# Event-Driven Automation and chat-ops on IHEP computing cluster

Kotliar A.A., Kotliar V.V.
Institute for High Energy Physics named by A.A. Logunov of National Research Centre "Kurchatov Institute", Protvino, Russia

Stackstorm

Dealing with cluster-systems, you have multiple ordinary situations which can be solved using automation tools. Stackstorm is quiet a good event-driven system which helps to manage typical problems and to communicate with the IHEP cluster via chat-ops or any other extension. Stackstorm consists of four main parts: sensor, trigger, rule, action. Mostly rules and actions are used in daily operations at IHEP.
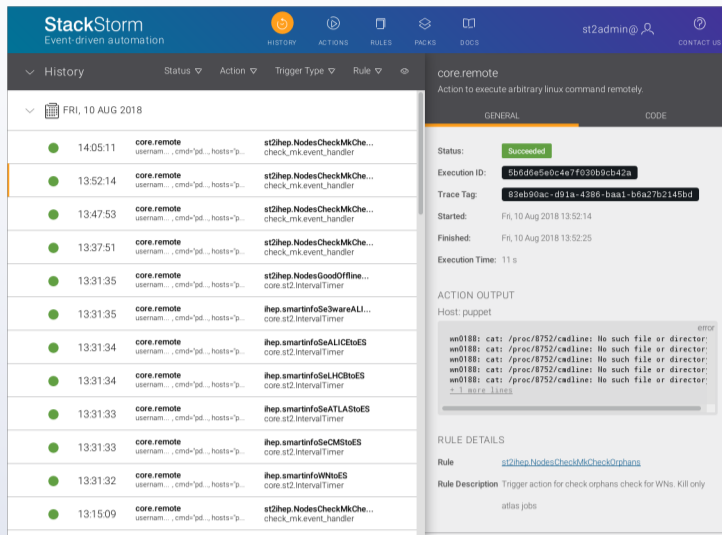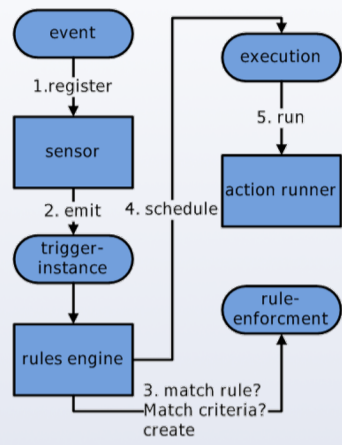
Events, triggers, rules and actions

Nagios and CheckMK

For monitoring troubles on cluster we use Nagios with CheckMK. CheckMK is a flexible monitoring framework which is really fast to set to watch the cluster work. So that to communicate with a Stackstorm we need just to do some little steps.
1) On Stackstorm server - install check_mk integration pack, write and register some triggers on yaml to handle events from CheckMK.
2) On CheckMk server - Install and set Stackstorm handler and set it to send some relevant events to Stackstorm server.

After that, just write a rule on Stackstorm server, which will describe what to do in case of a triggered trouble.

Event

Rule

```
name: "NodesCheckMkCheckOrphans"
description: "Trigger action for check orphans check for WNs. Kill only atlas jobs"
pack: "st2ihep"
enabled: true

trigger:
    type: "check_mk.event_handler"

criteria:
    trigger.service.state:
        pattern: "CRITICAL"
        type: "equals"
    trigger.host.name:
        type: "regex"
        pattern : "^wn0"
    trigger.service.desc:
        pattern: "check orphans"
        type: "equals"

action:
    ref: "core.remote"
    parameters:
        hosts: puppet
        username: root
        cmd: pdsh -w {{trigger.host.name}} "/usr/share/check-mk-agent/plugins/check_unparent |
            egrep \"(atl|sgmal)\" | egrep -o \"pid=[[:digit:]]+\" | awk -F = '{print \$2;}' | xargs
kill "
```

Trigger

```
st2 trigger get check_mk.event_handler
+-------------------+-----------------------------------+
| Property          | Value                             |
+-------------------+-----------------------------------+
| id                | 5ac5dff70c4e7f15a11b9129          |
| ref               | check_mk.event_handler            |
| pack              | check_mk                          |
| name              | event_handler                     |
| description       | Trigger type for check_mk events  |
| parameters_schema |                                   |
| payload_schema    |                                   |
| tags              |                                   |
| uid               | trigger_type:check_mk:event_handler |
+-------------------+-----------------------------------+
```

Action

ChatOps technology

ChatOps

ChatOps is an operational paradigm where work that is already happening in the background today is brought into a common chatroom. In doing so, you are unifying the communication about what work should get done with actual history of the work being done. Deploying code from the chat, viewing graphs from a TSDB or logging tool, or creating new Jira tickets are all examples of tasks that can be done via ChatOps.

Mattermost and hubot

Mattermost is a messaging workspace which has a possibility to use many different integrations. One of them is Hubot. Hubot is an open source bot, written in CoffeeScript on Node.js which can do things you need. You can ask it to post image to chat, to translate a text, to give a weather forecast, etc. In our case it is used to communicate with Stackstorm and to post answers to the common chat from the server.
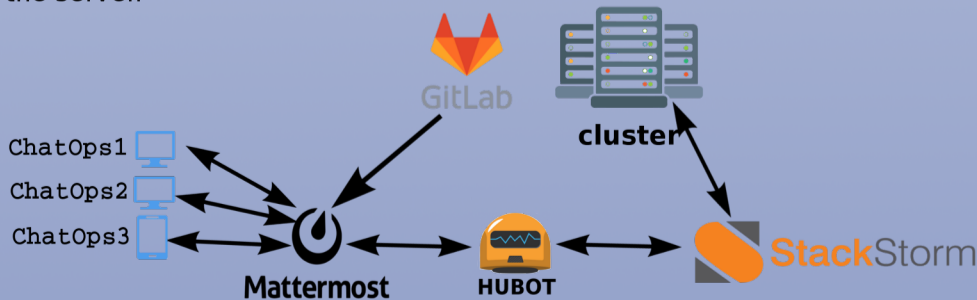
IHEP cluster management system

To manage IHEP computing cluster it was created a distributed multicomponent cluster management system. The main goal of such system is to reduce overall cost and complexity of cluster management by simplifying the tasks of configuring, operating and maintaining. Such system consists of monitoring system, system configuration managers, control version systems, chat-ops system and event-driven automation system.

In this implemenation, we use two control paths for the cluster management. First of all it is a classical system administrator approach where the administrator has full control for the cluster through a set of tools and second is a computing automation approach implemented with Stackstorm software.
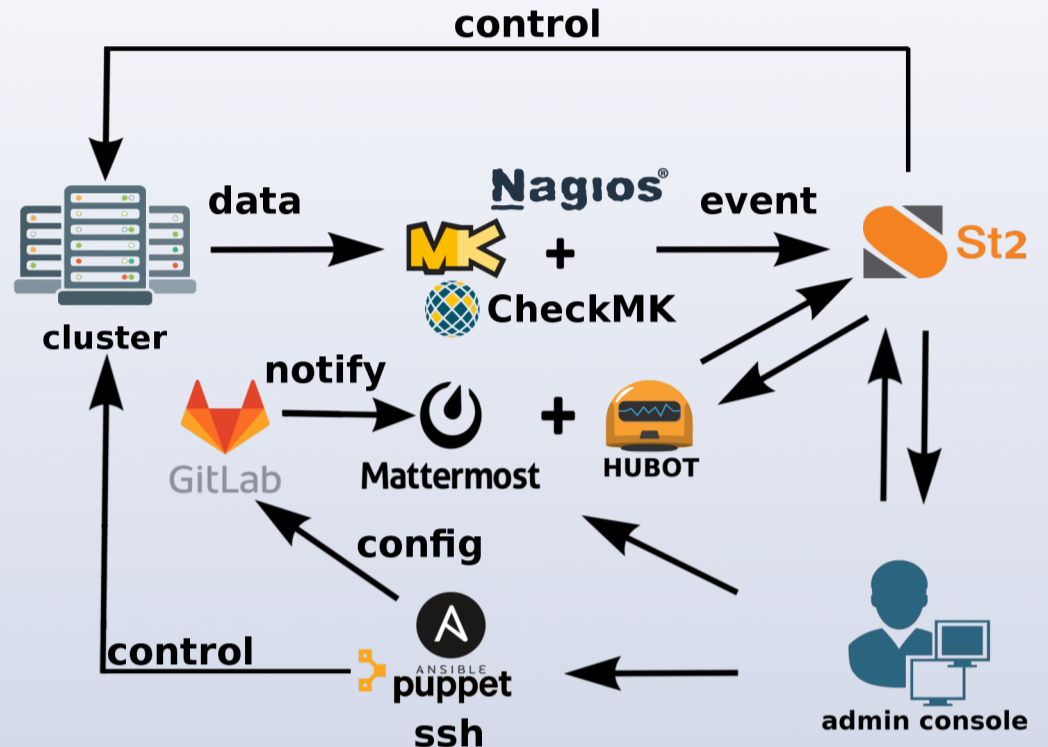

Complex management actions

Mistral workflow

Most cluster management processes consist of multiple distinct interconnected steps that need to be executed in a particular order in a distributed environment. A system administrator can describe such a process as a set of tasks and their transitions. After that, it is possible to upload such description to the StackStorm mistral service, which will take care of state management, correct execution order, parallelism, synchronization and high availability. In Mistral terminology such set of tasks and relations between them is called a workflow.

For several reasons kernel upgrade on Working nodes of the cluster has to be applied. Such upgrades consist of many steps which involve many different components of the cluster like batch system, monitoring system, working node itself. It was always a big headache for system administrators because such upgrade should be done as effectively as possible with minimum downtime. To make such task more effective and simple it was developed a stackstorm mistral workflow in the distributed cluster environment.


```
[root@stackstorm st2ihep]# st2 execution get 5b6d23fd0c4e7f7cc1be5c5e
id: 5b6d23fd0c4e7f7cc1be5c5e
action.ref: st2ihep.KernelUpgrade
parameters:
    wn: wn0189.m45.ihep.su
status: succeeded (537s elapsed)
result_task: Notify
result:
    failed: false
    return_code: 0
    stderr: ''
    stdout: Node wn0189.m45.ihep.su set online
    succeeded: true
start_timestamp: 2018-08-10T05:34:53.766709Z
end_timestamp: 2018-08-10T05:43:50.747885Z
+--------------------------+----------------------+----------------------+-------------+------------------------------+
| id                       | status               | task                 | action      | start_timestamp              |
+--------------------------+----------------------+----------------------+-------------+------------------------------+
| 5b6d23f0c4e7f7cc1be5c61  | succeeded (11s elapsed) | OfflineNode       | core.remote | Fri, 10 Aug 2018 05:34:55    |
| 5b6d240a0c4e7f7cc1be5c65 | succeeded (3s elapsed)  | CheckNodeForFree   | core.remote | Fri, 10 Aug 2018 05:35:06    |
| 5b6d240d0c4e7f7cc1be5c68 | succeeded (2s elapsed)  | PerformUpgrade     | core.remote | Fri, 10 Aug 2018 05:35:09    |
| 5b6d258a0c4e7f7cc1be5c6d | succeeded (4s elapsed)  | CheckNodeInMonitoring | core.remote | Fri, 10 Aug 2018 05:41:30 |
| 5b6d25ca0c4e7f7cc1be5c6f | succeeded (1s elapsed)  | CheckNodeInMonitoring | core.remote | Fri, 10 Aug 2018 05:42:34 |
| 5b6d26070c4e7f7cc1be5c71 | succeeded (1s elapsed)  | CheckNodeInMonitoring | core.remote | Fri, 10 Aug 2018 05:43:35 |
| 5b6d26090c4e7f7cc1be5c74 | succeeded (2s elapsed)  | OnlineNode         | core.remote | Fri, 10 Aug 2018 05:43:37    |
| 5b6d260c0c4e7f7cc1be5c79 | succeeded (1s elapsed)  | EmailNotify        | core.sendmail | Fri, 10 Aug 2018 05:43:40  |
+--------------------------+----------------------+----------------------+-------------+------------------------------+
```
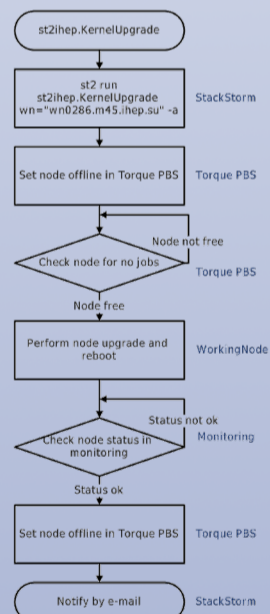

PLANS

* use event-driven automation for whole IT infrastructure
* automate everything for no human action required
* make system administrators become programmers
* try to use control system based on stackstorm not only for IT but for experimental sets for physics at IHEP