

IOC04
Epics IOC
PyEpics script

MVD

IOC05
Epics IOC
PyEpics script

LUMI

For each sub-system

- HV ch. = 1000;
- LV ch. = 1000;

---

12.002 PV's



TOTAL  
60.010 PV's

# Apache Cassandra

*A distributed storage system for managing very large amounts of structured data spread out across many commodity servers (Avinash Lakshman, Prashant Malik – 2009, Facebook)*

- Non-relational database management system providing high availability, no single point of failure and linear scalability;
- Open source software distributed free under Apache License.

<b>Relational Database</b>	<b>Cassandra</b>
Handles moderate incoming data velocity	Handles high incoming data velocity
Data arriving from one/few locations	Data arriving from many locations
Manages primarily structured data	Manages all types of data
Supports complex/nested transactions	Supports simple transactions
Single points of failure with failover	No single points of failure; constant uptime
Supports moderate data volumes	Supports very high data volumes
Centralized deployments	Decentralized deployments
Data written in mostly one location	Data written in many locations



# Cassandra PV Archiver

Database server 01

Cassandra PV Archiver

- Application used to archive control systems data - ready to run in Epics based SCADA systems.
- Stores data in an Apache Cassandra database;
- Open source software available under the terms of the Eclipse Public License v1.0.
- Latest Release 3.2.5 (July 30th, 2017)
- <https://oss.aquenos.com/cassandra-pv-archiver/#download>

## Practice:

- Installation is very easy – tested in Ubuntu 16 and CentOS 7
- Well written and detailed documentation;
- Basic admin tasks can be performed from <http://panda-dcs-server01:4812/admin/ui/>;
- Admin via Scripts:
  - JSON-based archive access protocol;
  - requests (POST, GET) <http://panda-dcs-server01:9812/archive-access/api/1.0>;
- Importing large no of PV can be done via xml file:
  - *we developed a python script to generate the xml for each sub-system*
- Performance monitoring is very poor from the /admin/ui/ - Dashboard

# Cassandra PV Archiver

Database server 01

Cassandra PV Archiver

## Raw data sample request:

```
requests.get("http://localhost:9812/archive-  
access/api/1.0/archive/1/samples/STT:MON_LV_000?start=0&end=1528328755000  
000000&prettyPrint HTTP/1.0")
```

```
{"time":1518087797424860710,"severity":{"level":"MINOR","hasValue":true},"status":"HIGH","quality":"Original","metaData":{"type":"numeric","precision":4,"units":"V","displayLow":0.0,"displayHigh":0.0,"warnLow":11.9,"warnHigh":12.1,"alarmLow":11.8,"alarmHigh":12.2},"type":"double","value":[12.104959532825522]}
```

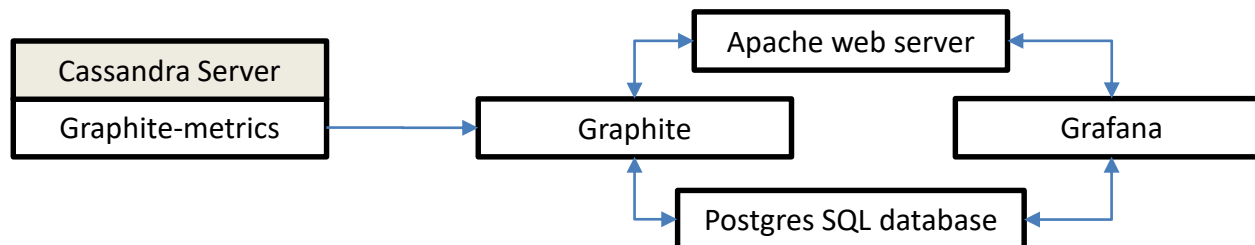
# Cassandra PV Archiver

Database server 01

Cassandra PV Archiver

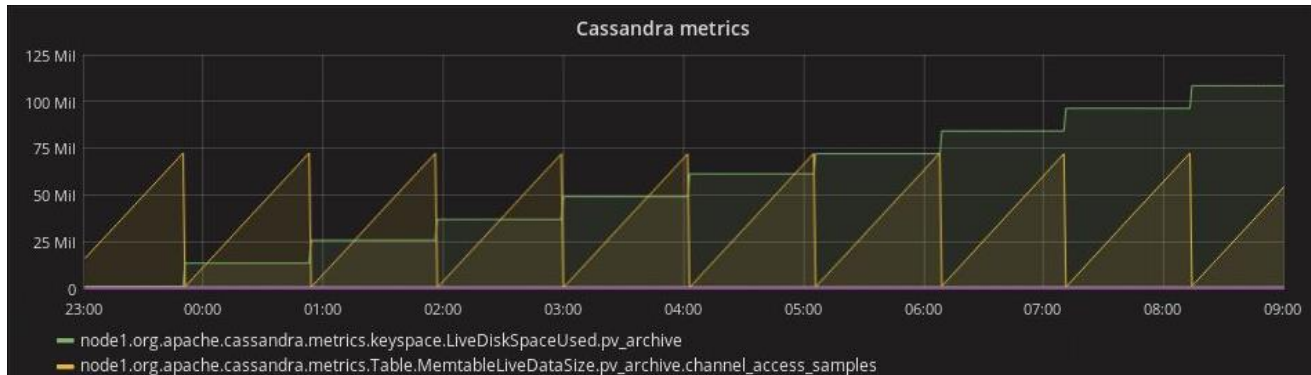
## Performance monitoring

- Apache Cassandra does not provide a dedicated GUI for monitoring (over time) the performance. However a metrics library is provided and this can be used to collect various data.
- A solution based on open source software Graphite, Grafana, Apache web server and Postgres SQL was implemented on the Database server 01  
(<https://blog.pythian.com/monitoring-apache-cassandra-metrics-graphite-grafana/>)

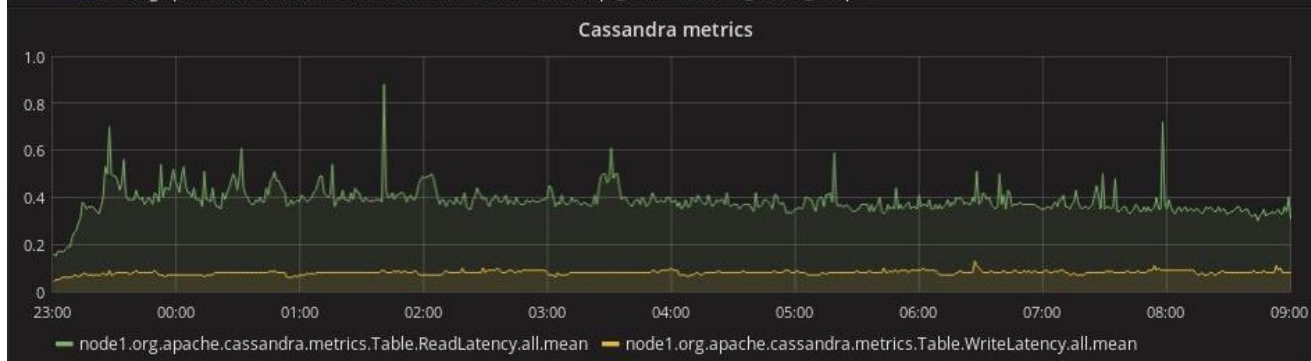


# Cassandra performance monitoring

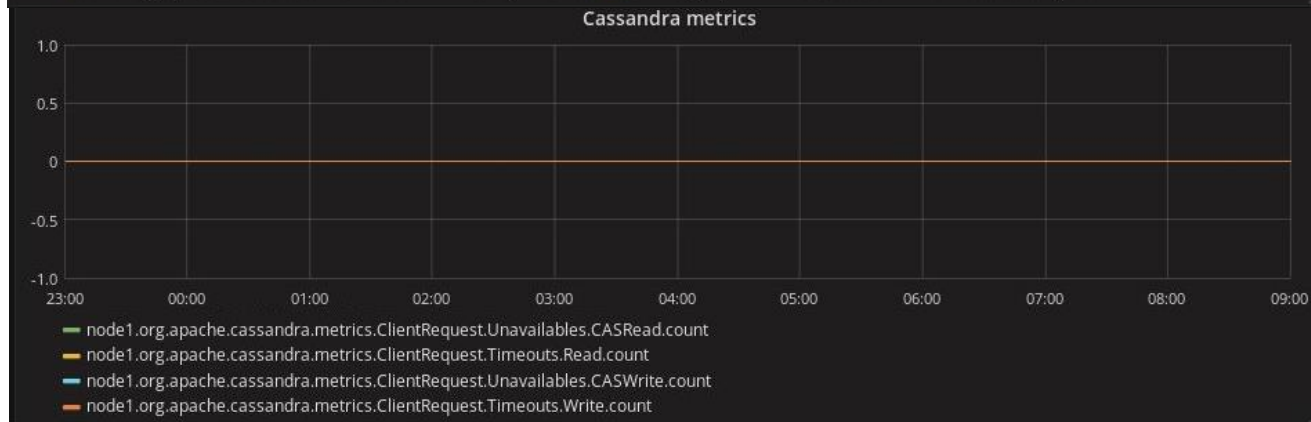
Database server 01  
Cassandra PV Archiver



LiveDiskSpaceUsed  
MemTableLiveDataSize



ReadLatency  
WriteLatency



Unavailables Read  
Timeouts Read  
Unavailables Write  
Timeouts Write

# Retrieving data from Cassandra Db in CS-Studio

## JSON Archive Proxy client plugin tool

mario@panda-css-client: ~/css/org.csstudio.ifin.product/repository/target/products/cs-studio-ifin/linux/gtk/x86\_64/cs-studio

CS-Studio

File Edit Search CS-Studio Window Help

Archive Search

URL:  Info

Name	Descrip	Key

Pattern:  Search

Add...  Replace search  Reg.Exp.

PV Name	Name
STT:MON_HV_000	Cassa
STT:MON_HV_001	Cassa
STT:MON_HV_002	Cassa
STT:MON_HV_003	Cassa
STT:MON_HV_004	Cassa
STT:MON_HV_005	Cassa
STT:MON_HV_006	Cassa
STT:MON_HV_007	Cassa
STT:MON_HV_008	Cassa
STT:MON_HV_009	Cassa
STT:MON_HV_010	Cassa
STT:MON_HV_011	Cassa
STT:MON_HV_012	Cassa
STT:MON_HV_013	Cassa

Properties

Traces | Time Axis | Value Axes | Misc.

When archived data arrives:  Do nothing  Perform 'stagger'

Show	Axis Name	Axis Name?	Trace Names?	Grid	On Right	Color	Min	Max	Auto-Scale	Scale Type	Line
<input checked="" type="checkbox"/>	Value 1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		1780.0	1820.0	<input type="checkbox"/>	Linear	1

<Not saved to file> \*<Not saved to file>

STT:MON\_HV\_000 [V]

1780 1790 1800 1810 1820

13:40 13:50 14:00 14:10 14:20 14:30

2018-02-08

STT:MON\_HV\_000 [V]

mario

# Cassandra PV Archiver

Database server 01

Cassandra PV Archiver

## 60.010 PV's :

- Import, via .xml, takes about 50 minutes with no errors;
- Archiving works but retrieval is stuck and the interface is not-responsive ;
  
- *With a single sub-system loaded (12.002 channels) I encountered no problems in the admin or retrieval of data from the Cassandra db.*

Single node installation shows its limits ... Fortunately the PV Archiver is scalable. A cluster of nodes can be implemented (hopefully with ease). I am tempted to try it soon ....

# Archiver Appliance

Database server 02

Archiver Appliance

- Java based application archiver for EPICS Control systems;
- Developed and used at SLAC, BNL and MSU (aiming) to archive millions of PV's.
- [https://slacmshankar.github.io/epicsarchiver\\_docs/details.html](https://slacmshankar.github.io/epicsarchiver_docs/details.html)

## Practice:

- Installation is not simple. I used the site-specific install from <https://github.com/jeonghanlee/epicsarchiverap-sites> for a single production node
- The documentation can be better organized ... but is doing the job;
- Basic admin tasks can be performed from <http://panda-dcs-server02:17665/mgmt/ui/index.html>
- Admin via Scripts:
  - JSON-based archive access protocol;
  - requests (POST, GET) <http://panda-dcs-server01:17665/mgmt/bpl/>
- Importing large no of PV can be done via xml file ( Channel Archiver configuration file)
  - *we developed a python script to generate the xml for each sub-system*
- Performance monitoring provides a lot of useful information

# Archiver Appliance Metrics

Database server 02

Archiver Appliance

appliance archiver - Metrics - Mozilla Firefox

appliance archiver - ... x +

192.168.1.11:17665/mgmt/ui/metrics.html | Search

Most Visited Getting Started GitHub - jeonghanle... EPICS Archiver Appli...

Home Reports Metrics Storage Appliances Integration Help

25 Page 1 of 1

Instance Name	Status	PV Count	Connected	Event Rate	Data Rate (GB/day)	Engine write thread(s)	Max ETL(%)
appliance0	Working	12002	12002	106.83	0.2	0.33	1

Here are the some detailed metrics of the appliance **appliance0**

Attribute	Detail
Appliance Identity	appliance0
Total PV count	12002
Disconnected PV count	0
Connected PV count	12002
Paused PV count	0
Total channels	80004
Approx pending jobs in engine queue	1
Event Rate (in events/sec)	106.83
Data Rate (in bytes/sec)	2,432.99
Data Rate in (GB/day)	0.2
Data Rate in (GB/year)	71.46
Time consumed for writing samplebuffers to STS (in secs)	0.33
Benchmark - writing at (events/sec)	3,224.66



# Archiver Appliance

Database server 02

Archiver Appliance

## Raw data sample request:

- Raw data [decoding utils](#) are included in the src:

`./pb2json.sh`

`/mnt/storage/arch/sts/ArchiverStore/STT/MON_LV_000\ :2018_02_08_11.pb`

```
{"timeStamp":"2018-02-08T11:59:57.412Z","severity":1,"value":"12.136834570488617","status":4}
```

```
├── MON_LV_994:2018_02_08_12.pb
├── MON_LV_995:2018_02_08_12.pb
├── MON_LV_996:2018_02_08_12.pb
├── MON_LV_997:2018_02_08_12.pb
├── MON_LV_998:2018_02_08_12.pb
└── MON_LV_999:2018_02_08_12.pb

8 directories, 14287 files
[root@panda-dcs-server02 storage]# tree -d
.
├── arch
│   ├── lts
│   │   └── ArchiverStore
│   ├── mts
│   │   ├── ArchiverStore
│   │   └── STT
│   └── sts
│       ├── ArchiverStore
│       └── STT
└── lost+found

10 directories
[root@panda-dcs-server02 storage]#
```

- Raw data file created for each PV;
- A LOT of files ... a single database is by far more suitable (see Cassandra) from management point of view;

# Archiver Appliance

Database server 02

Archiver Appliance

## 60.010 PV's :

- Import, via .xml, took about 3 days with many start stop services and reboots. I stopped trying ....;
- *With a single sub-system loaded (12.002 channels) I encountered no problems in the admin or retrieval of data from the raw storage.*

Single node installation shows its limits ... Fortunately the Archiver is scalable. A cluster of nodes can be implemented (hopefully with ease). I am tempted to try it soon ...

# Summary and Outlook

- Controls TDR preparation is moving ahead. Some more work is needed for the evaluation of overall controls data throughput, storage architecture and retrieval.