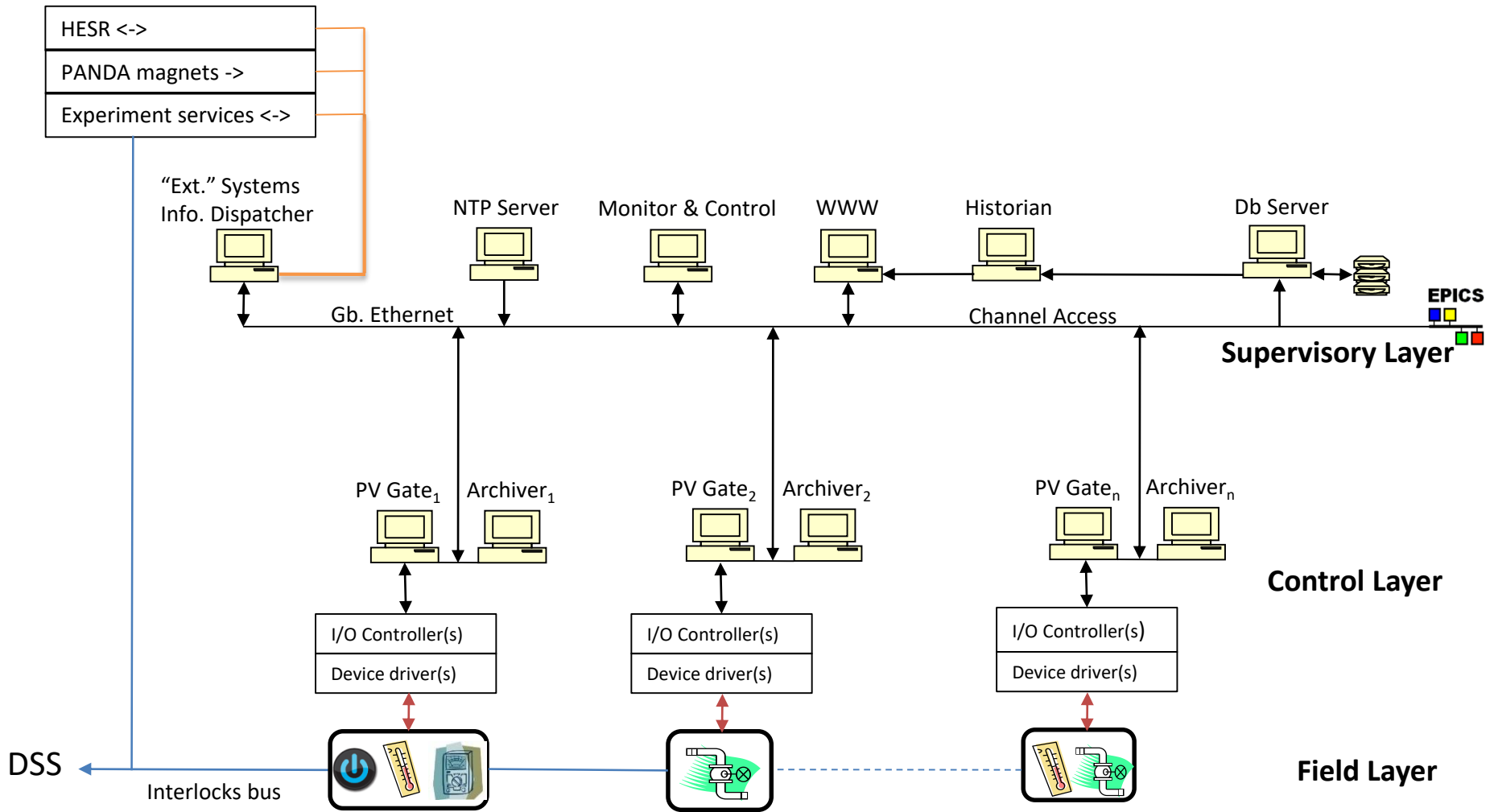


# PANDA PV archiving

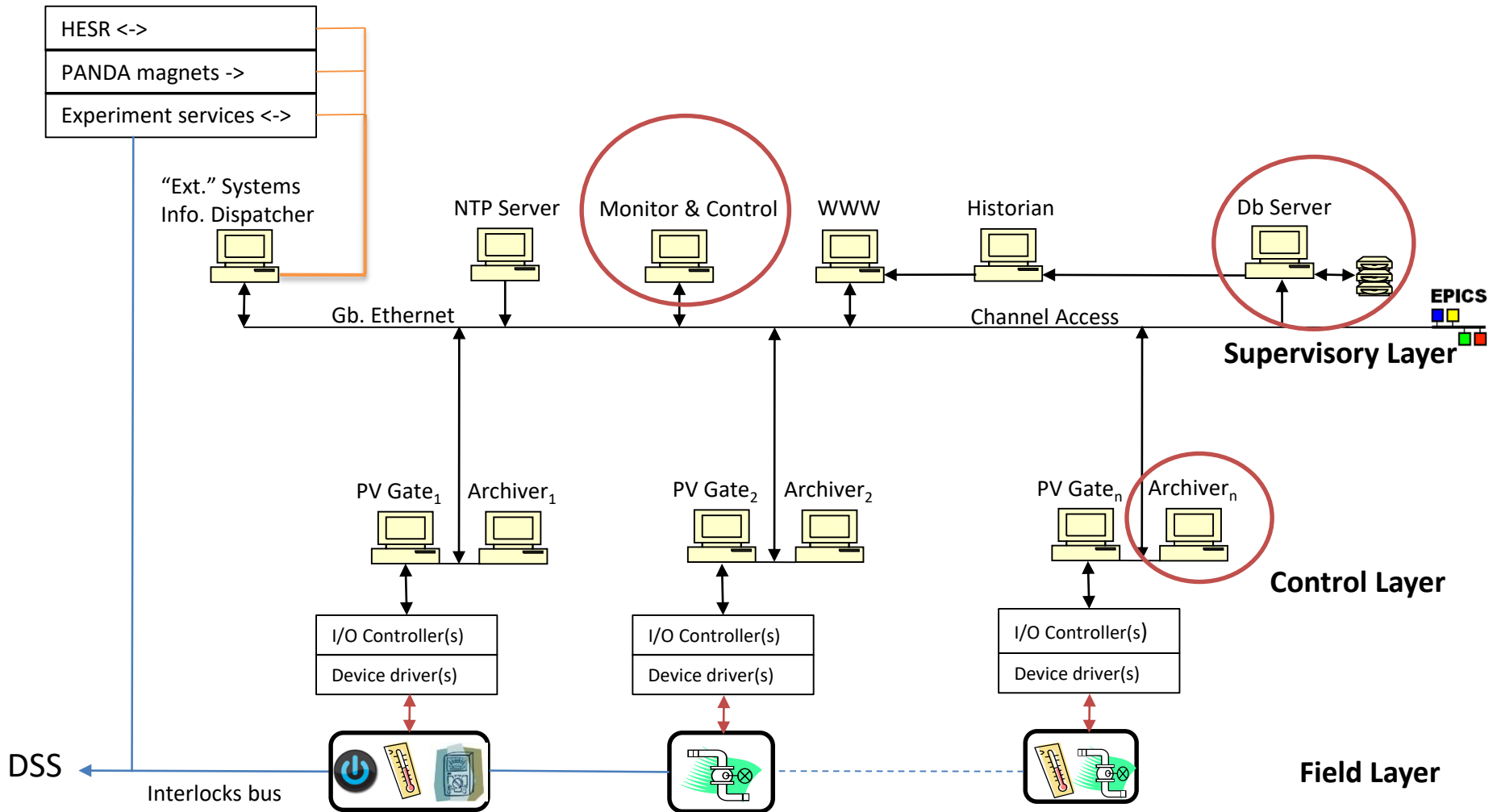
# Outline

- Introduction;
- IFIN-HH database testbed:
  - before PANDA DCS core meeting (08.02.2018);
  - upgraded (from slide no. 20).
- Outlook

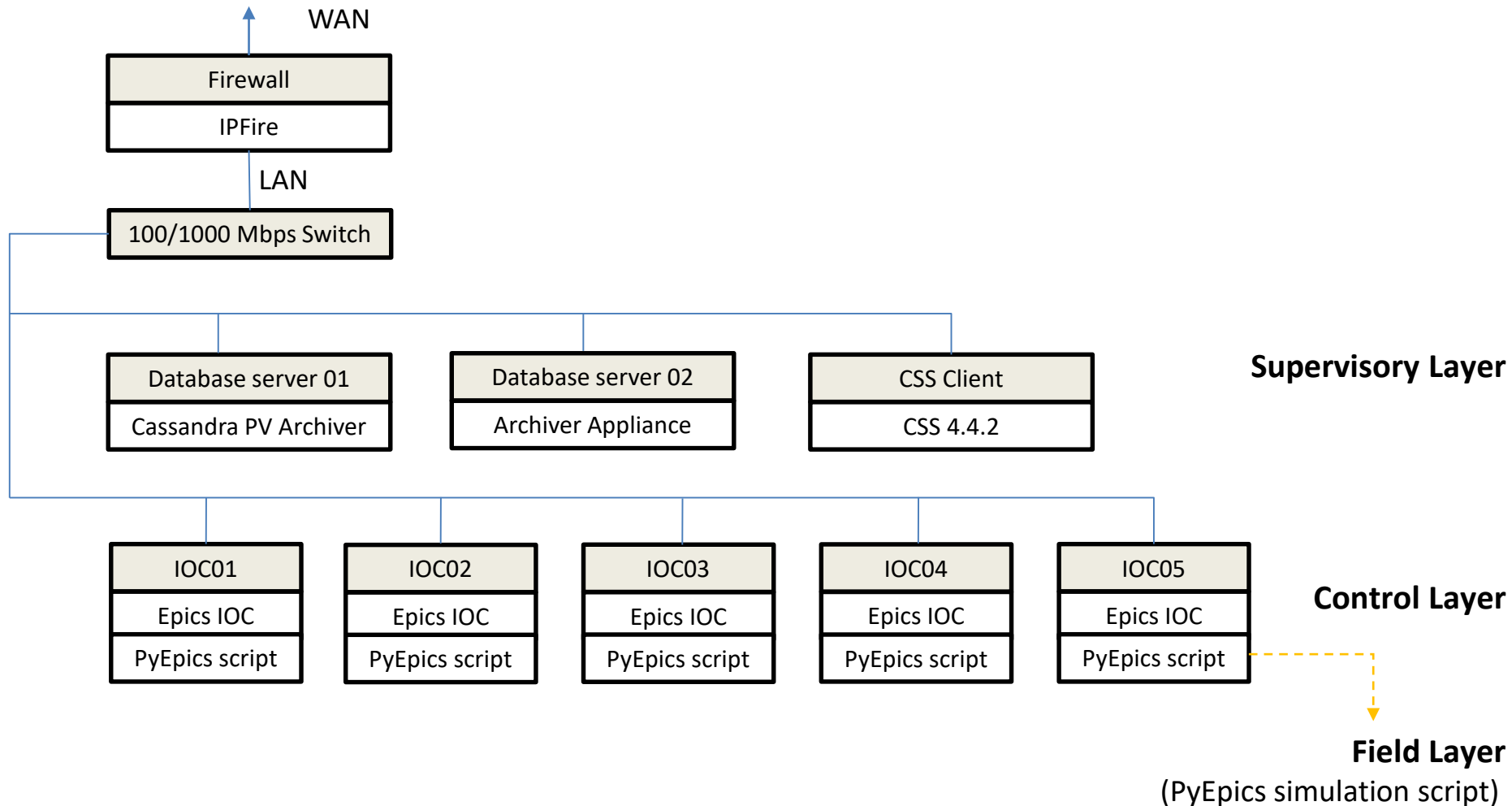
# PANDA DCS Architecture



# 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:

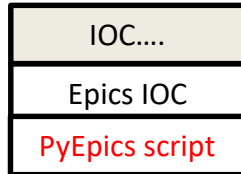
Prefixes: S- sub-system, HVCh- HV channel, LVCh –LV channel

- 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:

- **read** (all) process variables generated by the Epics IOC;
- **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)
```

- **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;
- HV General Switch =1;
- LV General Switch =1;

-----  
Total PV's 12.002



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

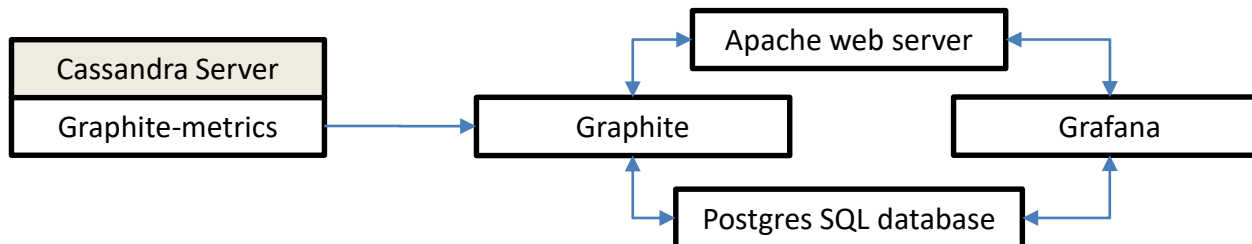
Database server 01

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 node and cluster statistic data (*nodetool*, *Jconsole* – not for production).
- 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

The screenshot shows the CS-Studio interface with the following components:

- Archive Search Panel:** URL: `Cassandra - json: http://`. Search Pattern: `STT:MON_HV_*`. Options:  Add...,  Replace search,  Reg.Exp.
- Properties Panel:** Traces | Time Axis | Value Axes | Misc. |  
When archived data arrives:  Do nothing  Perform 'stagger'
- Table:**

Show	Axis Name	Axis Name?	Trace Names?	Grid	On Right	Color	Min	Max	Auto-Scale	Scale Type	Level
<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

**Main Visualization:** A line graph showing voltage data for `STT:MON_HV_000 [V]` over time. The Y-axis ranges from 1780 to 1820 V. The X-axis shows time from 13:40 to 14:30 on 2018-02-08. The data is plotted as a red line with a sawtooth pattern, indicating periodic voltage fluctuations.

# 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

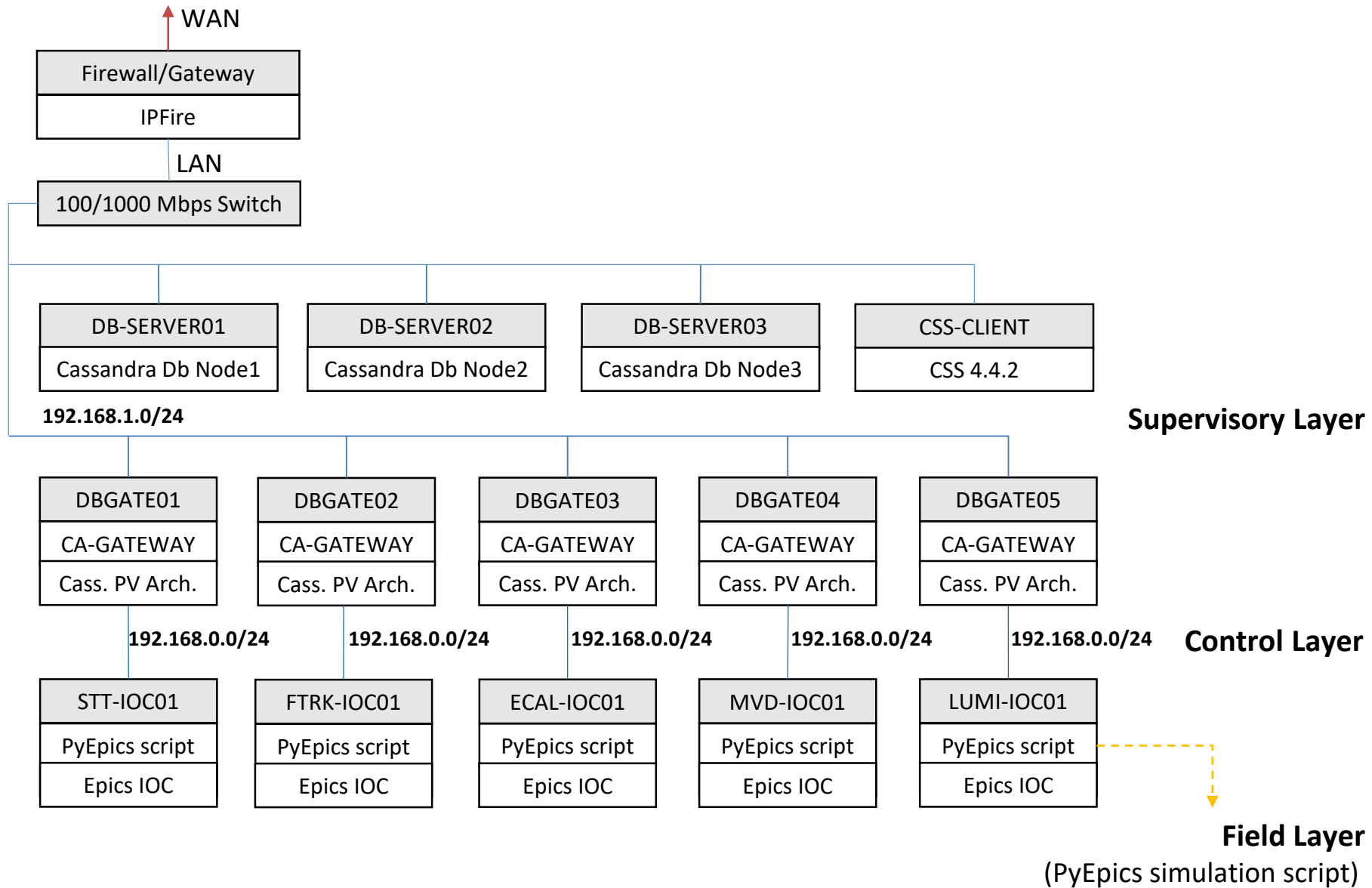
Archiver Appliance

## 60.010 PV's :

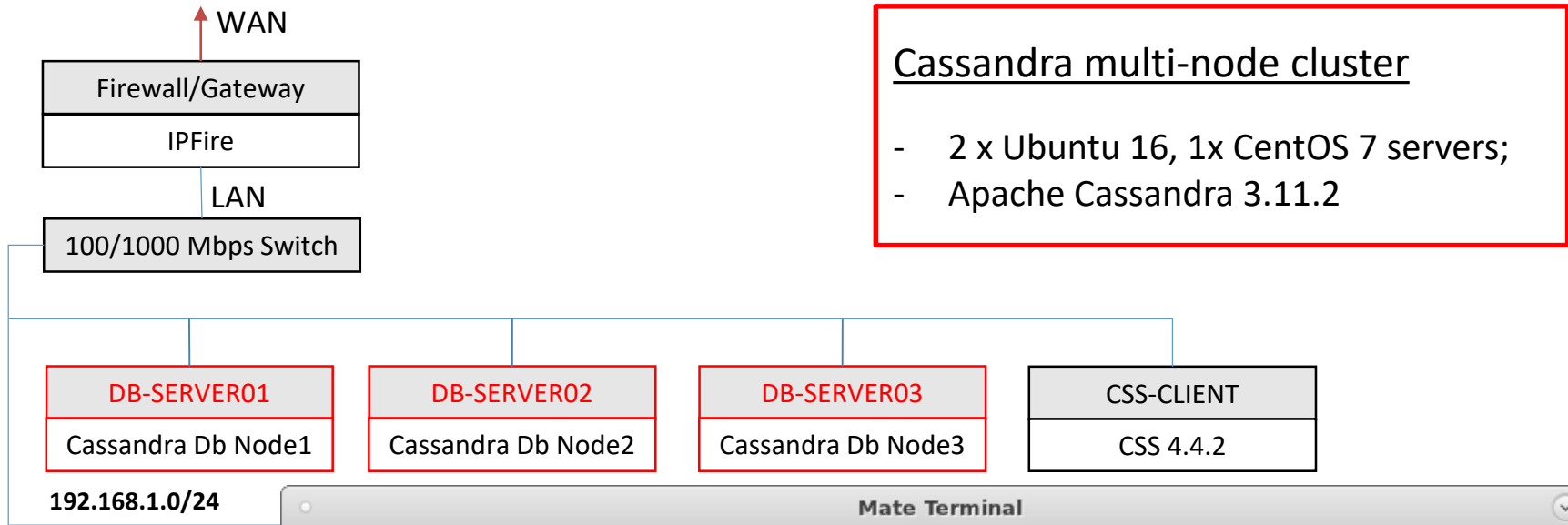
- Import, via .xml, took about 3 days with many start stop services and reboots. I stopped trying ....;
- *With a single sub-system loaded (12.002 channels) I encountered no problems in the admin or retrieval of data from the raw storage.*

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 ...

# IFIN-HH (upgraded) database testbed

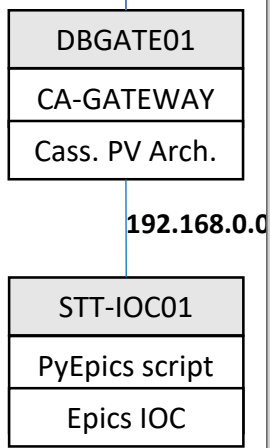


# IFIN-HH (upgraded) database testbed



Cassandra multi-node cluster

- 2 x Ubuntu 16, 1x CentOS 7 servers;
- Apache Cassandra 3.11.2



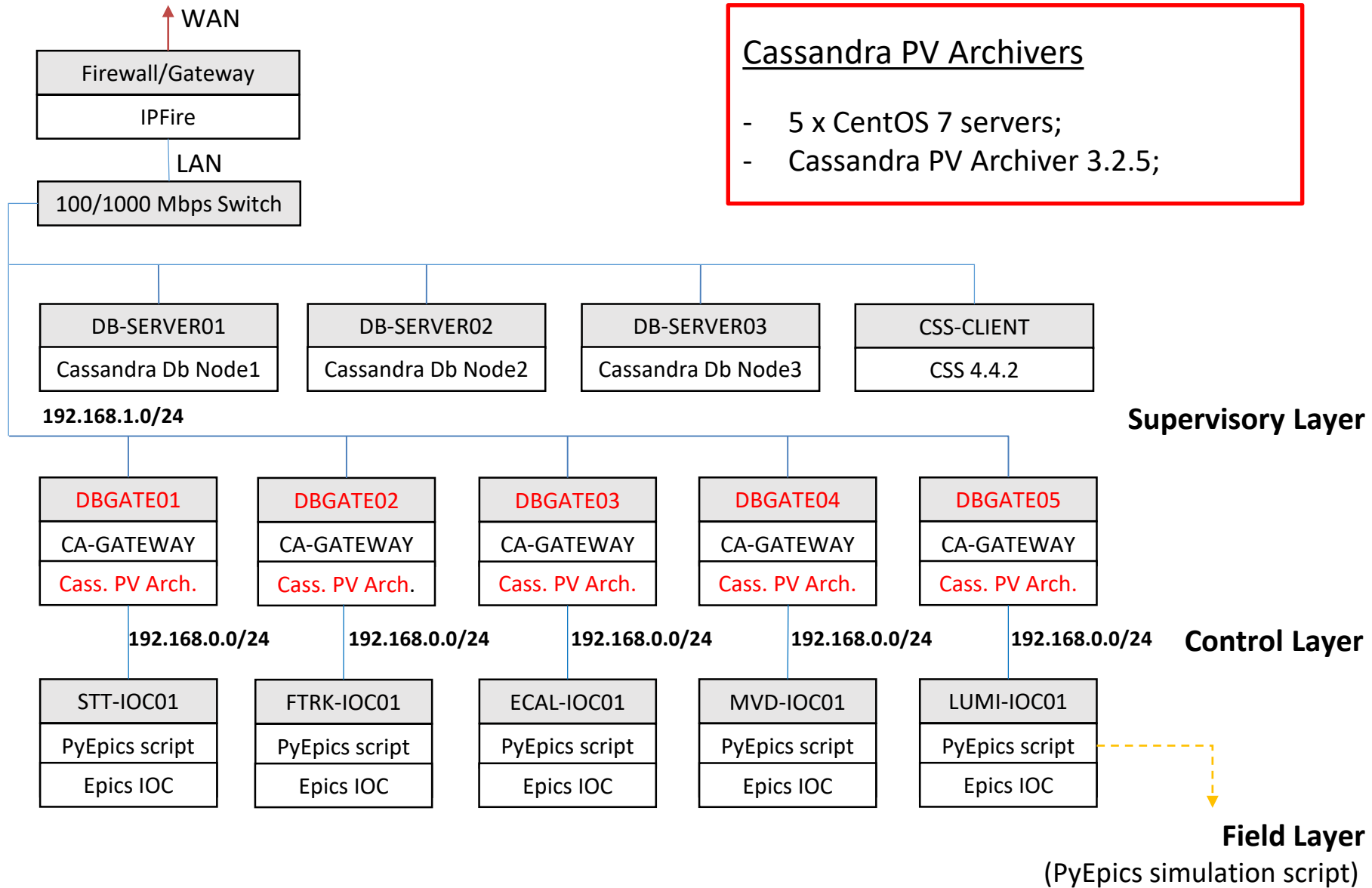
```
Mate Terminal
File Edit View Search Terminal Help
[mario@panda-dcs-dbgate01 ~]$ ssh mario@panda-dcs-server01
mario@panda-dcs-server01's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-116-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

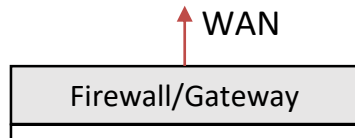
1 package can be updated.
0 updates are security updates.

Last login: Fri Feb 23 11:35:12 2018 from 194.102.58.183
mario@panda-dcs-server01:~$ nodetool status
Datacenter: dcl
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address          Load        Tokens      Owns (effective)  Host ID                               Rack
UN  192.168.1.10      97.8 MiB    256         100.0%            00546fd5-2f65-4f87-ad5a-da9c7fcf8c82  rack1
UN  192.168.1.11      99.47 MiB   256         100.0%            e3dc7d96-1990-4a40-9e1e-14bb1d9868e7  rack1
UN  192.168.1.12      99.28 MiB   256         100.0%            ec387a29-10a3-4126-8e99-ec09c2c85153  rack1
mario@panda-dcs-server01:~$
```

# IFIN-HH (upgraded) database testbed



# IFIN-HH (upgraded) database testbed



Cassandra PV Archivers

Cassandra PV Archiver - Channels - Mozilla Firefox

Cassandra PV Archiver - ... x +

← | panda-dcs-dbgate01:4812/admin/ui/channels/ | Search

Cassandra PV Archiver [Dashboard](#) [Channels](#) [About](#) [Sign in](#)

Channels

## Channels

Search:

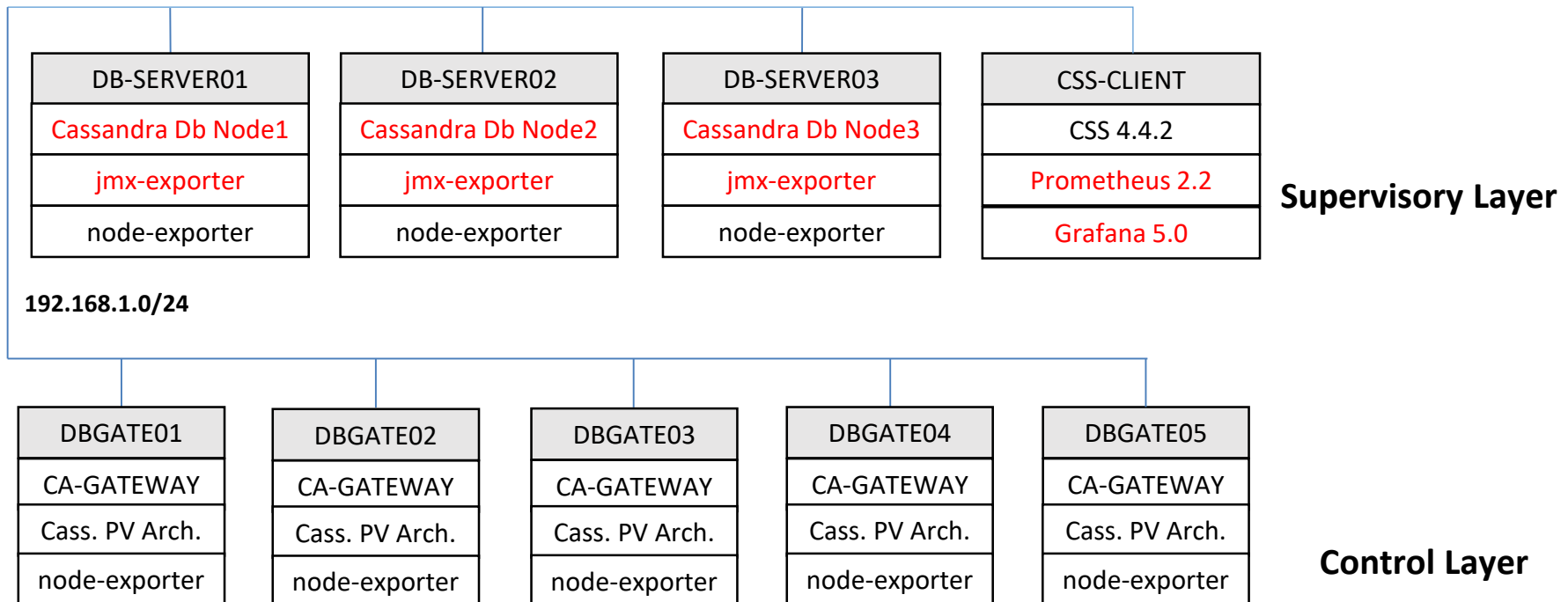
Server ID	Server Name	Server Online?	# Channels
<a href="#">All Channels</a>			60010
097f11ed-6485-49e6-915a-76630cf40e93	<a href="#">panda-dcs-dbgate01</a>	✓	12002
0b8a001c-c514-411f-b2c0-7aea0cacc0a8	<a href="#">panda-dcs-dbgate02</a>	✓	12002
25567aa9-a588-42eb-ab0e-3ae74046932d	<a href="#">panda-dcs-dbgate04</a>	✓	12002
e7f60b07-fddf-4260-b831-23a4fad4fc28	<a href="#">panda-dcs-dbgate05</a>	✓	12002
fec63b7d-531f-40e5-8fbb-bbcf1657f496	<a href="#">panda-dcs-dbgate03</a>	✓	12002

Showing 1 to 5 of 5 entries

# IFIN-HH (upgraded) database testbed

## Cassandra Performance monitoring

- We switched from Graphite to Prometheus;
- jmx-exporter: java agent which exports local JVM metrics as Prometheus metrics;

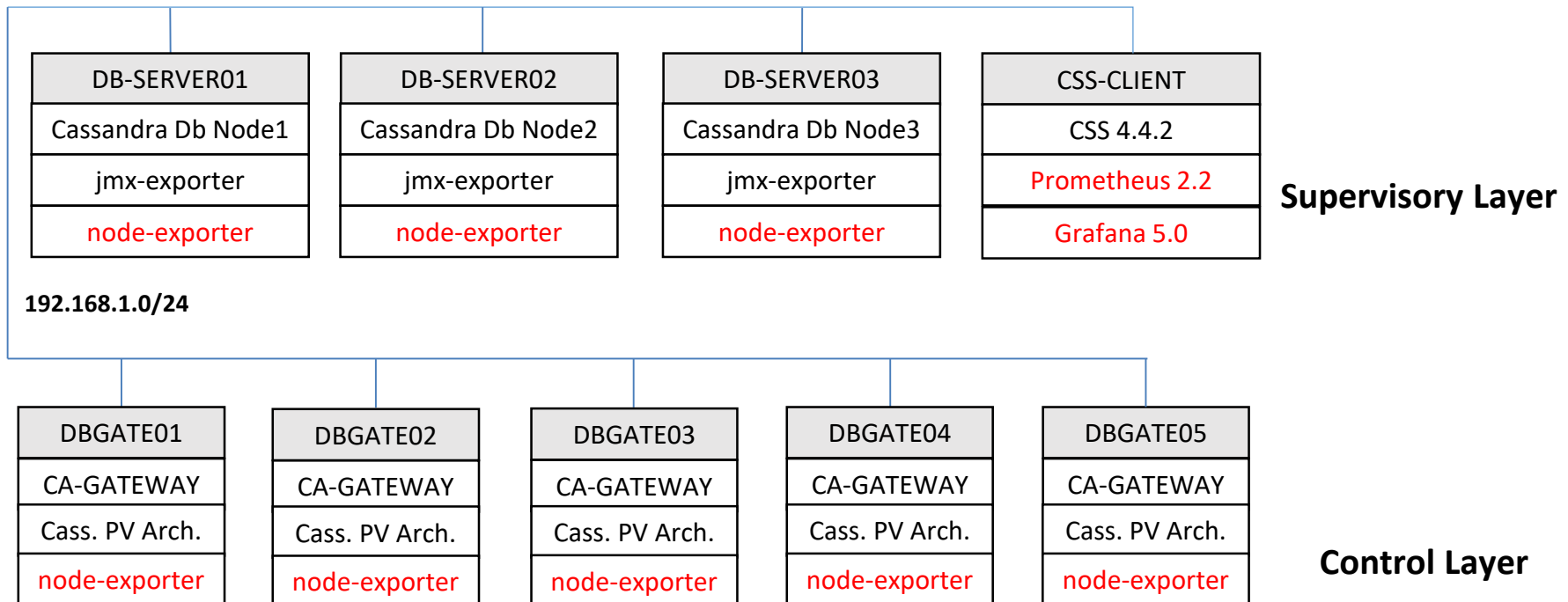




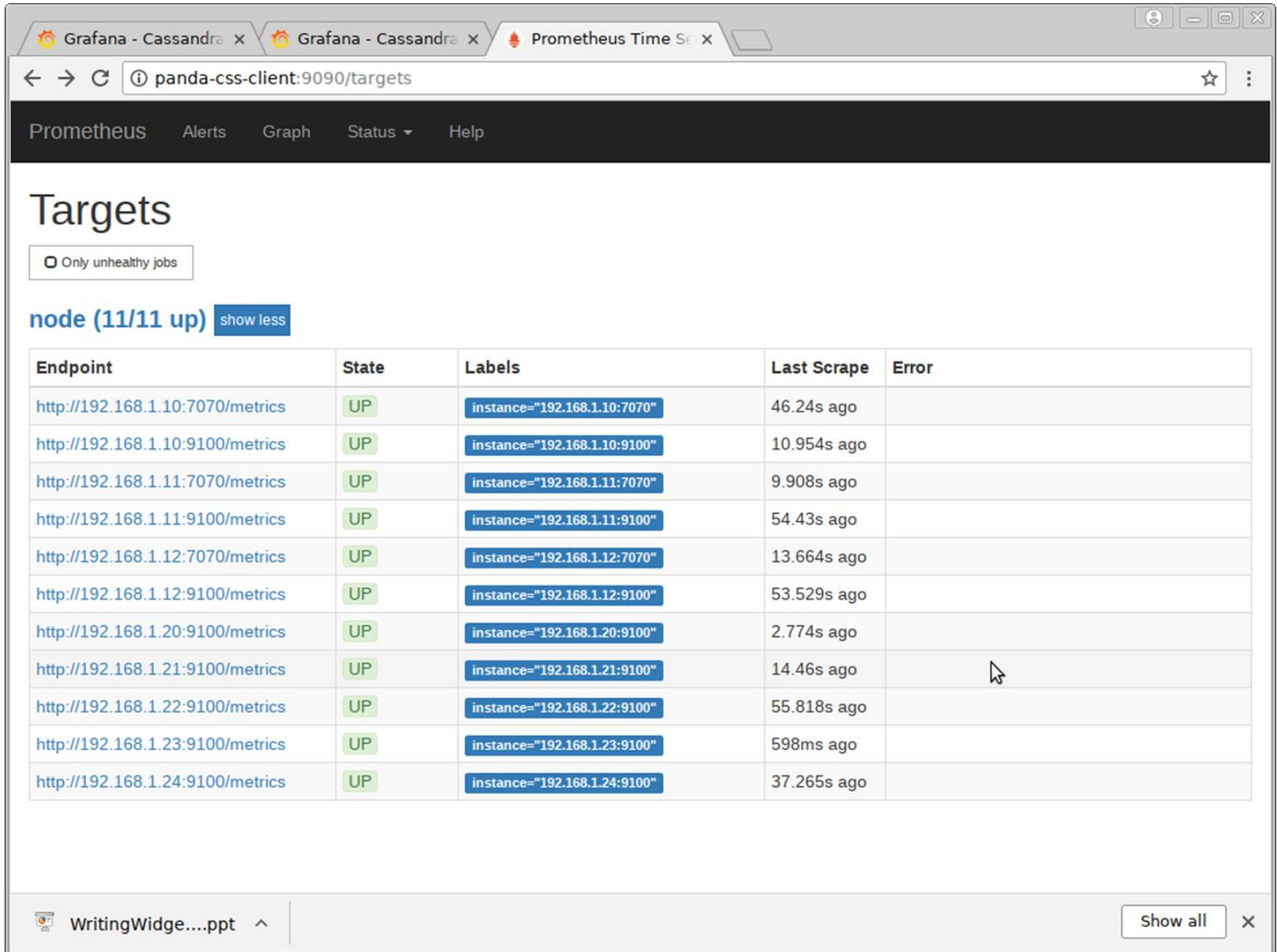
# IFIN-HH (upgraded) database testbed

## General Performance monitoring

- Node-exporter: exports machine hardware and OS metrics to Prometheus;
- Grafana dashboard to monitor CPU, RAM, HDD, ...



# IFIN-HH database testbed – Performance Monitoring

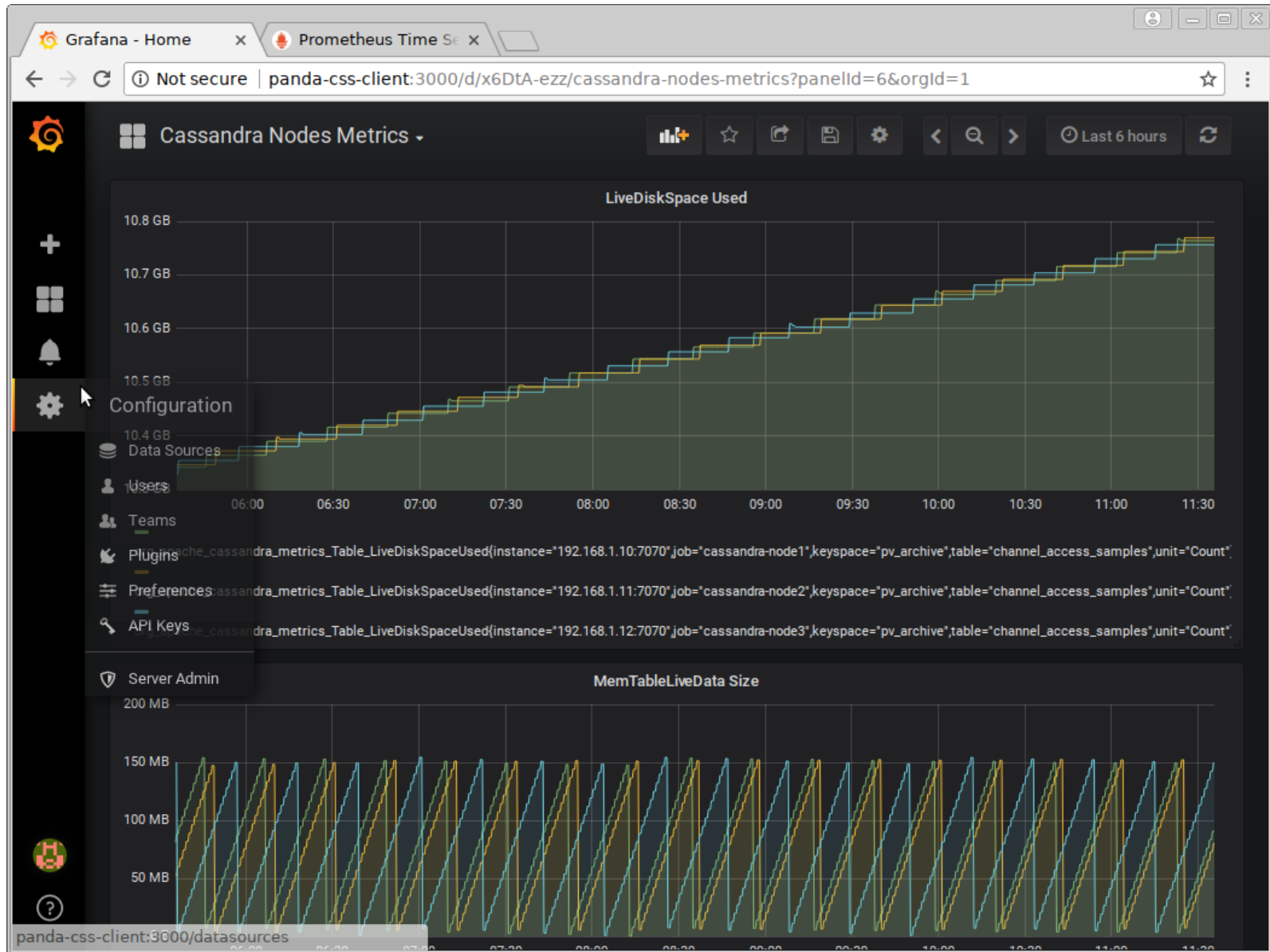


The screenshot shows the Prometheus Targets page in Grafana. The browser tabs include 'Grafana - Cassandra', 'Grafana - Cassandra', and 'Prometheus Time Series'. The address bar shows 'panda-css-client:9090/targets'. The navigation bar includes 'Prometheus', 'Alerts', 'Graph', 'Status', and 'Help'. The main heading is 'Targets', with a filter 'Only unhealthy jobs' (unchecked). Below the heading, it says 'node (11/11 up)' with a 'show less' button. A table lists 11 targets, all with a state of 'UP'. The table columns are Endpoint, State, Labels, Last Scrape, and Error.

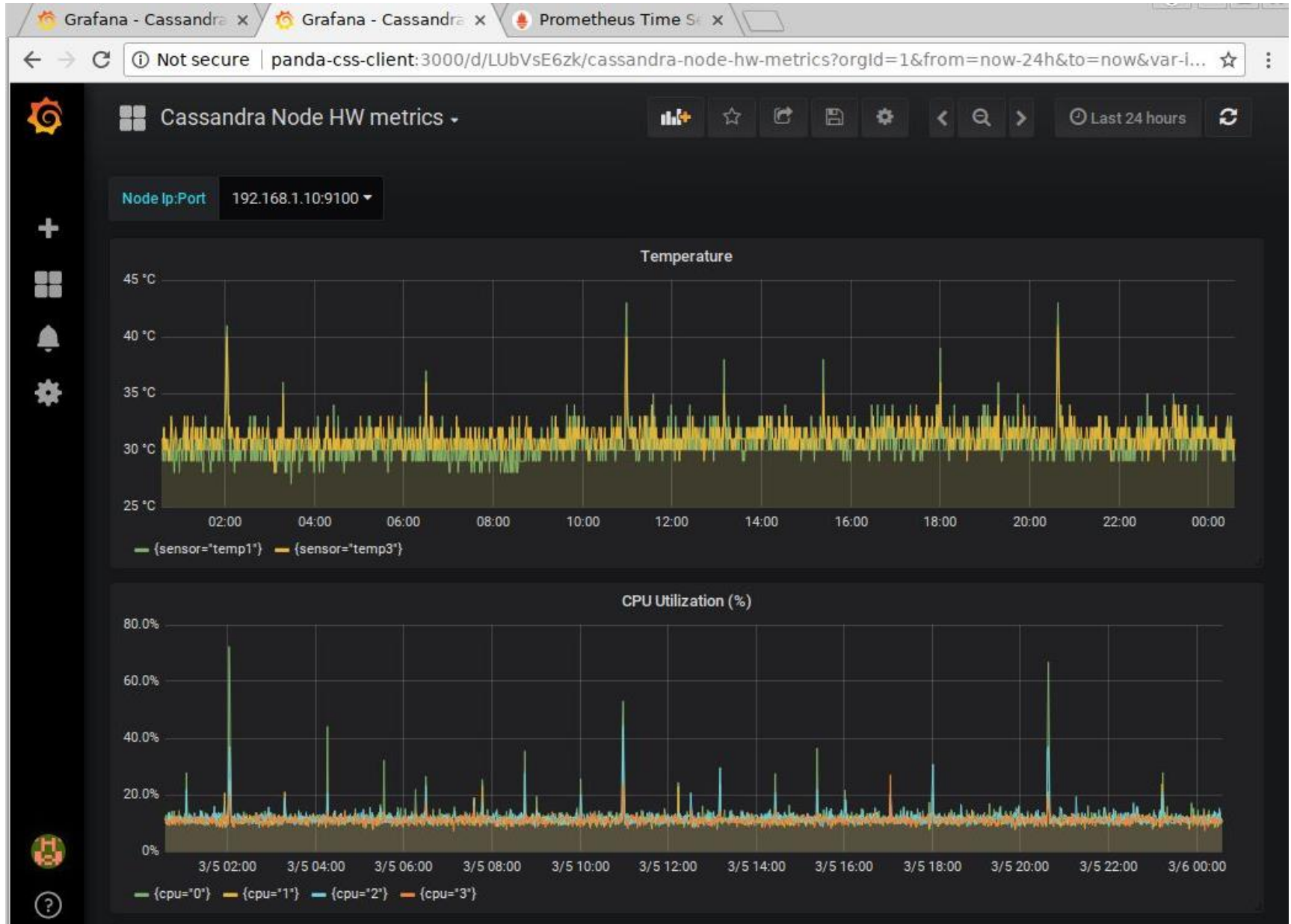
Endpoint	State	Labels	Last Scrape	Error
<a href="http://192.168.1.10:7070/metrics">http://192.168.1.10:7070/metrics</a>	UP	instance="192.168.1.10:7070"	46.24s ago	
<a href="http://192.168.1.10:9100/metrics">http://192.168.1.10:9100/metrics</a>	UP	instance="192.168.1.10:9100"	10.954s ago	
<a href="http://192.168.1.11:7070/metrics">http://192.168.1.11:7070/metrics</a>	UP	instance="192.168.1.11:7070"	9.908s ago	
<a href="http://192.168.1.11:9100/metrics">http://192.168.1.11:9100/metrics</a>	UP	instance="192.168.1.11:9100"	54.43s ago	
<a href="http://192.168.1.12:7070/metrics">http://192.168.1.12:7070/metrics</a>	UP	instance="192.168.1.12:7070"	13.664s ago	
<a href="http://192.168.1.12:9100/metrics">http://192.168.1.12:9100/metrics</a>	UP	instance="192.168.1.12:9100"	53.529s ago	
<a href="http://192.168.1.20:9100/metrics">http://192.168.1.20:9100/metrics</a>	UP	instance="192.168.1.20:9100"	2.774s ago	
<a href="http://192.168.1.21:9100/metrics">http://192.168.1.21:9100/metrics</a>	UP	instance="192.168.1.21:9100"	14.46s ago	
<a href="http://192.168.1.22:9100/metrics">http://192.168.1.22:9100/metrics</a>	UP	instance="192.168.1.22:9100"	55.818s ago	
<a href="http://192.168.1.23:9100/metrics">http://192.168.1.23:9100/metrics</a>	UP	instance="192.168.1.23:9100"	598ms ago	
<a href="http://192.168.1.24:9100/metrics">http://192.168.1.24:9100/metrics</a>	UP	instance="192.168.1.24:9100"	37.265s ago	

At the bottom of the page, there is a 'WritingWidge...ppt' tab and a 'Show all' button with a close icon.

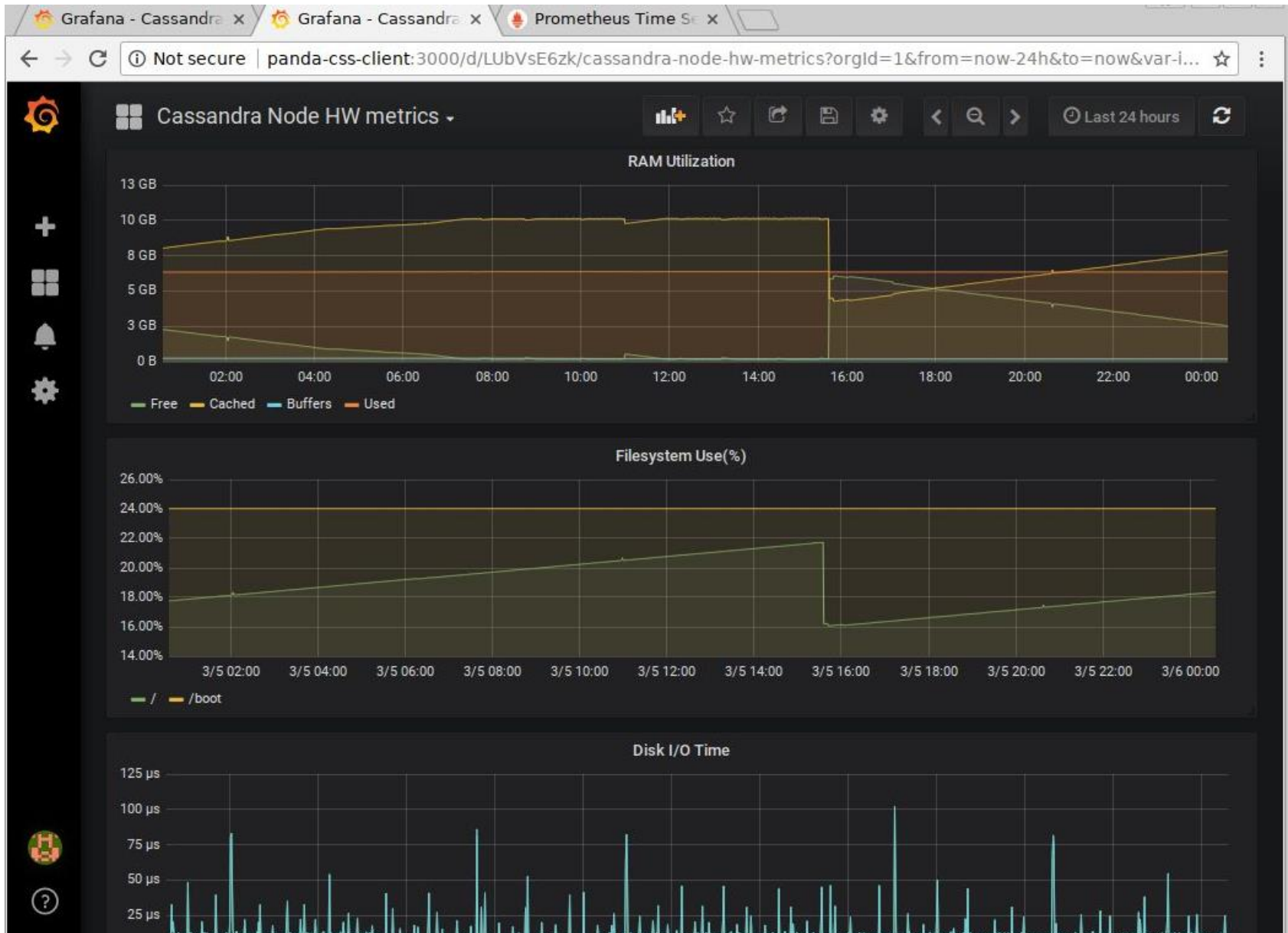
# IFIN-HH database testbed – Cassandra Monitoring



# IFIN-HH database testbed – Hardware Monitoring

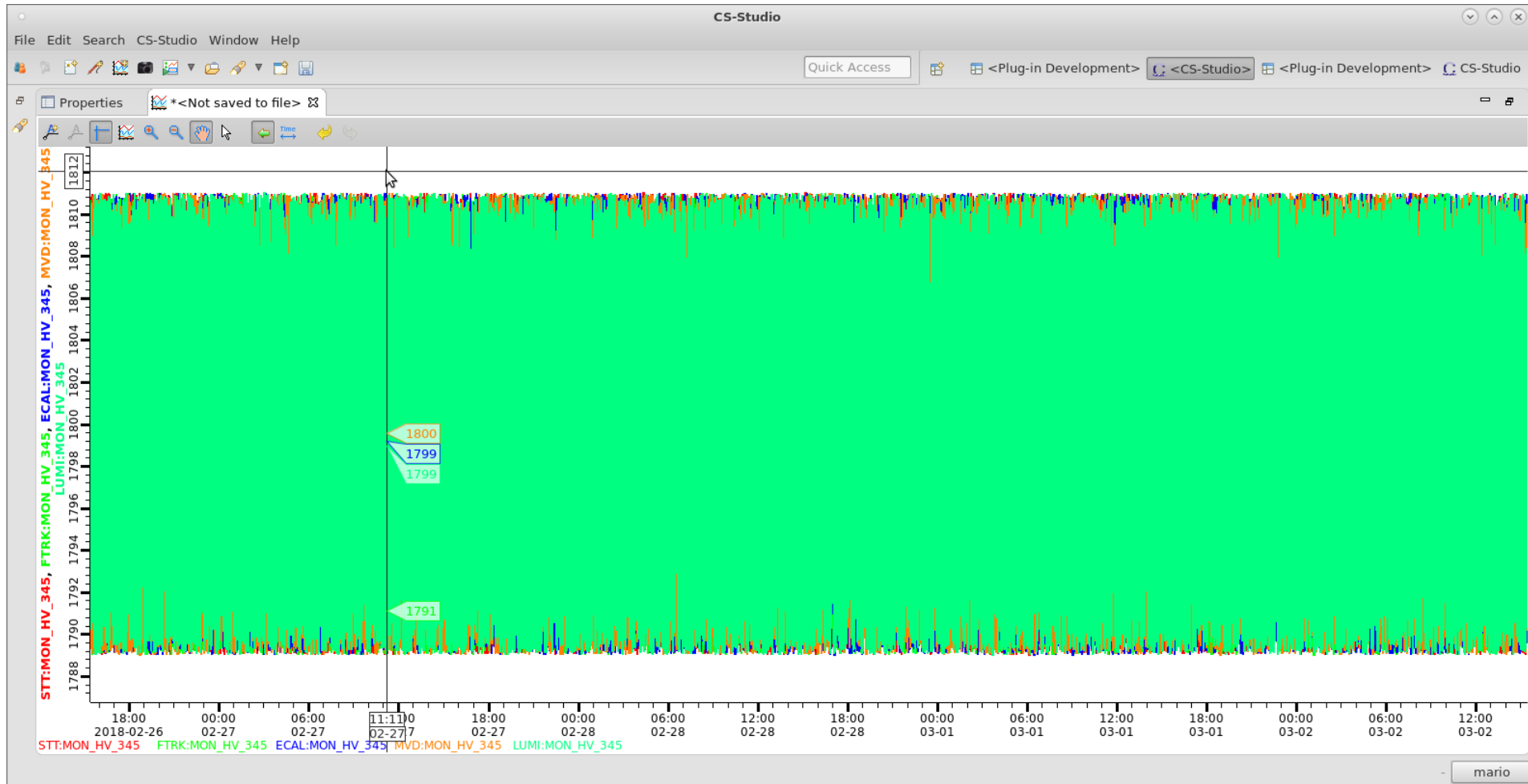


# IFIN-HH database testbed – Hardware Monitoring



# CSS – Data Browser retrieving from Cassandra database

CSS4 repo: <http://panda-repo.gsi.de/Mario/ControlSystemStudio.git>



- 5 PV's ( 1 / sub-system) – 5 days history (no failures in the IOC and/or Python Epics scripts )

# Custom code to query the Cassandra database

- Based on Datastax Cassandra drivers (C/C++, C#, Java, Node.js, Python, and Ruby)
- The drivers can be used to send CQL statements from client – Cassandra cluster;
- The tool is very powerful: retrieve, **manipulate or remove data**;
- Python script to retrieve PV in a given time interval is ready;

```
Output: STT:MON_HV_123 1520295168766427826
channel_access_scalar_double(value=1809.2962690444097, alarm_severity=1,
alarm_status=4, precision=2, units=u'V', lower_warning_limit=1795.0,
upper_warning_limit=1805.0, lower_alarm_limit=1790.0,
upper_alarm_limit=1810.0, lower_display_limit=0.0, upper_display_limit=0.0,
lower_control_limit=0.0, upper_control_limit=0.0)
```

- Usage of GnuPlot for PV visualization is under development.

# PV archiving hints from EPICS community

Spring 2017 EPICS Collaboration Meeting, Kyoto University

**Thursday, May 18, 2017**

**9:00-9:40 Bread and coffee**

**9:40-10:40 Tools and devices**

**9:40 CS-Studio Display Builder Update**

Kay Kasemir (ORNL/SNS)

**10:00 Lessons learned implementing a Channel Access gateway in Python with pyuv**

Daniel J. Lauk (PSI (Paul Scherrer Institute))

**10:20 Migration from Channel Archiver to Archiver Appliance in J-PARC MR**

Shuei Yamada (KEK/J-PARC)

**10:40 InfluxDB as Archive Data Storage**

Kay Kasemir (ORNL/SNS)



# Summary and Outlook

- Controls TDR preparation is moving ahead.
  - the scheme with (at least) one archiver machine / sub-system and a dedicated database cluster looks feasible.
- Some more work is needed for:
  - The evaluation of overall controls data throughput, storage and retrieval;
  - Detector Safety System;
  - Hardware proof of concept (dedicated meeting scheduled for 08.03 @ 14:30 ).