

PanDA centralized site exclusion

[0. Motivation](#)

[1. Centralized blacklisting system for PanDA: basic workflow](#)

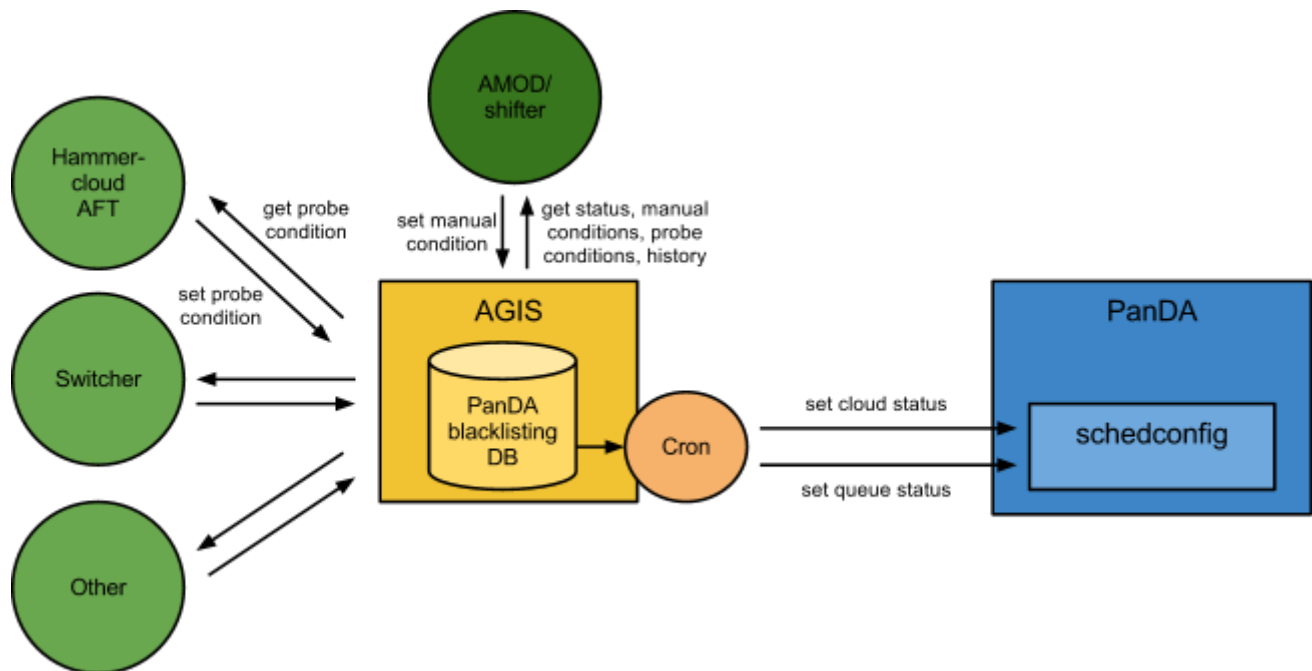
[2. Requirements](#)

[3. \(High level\) Implementation steps](#)

0. Motivation

We propose the implementation of a similar centralized blacklisting system for PanDA as exists in DDM. The reason is that currently there are different agents (Hammercloud and Switcher) that update the status of the PanDA queues concurrently and in occasions fight each other. A central system should get the information from the agents, combine it and update the status in PanDA consistently.

1. Centralized blacklisting system for PanDA: basic workflow



2. Requirements

- Possibility of blacklisting a queue/cloud for a given activity
 - **Parallel tables will be held for clouds and queues.** Cloud and queue status don't interfere with each other, we leave to PanDA the logic that is being carried out right now.
 - The activity will be set initially to a default/NULL, in the future we might be interested to set it to production or analysis
- There is a **manual condition** for each queue/cloud. It is the responsibility of the **operator** (AMOD/shifter) to set the value. The possible values are [**OFFLINE** | **TEST** | **BROKEROFF** | **ONLINE** | **AUTO**]

- There are **probe conditions** that would have to be written by the **automatic agents**. The possible values are [**OFFLINE | TEST | BROKEROFF | ONLINE**].
 - Notice that there is no AUTO probe condition, this is only a condition that an operator can set.
 - At the moment there are the Switcher and Hammercloud probes that act on the queue status. There are no probes for the clouds, but maybe they appear in the future.
 - Hammercloud and Switcher would have to be modified: instead of issuing the curl status update against PanDA, they would have to update the proposed system
- How to calculate the **final status** of a queue/cloud
 - The manual condition **overwrites** any probe condition
 - If a queue/cloud has manual condition **AUTO**, then the status of the queue/cloud will be set to the **most restricting** probe condition. (Order of restriction: OFFLINE > TEST > BROKEROFF > ONLINE)

MANUAL	PROBE 1	PROBE 2	STATUS
OFFLINE	*	*	OFFLINE
TEST	*	*	TEST
BROKEROFF	*	*	BROKEROFF
ONLINE	*	*	ONLINE
AUTO	OFFLINE	*	OFFLINE
AUTO	TEST	TEST BROKEROFF ONLINE	TEST
AUTO	BROKEROFF	BROKEROFF ONLINE	BROKEROFF
AUTO	ONLINE	ONLINE	ONLINE

- Information to store each time a condition is modified
 - **Comment.** The comment passed to schedconfig will be the first one available in the **order**: manual, switcher, hammercloud.
 - **Timestamp**
 - **IP&DN of the submitter.** Only people with production role (or some other restriction) should be allowed to change the values of a condition
 - **Expiration date**
- The final status of the site will be periodically pushed/pulled to **schedconfig**
 - **There are NO changes needed in any internal logic in PanDA.**
 - **The existing commands to write the information into schedconfig will be used.**
 - Initially a daemon would update schedconfig e.g. every 5 minutes.
 - Other models could be considered, such as schedconfig pulling the information or such as triggering updates immediately when the situation of a queue/cloud changes
- The system will keep the history of changes in manual and probe conditions
- The tables will be included in the AGIS schema.

3. (High level) Implementation steps

1. Create database tables in AGIS schema. The tables are similar to those implemented in the DDM centralized blacklisting system
2. Implement server side methods to access information from DB
3. Implement client CLI&API using AGIS clients as model

4. Modify Hammercloud and Switcher. An integration phase would be desirable: HC and Switcher write both into schedconfig and the proposed system. The proposed system would not populate schedconfig, but we would monitor that the output is correct.