

# Multicomponent cluster management system for the computing center at IHEP

**V. Ezhova<sup>1</sup>, V. Kotliar<sup>1\*</sup>, A. Kotliar<sup>1</sup>, E. Popova<sup>1</sup>**

<sup>1</sup>Institute for High Energy Physics named by A.A. Logunov of National Research Center "Kurchatov Institute", RU-142281, Protvino, Moscow region, Russia

E-mail: {Victoria.Ezhova, Viktor.Kotliar, Anna.Kotliar, Ekaterina.Popova}@ihep.ru

\* Corresponding author

# topics

- ✓ Overview of the IHEP cluster management system which is a core part of the computing infrastructure.
- ✓ This system includes generic components for allocating and controlling over resources for different computing tasks, components for configuration management and software distribution on the computing hardware, components for monitoring and management software for the whole distributed infrastructure.
- ✓ The main goals of such system are:
  1. to create autonomic computing system with functional areas such as self-configuration, self-healing, self-optimization and self-protection
  2. to help to reduce the overall cost and complexity of IT management by simplifying the tasks of installing, configuring, operating, and maintaining cluster.
- ✓ Current implementation of the multicomponent cluster management system for the IHEP computing center will be shown.
- ✓ For the moment this system consists of event-driven management system, configuration management system, monitoring and accounting system and a chat-ops technology which is used for the administration tasks.

# Intro



# Computer center at IHEP 55 years of history



1965 – Minsk-2, Minsk-22, M-220, BESM-4, BESM-6,  
Minsk-32, EC-1040, EC-1045  
1972 – ICL  
1977 – DEC 10  
1991 - mVAX-II – mail  
1993 – Internet  
2003 – Grid-cluster  
2011 – 10Gb/s

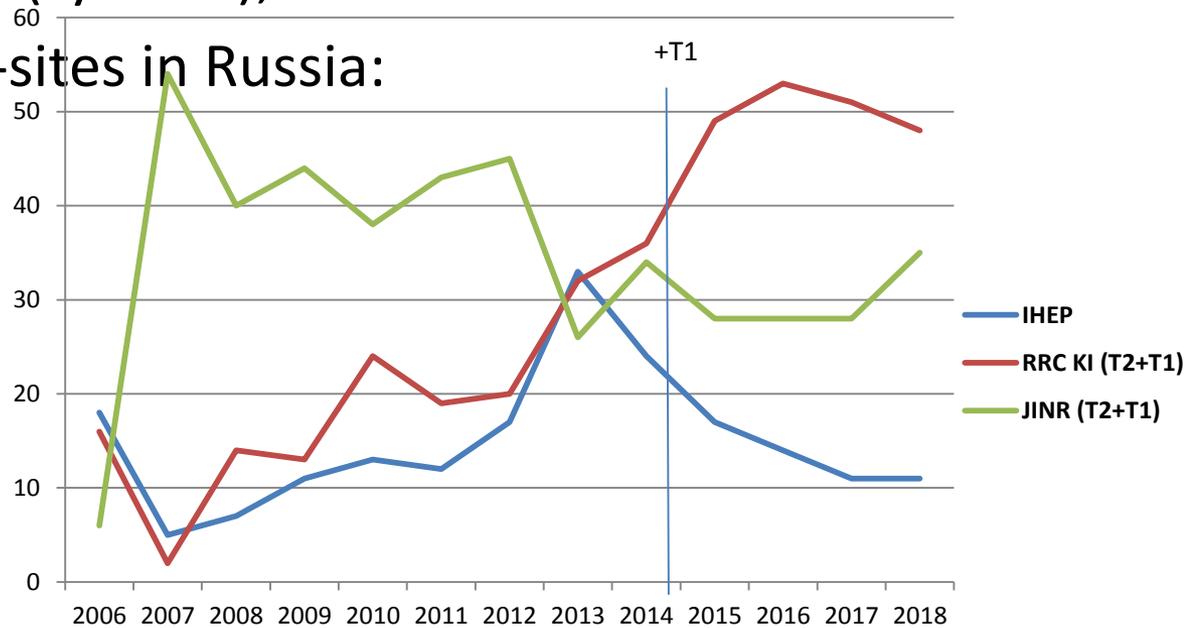


# Introduction

- RU-Protvino-IHEP site participates in the Worldwide LHC Computing Grid from very beginning since **2003**.
- In that time were installed and configured the first grid infrastructure services like CE, SE, WNs, UI on **16** two-core Pentium III 900MHz.
- After increasing network bandwidth to 100Mb/s, then to 1Gb/s and in the end to 10Gb/s we became one of the biggest Tier-2 site in Russia with **3k CPU (24000 HEP-SPEC06)** and **2PTB** disks space.
- In the present time our site serves for four LHC experiments (Atlas, Alice, CMS, LHCb) and many experiments inside of the Institute. We implement shared CPU schema that allows achieving 24x7 CPU resource usage.

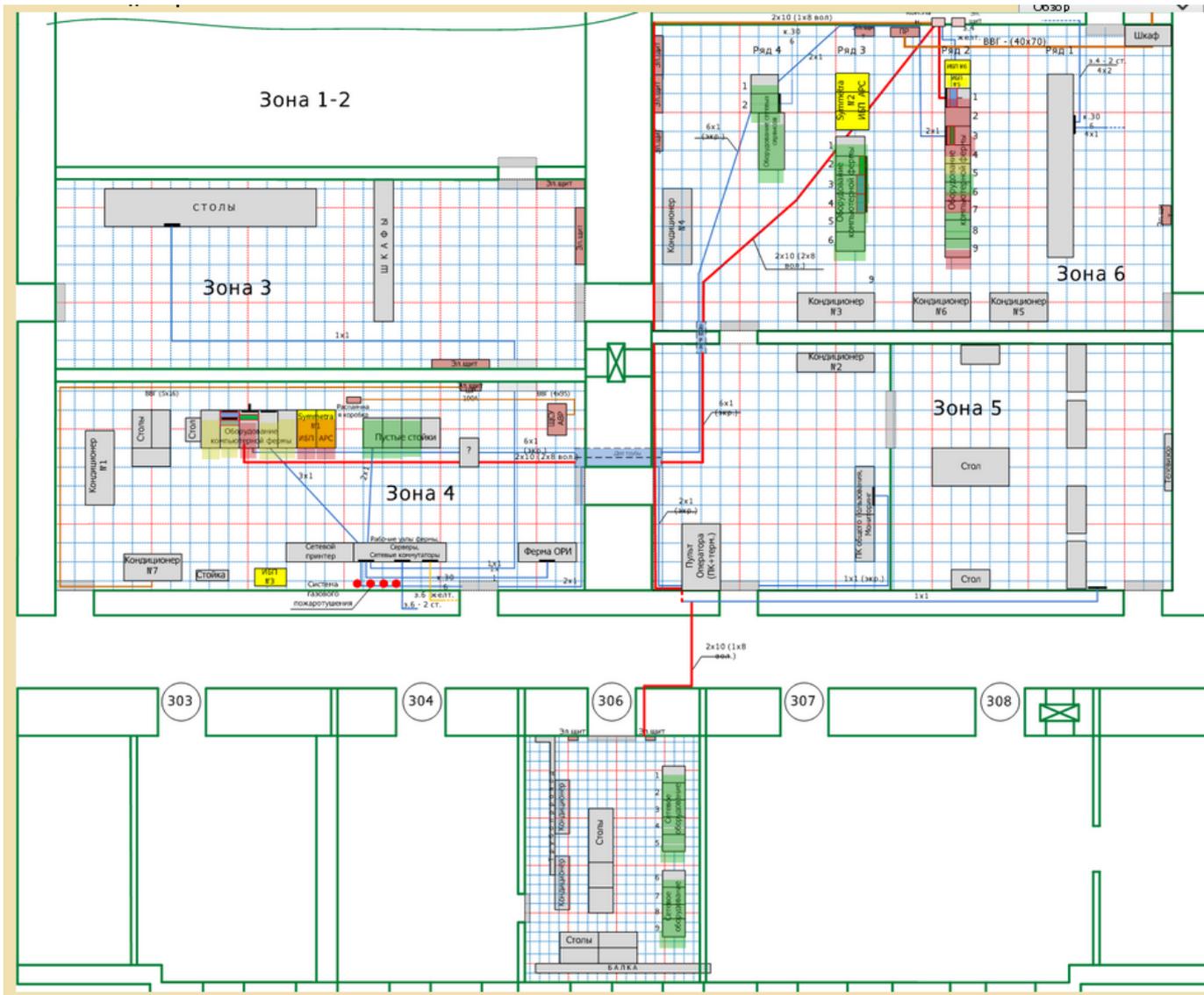
# Current status: resources

- 2828 CPU, 24390 HEP-SPEC06;
- 2182TB: Atlas 1185, CMS 395, Alice 297, IHEP 240, LHCb 65;
- 2x10Gb/s Internet channels  
(LHCONE shared with RDIG 10Gb/s);
- Manpower – 3 people (sysadm);
- One of three big grid-sites in Russia:



Contribution in % to RDIG by normalised CPU time  
Grid 2018

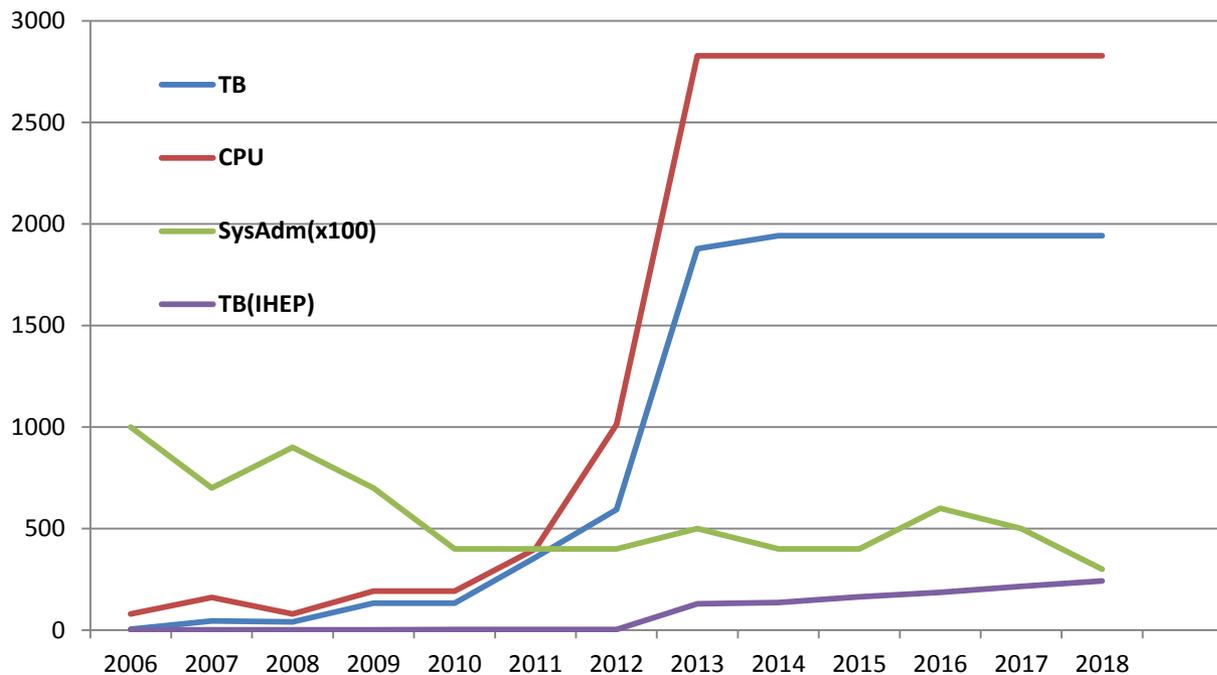
# Location



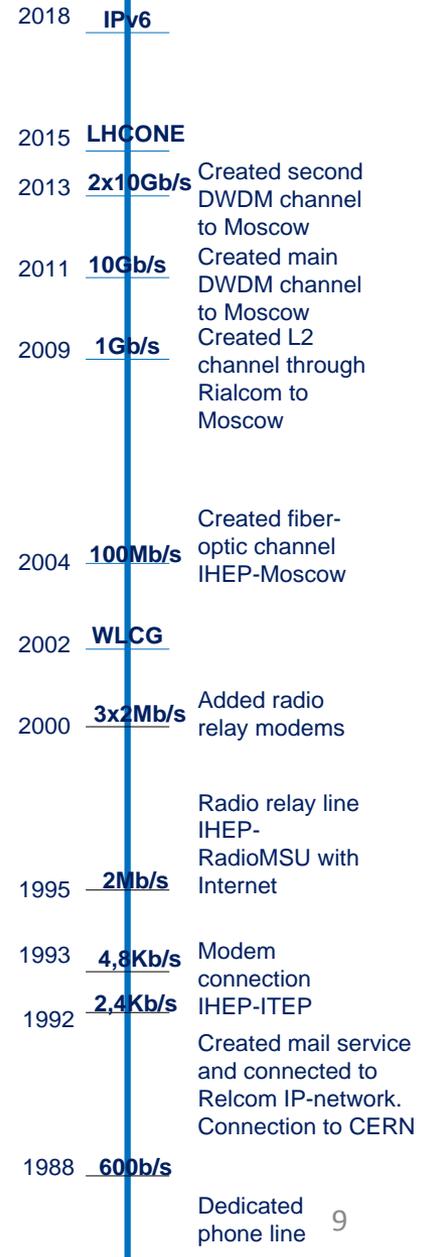
# Why management system?



# IHEP resources evolution



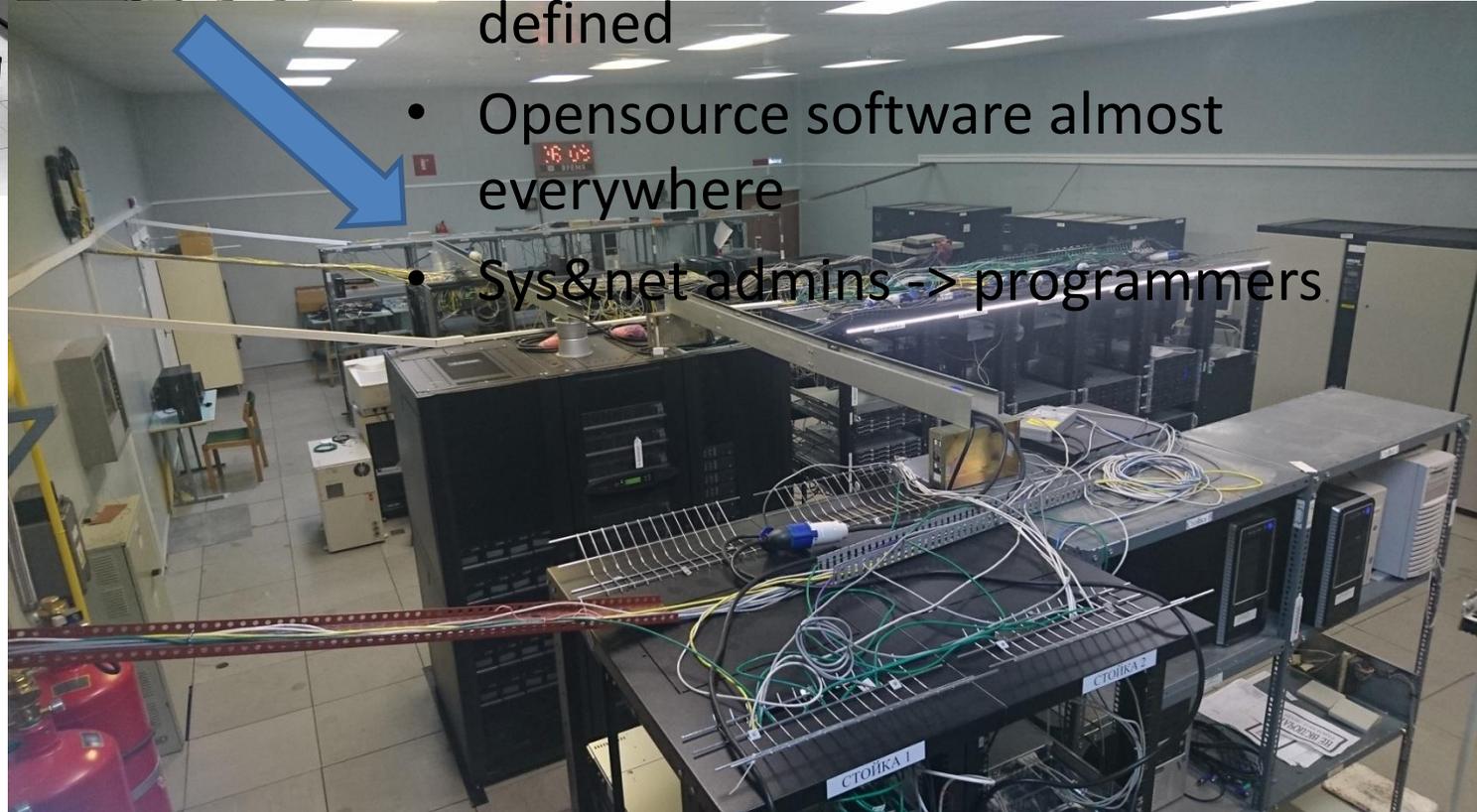
Evolution of the IHEP resources by year in TB, CPU, humans



# Evolution



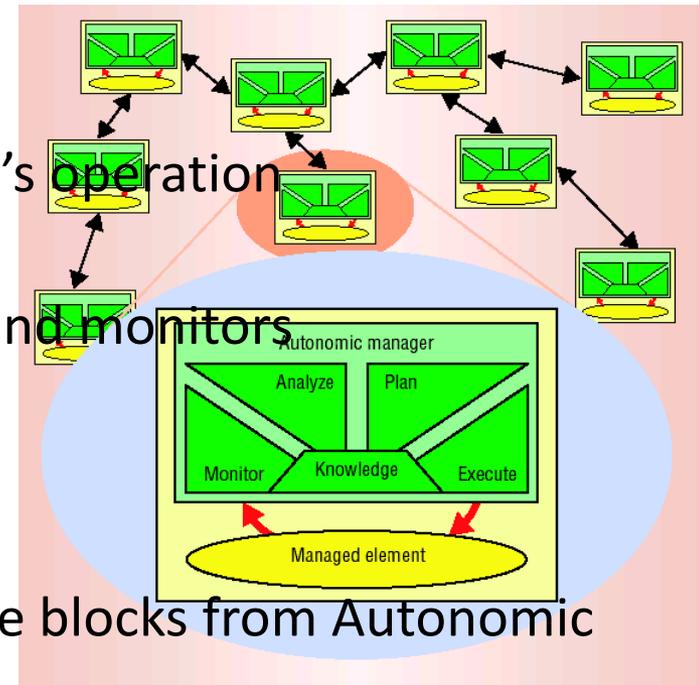
- Less humans more servers
- Generic hardware
- Zillions parameters for monitoring and tuning
- Almost everything software defined
- Opensource software almost everywhere
- Sys&net admins -> programmers



# What management system?

# What we want: theoretically

- Four aspects of self-management
  - Self-configuration
    - Configure themselves automatically
    - High-level policies (what is desired, not how)
  - Self-optimization
    - Hundreds of tunable parameters
    - Continually seek ways to improve it's operation
  - Self-healing
    - Analyze information from log files and monitors
  - Self-protection
    - Malicious attacks
    - Cascading failures
- Cluster management system should include blocks from Autonomic manager



# Cluster management system

The term cluster management defined as the set of common tasks required to run and maintain a cluster where the cluster is a set of computers with common goal . These tasks are:

- Management infrastructure;
- Provisioning
- Configuration and software package management
- Monitoring
- Operation management

# Cluster management software

## Free

- Apache Mesos, from the Apache Software Foundation
- Kubernetes, founded by Google Inc, from the Cloud Native Computing Foundation
- Heartbeat, from Linux-HA
- oneSIS
- OpenHPC
- Rocks Cluster Distribution
- Stacki, from StackIQ
- YARN, distributed with Apache Hadoop
- xCAT
- Warewulf
- Foreman

## Proprietary

- Bright Cluster Manager, from Bright Computing
- CycleCloud, from Cycle Computing
- HPE Performance Cluster Manager, from HPE
- IBM Tivoli System Automation for Multiplatforms, from IBM
- IBM Platform Cluster Manager, Community Edition (no charge infrastructure management)
- Microsoft Cluster Server, from Microsoft
- Scyld ClusterWare, from Penguin Computing

wikipedia

- Many for specific clusters: containers, HPC
- Not all features implemented
- For specific Linux distributions

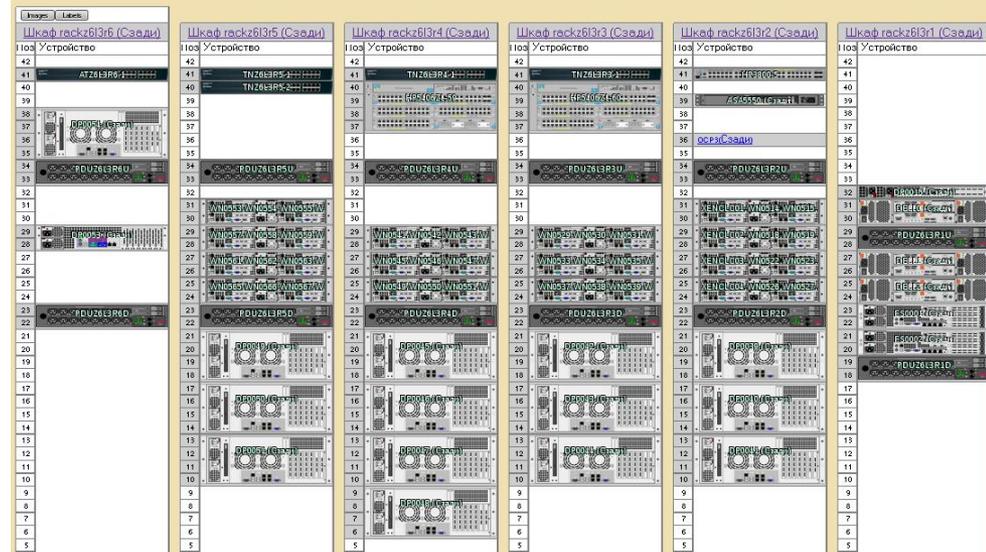
# Software for CMS

- Trade off for functionality over complexity must meet IHEP environment
- Has to support all OSs which are used on the IHEP cluster (Debian, RHEL like (CentOS, Scientific Linux))
- Has to be Open source with big community and has to be stable (not experimental)
- Has to be implemented in such way that different components could be installed independent on each other step by step
- Has to be pluggable that it is possible to customize functionality for IHEP needs

# Management infrastructure

# Management infrastructure

- Data Center Infrastructure Manager (opendcim) all information about the cluster
- Set of IPMI tools:
  - Linux ipmitool package
  - Supermicro IPMItool software
- PDU and UPS power control software or SNMP



# MI: Emergency soft cluster shutdown

## USB BIG RED BUTTON

Each working zone has its own BRB.

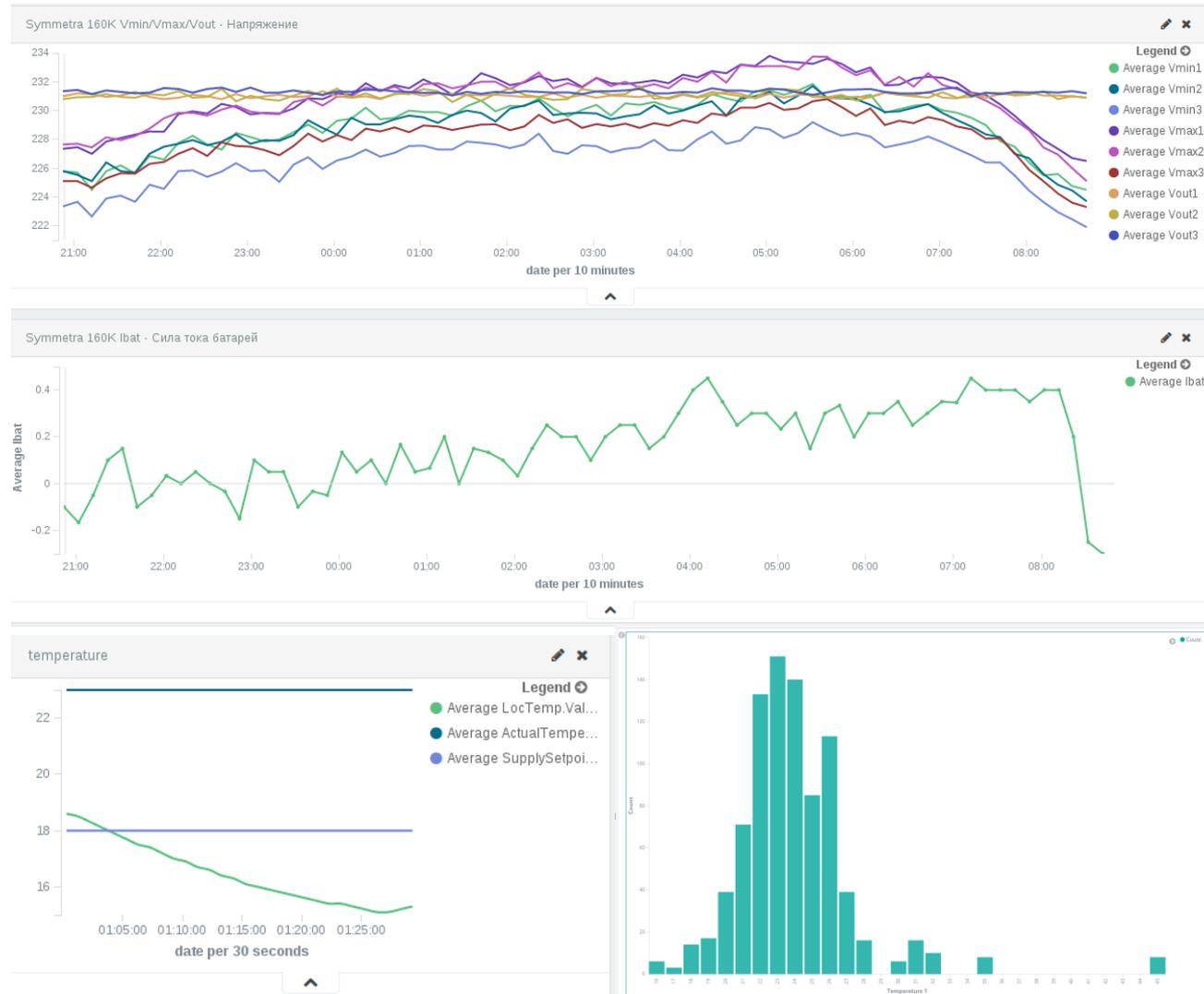
Each BRB – has a set of scripts to power off cluster as soft as possible in case of emergency. Usually that are cooling problems.

```
logger BUTTON PRESSED sending power off to WNs
echo "select label,DeviceType from fac_Device
where DeviceType='server' AND Cabinet in (select
CabinetID from fac_Cabinet where
DataCenterID='3' and ZoneID='1');" | mysql -h
dcim.ihep.su -u dcim -u dcim --password=*** dcim
| awk '{ if ($2=="Server") { sizeAr=split($1,ar,","); for
(i=1;i<=sizeAr;i++) print ar[i];}}' | grep WN | xargs -
iSERVER ipmitool -U ***-P ***-H SERVER chassis
power off
```



# MI: Cluster power usage and cooling

- Software has control to power utilization data
- Cooling parameters
- Power supplies parameters



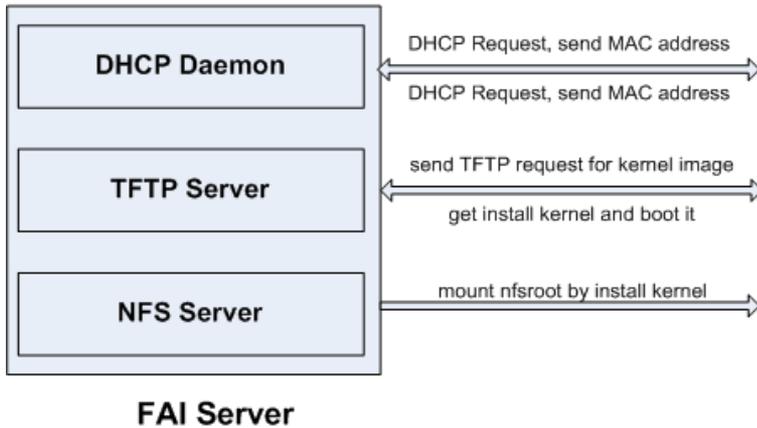
# Provisioning

# Provisioning

- PXE/FAI (Fully Automatic Installation) for WNs or SEs
  - Installs Debian GNU/Linux, Ubuntu, CentOS, SuSe, Scientific Linux, ....
  - Class concept supports heterogeneous configuration and hardware
  - Central configuration repository for all install clients
  - Full remote control via ssh during installation process
  - FAI runs on i386, AMD64, PowerPC, SPARC and IBM z10 mainframe
  - Fast automatic installation for Beowulf clusters
  - **Hooks can extend or customize** the normal behavior
- Ansible for unique custom build servers and preparing for containers
  - Simple/Easy-to-Learn
  - Written in Python
  - Agentless
  - YAML-Based Playbooks
  - Ansible Galaxy

# Provisioning

## 1. Boot via network card (PXE)

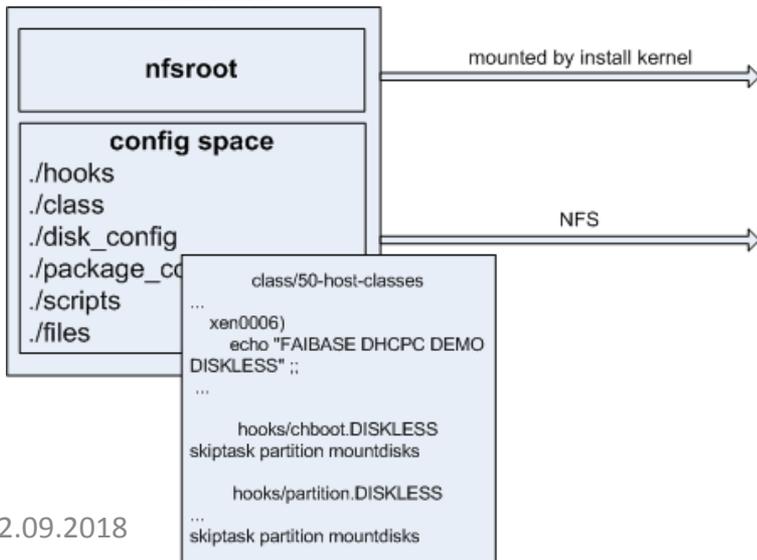


## Custom installation scripts

- Build with lustre support
- Used predefined mini images for systems
- Configuration leave for configuration systems
- Idea to use the same logic for virtual and physical servers

```
default linux
prompt 1
timeout 120
label linux
kernel mboot.c32
append demohost/xen-4.0-amd64.gz dom0_mem=2048MB noreboot ---
demohost/vmlinuz-2.6.32.40 nfsroot=10.165.1.25:/var/nfs/
demohost,v3,tcp,nolock ip=dhcp init=/linuxrc root=/dev/ram0 rw
ramdisk_size=4194304 --- demohost/initrd.img-2.6.32.40
```

## 2. Get Configuration and Installation



```
linuxrc
...
mkfs.ext2 /dev/ram0
mount -t ext2 -o rw /dev/ram0 /real_root

tar -C / -c --exclude=real_root* --exclude=proc* --exclude=dev/* --exclude=sys/*
--exclude=/lustre/* . | tar -C /real_root -x

echo "$new_host_name" > /real_root/etc/hostname;
hostname -F /real_root/etc/hostname;

cd /real_root
pivot_root . real_root
umount -n /real_root/proc >/dev/null 2>&1
exec chroot . bash -c 'umount -n -l /real_root >/dev/null 2>&1;
exec /sbin/init ' dev/console 2>&1

fi
```

# Configuration and sw management

# Configuration and software package management

- Tools that help manage system configuration (files, dirs, permissions, etc.) , software packages and software behavior
- Usually in pair with content management system for configuration
- Some software has centralized place not packaged (Ansys, Mathematica, ...) – software distribution systems

# Configuration and software package management: Puppet

- Extensible, declarative language
- Understands dependencies
- Good reporting
- Auto generation of documentation
- agent-server could be different versions! And different OSs (**huge benefit**)
- Easy to start using

# Configuration and software package management: Puppet

- One puppet server manage all WN nodes configuration and packages
  - WNs Scientific linux 6
  - Disks Debian Linux 7,8,9
  - UIs Scientific linux + CentOS7
- Have two branches on the server
  - Production
  - Pre production

```

    .
    ├── bin
    ├── etc
    │   ├── ihp
    │   │   ├── tools
    │   │   │   ├── ipmievd.nmi.watchdog.pl
    │   │   │   ├── ipmievd.nmi.watchdog.shutdown.pl
    │   │   │   ├── ipmievd.voltage.watchdog.pl
    │   │   │   └── ipmievd.voltage.watchdog.shutdown.pl
    │   └── manifests
    │       ├── init.pp
    │       ├── ipmievdNmiWatchdog.pp
    │       ├── ipmievdNmiWatchdogShutdown.pp
    │       ├── ipmievdvoltagewatchdog.pp
    │       └── ipmievdvoltagewatchdogShutdown.pp
    └── puppet.conf
    
```

301 directories, 482 files  
 root@puppet:~# tree /etc/puppet

- Use git for version control: in Plan switching to gitlab with full DevOps functionality
- Broadly use puppet self-healing functionality: force daemon to run, force mount to present etc.
- Special monitoring sensor to check that each node has connection to the puppet server
- We use our own cvmfs for software distribution

# Monitoring

# Monitoring and accounting

Depends on needs there are several monitoring systems:

- Nagios+check\_mk to check computer centre services: 20000 sensors
- Splunk and central syslog to get and analyse all logs: 591,328,937 events for 4 years
- collectl – realtime monitoring: 160 servers on one screen
- ESK for engineering infrastructure: 2 symmetra, 1 cooling system, 26PDU, over 100 PSU on the cluster
- Pmact (sflow) + cacti for network traffic monitoring
- Self-build accounting system

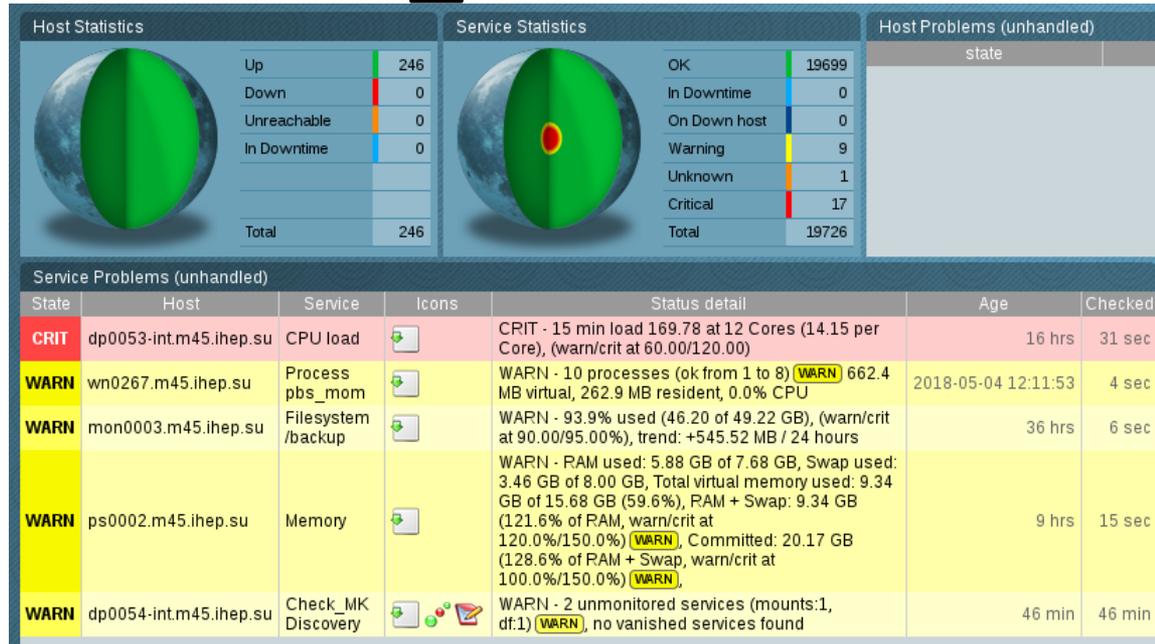
# Monitoring and accounting: Nagios+check\_mk

Nagios+check\_mk main monitoring system

- Live web interface with events history
- All checks configurable in one place
- Easy configuration for servers
- All Nagios plugins available
- Good API for external access
- Python based easy to write new checks even with performance data
- Powerful notification and plugin interface for integration with other systems

Good cmd line support! 😊

340 service checks/sec – 1Min for IHEP cluster



```

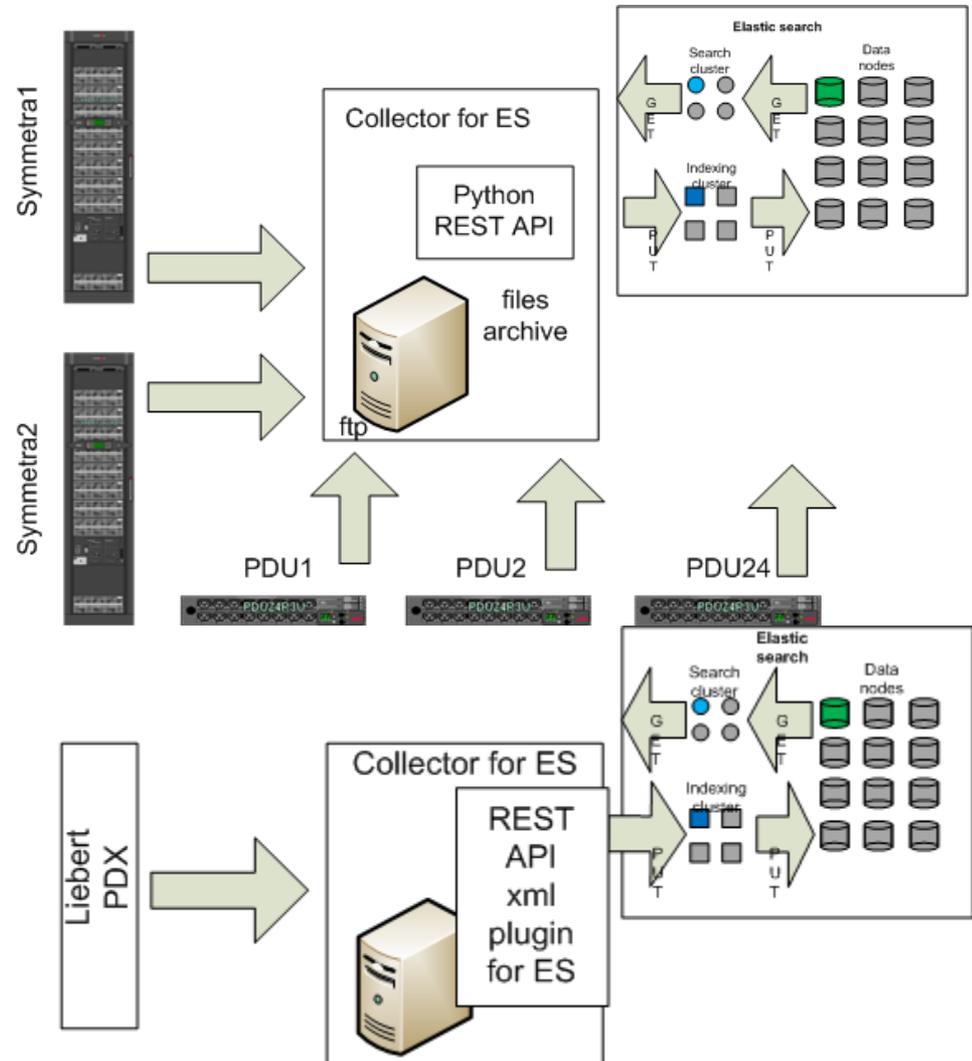
root@mon0004:~# cat /root/live.services.txt
GET services
Columns: host_name description plugin_output
root@mon0004:~# unixcat < /root/live.services.txt /usr/local/nagios/var/rw/live | grep SMA
RT | sed -e "/Temper/ d" | grep CRIT | head -n 2
dp0001-int.m45.ihep.su;SMART SATA_ST3120026AS_3JT4R9XY Stats;CRIT - Powered on: 94182 hour
s, Power cycles: 39, Reallocated sectors: 22(!) (was 21 during discovery), Spin retries:
0, Pending sectors: 1, UDMA CRC errors: 0
dp0002-int.m45.ihep.su;SMART SATA_ST3120026AS_3JT4RA79 Stats;CRIT - Powered on: 94701 hour
s, Power cycles: 48, Reallocated sectors: 8(!) (was 7 during discovery), Spin retries: 0,
Pending sectors: 0, UDMA CRC errors: 0

```

# Monitoring and accounting: ESK

## Elasticsearch and Kibana

- Store unstructured data
- We store something for somebody
- Easy analyze post factum
- Very powerful set of tool for data analyze
- Easy access data for external analyses
- Simple schema for data gathering even not for skilled sysadmins
- New technologies like ML and Docker available



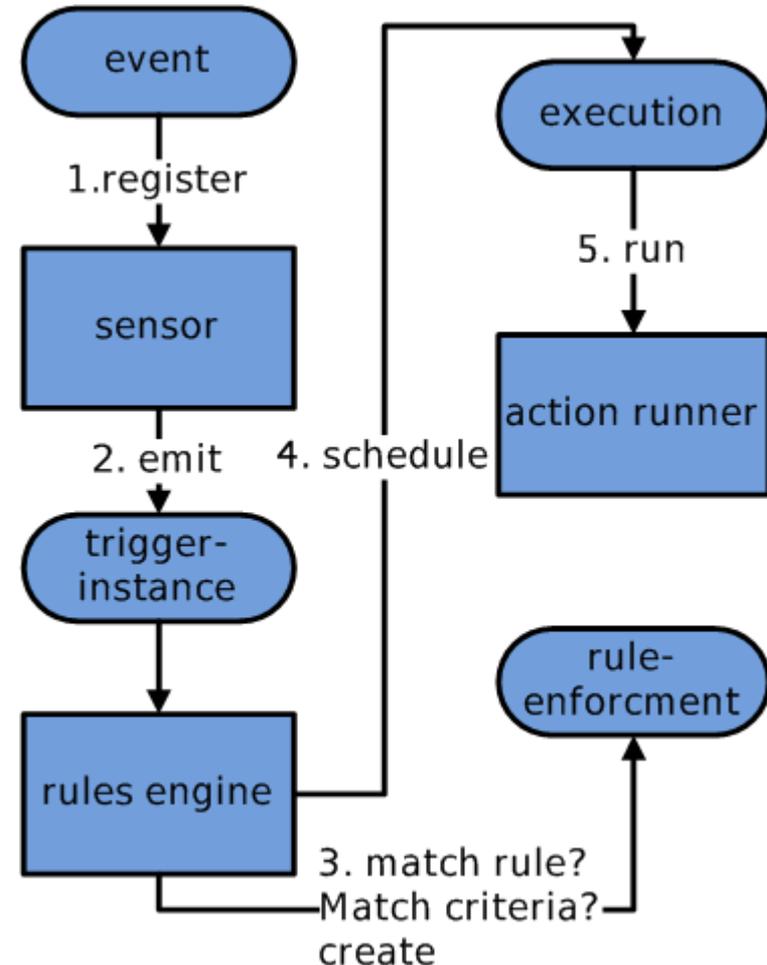
# Operation management

# Operation management: Automation system driven by events

Stackstorm event-driven system which allow to automate all actions in IHEP cluster distributed environment.

- Automated Remediation (self-healing + self-protection)
- Create automated tasks for cluster
- Comand executions on cluster with full log support
- Integration packs with many systems ( we use check\_mk, hubot)
- Possibility to create complex workflows for distributed systems
- Good API for interaction with system

**Automate Everything!**

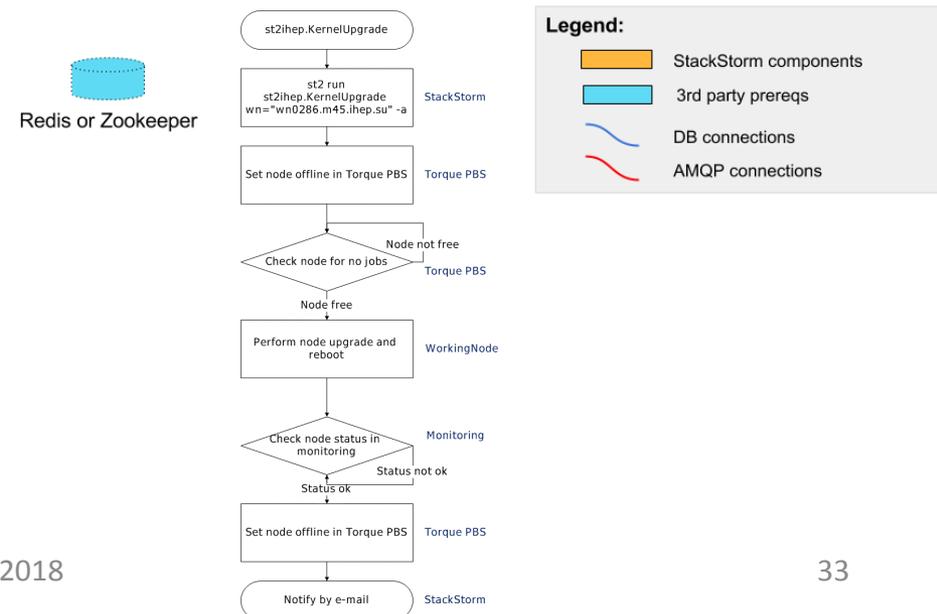
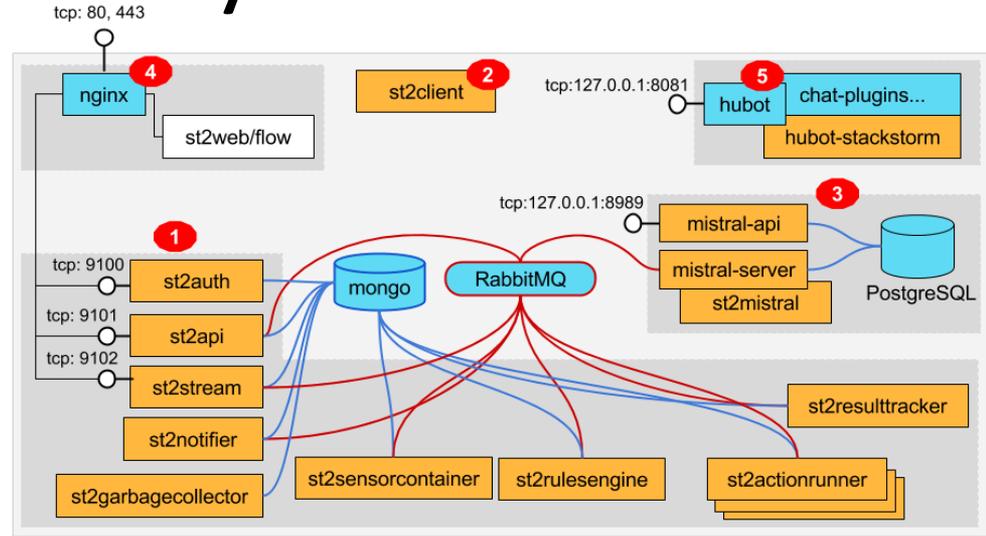


# Operation management :Automation

## system driven by events

Mostly rules and actions are used at IHEP.

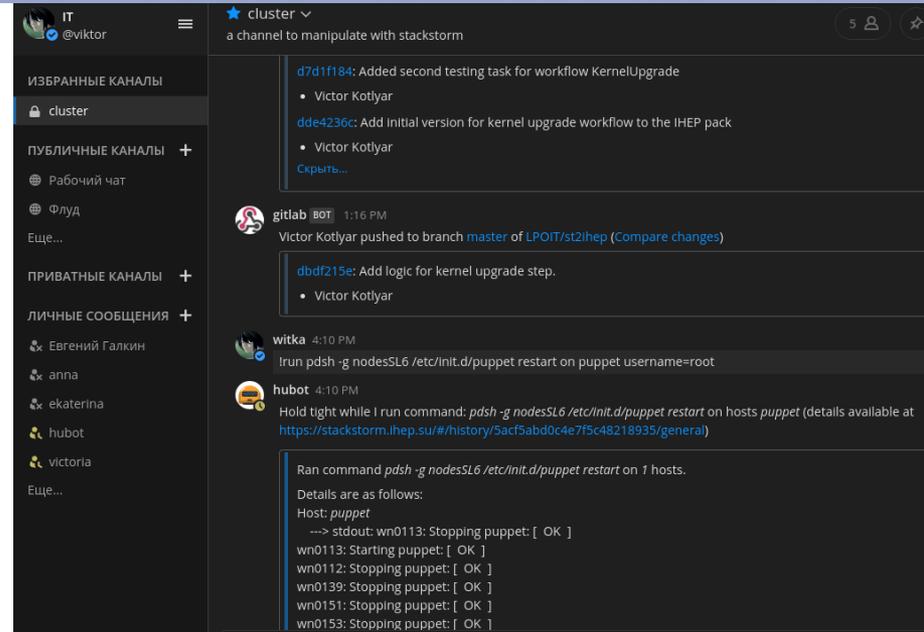
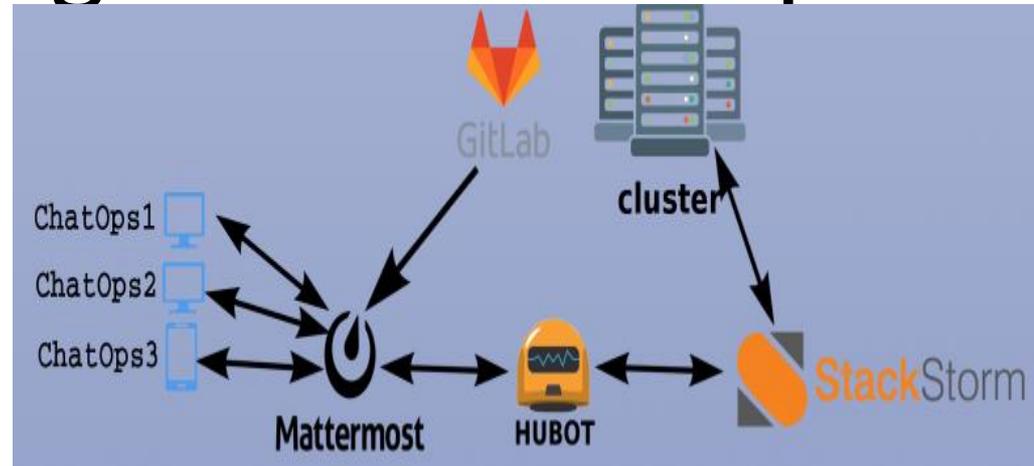
- Integration with check\_mk allowed to create actions for cluster events
- Mistral workflow are used for KernelUpgrade procedure (drain-upgrade-online)
- System allowed to mitigate software bugs (cvmfs automount bug on SL6)
- Many other small things – really good tool!



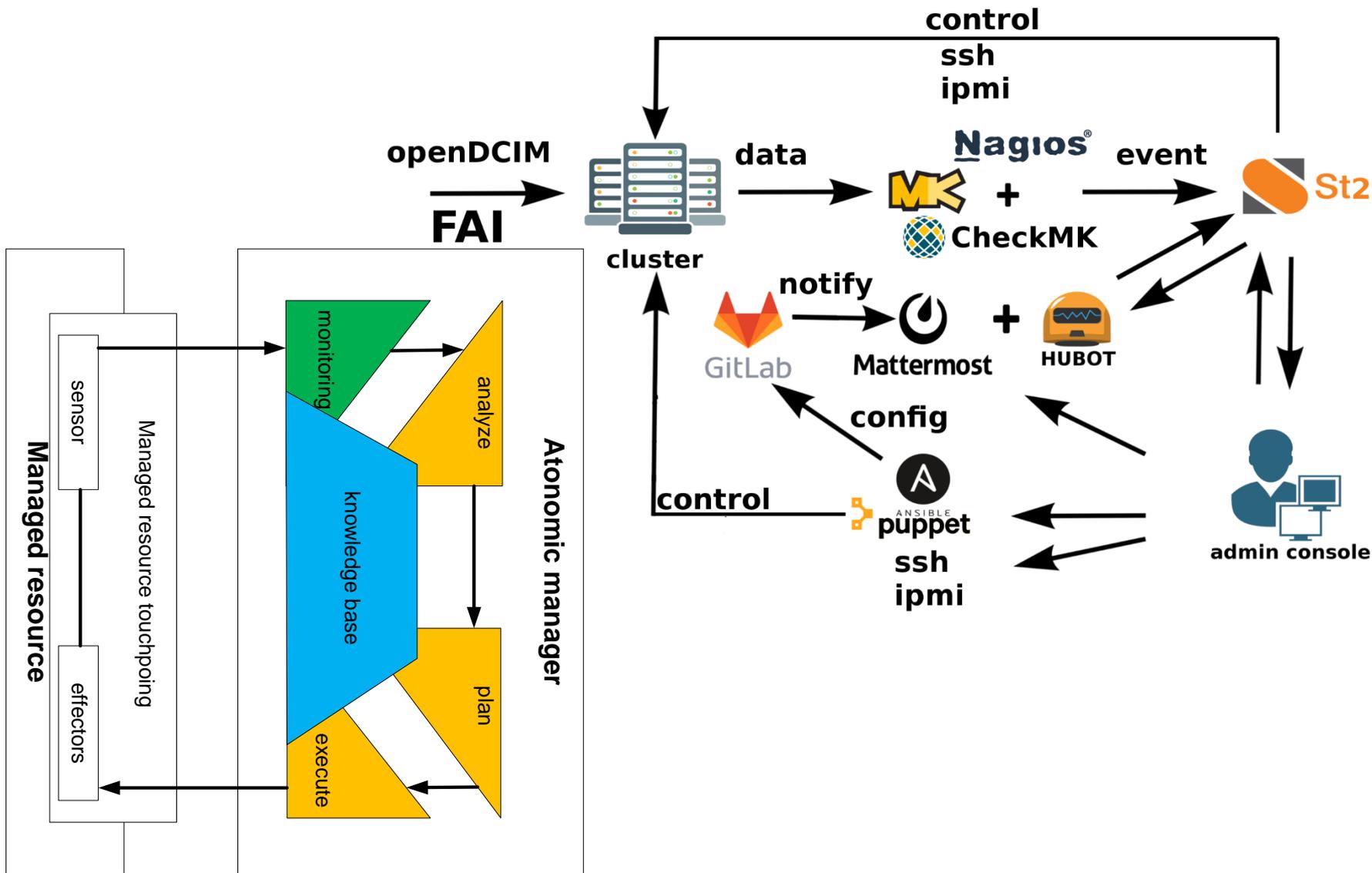
# Operation management :ChatOps

ChatOps is an operational paradigm where work that is already happening in the background today is brought into a common chatroom. Fully open IT operations.

- What should get done with actual work being done
- One place for any sysadmin's question
- One place for any worth notification
- Really collaborative work no more hidden alone sysadm
- Allow be in touch at any time from any place in convenient way (web, android, apple, linux, windows)
- Speedup new team member teaching!

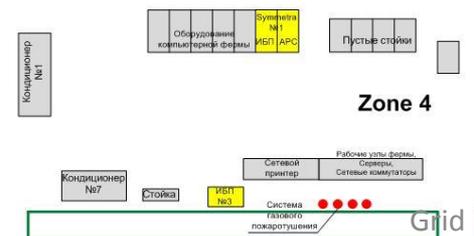
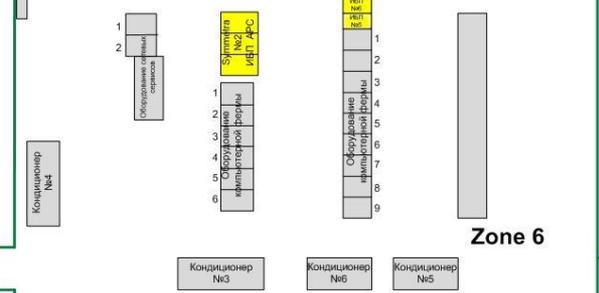
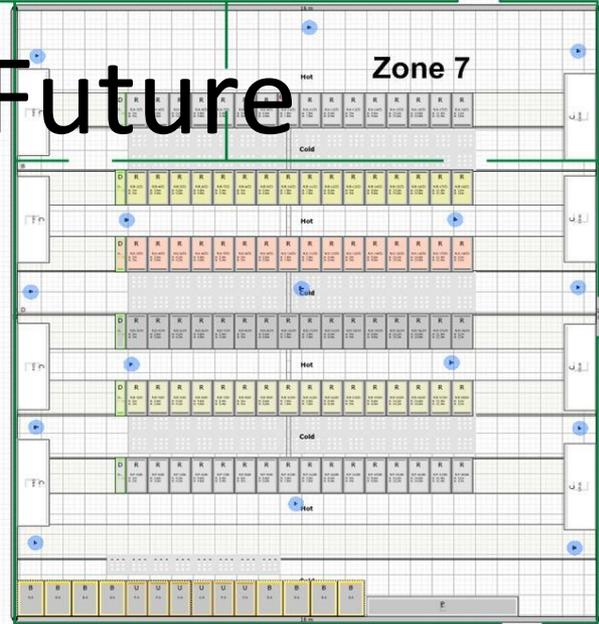


# CMS overview



# Conclusion & Future

Zone 1-2



- Almost all components for CMS in place
- Prepared for the future with less human but more hardware
- Prepared for the future with new software and hardware technologies
- Constantly need to create knowledge base for operations and automate everything
- Need to create allowed cluster states description DB + allowed actions ( Helix )
- Make cluster better: Effectiveness = Availability\* Reliability\* Maintainability\* Capability

$$[0.75 = 0.98 * 0.98 * 0.9 * 0.87]$$



Thank you!

Any questions?

# St2 overview



Component / Flow Diagram

