Proposal

See here.

Instructions of configuration on AGIS

Note: Currently all the following steps ONLY work for Harvester instance CERN_central_A and CERN_central_B.

Add new PQ to Harvester on AGIS for auto queue configuration

Open the AGIS page of the PQ.

Make sure all steps on AGIS of a Harvester PQs are done first (e.g. pilot manager = Harvester, and some UCORE or UPS setup if necessary).

Choose the step “Add a Normal Grid PQ” or “Add a PQ Requiring Special Template” below (yet not both!). And see if one needs to go through the optional “Add Associate Parameters”.

Add a Normal Grid PQ

Among SchedConfig parameters, fill in field "harvester" and "workflow" . E.g.

Thus, now the minimum auto pq configuration on AGIS looks like this:

harvester:	**CERN_central_B/Harvester (CERN-PROD)**

workflow:	**pull_ups**

For "harvester", choose an harvester instance, typically either CERN_central_A/Harvester (CERN-PROD) or CERN_central_B/Harvester (CERN-PROD)

For “workflow”, choose a workflow among Push, Pull, and Pull_UPS. For UPS (unified pilot streaming) PQs, choose Pull_UPS.

Note that the “harvester_template” must be left blank, otherwise it will override the template.

Done!

Associate parameters are not needed for most grid PQs (taking default values from template).

Description: If harvester_template is blank, then harvester will take the default template name as "<type>.<workflow>". For instance, default template of Taiwan-LCG2-HPC2_Unified in the example above will be "production.pull_ups", and this template is defined on the common template file on github.

Add a PQ Requiring Special Template

Only PQs which cannot use default templates need this setup:

Among SchedConfig parameters, fill in field "harvester" and "harvester template" . E.g.

harvester:	**CERN_central_B/Harvester (CERN-PROD)**

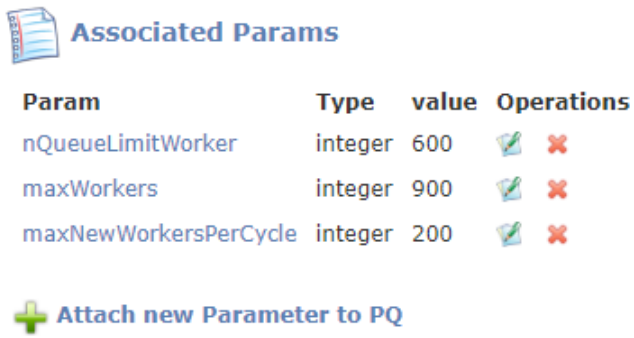
harvester template:	PRODUCTION_PULL_UPS_SHAREDFS_TEMPLATE

For "harvester", choose an harvester instance, typically either CERN_central_A/Harvester (CERN-PROD) or CERN_central_B/Harvester (CERN-PROD)
For "harvester template", insert the string of a template name in the common grid queueconfig template on GitHub or local config file. (More templates can be added to GitHub in the future if necessary.). The “harvester_template” field overrides the name template of default from PQ type + workflow.

Add Associate Parameters (Optional)

Do this only when it is not enough to work with parameters in default template on GitHub. For new normal Grid PQ, better to skip this section and start with default to see how it goes, and then ramp up/down parameters via the approach introduced here to tune stuff.

Under Associated Params, one can add harvester queue parameters. Click "Attach new Parameters to PQ" and then insert param and value. E.g.



Currently only parameters following for limits of job/workers are available:

- For jobs: nQueueLimitJob, nQueueLimitJobRatio, nQueueLimitJobMax, nQueueLimitJobMin
- For workers: nQueueLimitWorker, maxWorkers, maxNewWorkersPerCycle, nQueueLimitWorkerRatio, nQueueLimitWorkerMax, nQueueLimitWorkerMin

Coda

Then, after 20~30 minutes (considering cacher update period ~ 10 min. + qconf object update period ~ 10 min.), the harvester instance specified shall fetch the information on AGIS and start to submit workers for the PQ.

Or, if one does not want to wait, one can manually refresh harvester cacher and queue configurations on harvester node via harvester-admin commands (new feature after 181124):

```
$ <dir_of_harvester-admin>/harvester-admin cacher refresh
$ <dir_of_harvester-admin>/harvester-admin qconf refresh
```

Done.

Move configuration of existing PQ in Local queueconfig file to AGIS

Do the same steps in "**Add new PQ to Harvester on AGIS**". Basically one can easily translate JSON object of a normal GRID PQ in local queueconfig file to configuration on AGIS.

E.g. Actually the example of AGIS configuration above is translated from the following PQ in local queueconfig file on CERN_central_B instance:

```
"Taiwan-LCG2-HPC_Unified":{
        "queueStatus":"online",
        "templateQueueName":"PRODUCTION_PULL_SHAREDFS_TEMPLATE",
        "prodSourceLabel":"managed",
        "nQueueLimitWorker":600,
        "maxWorkers":900,
        "maxNewWorkersPerCycle":200,
        "runMode":"slave",
        "mapType":"NoJob"
}
```

where the "runMode" and "mapType" are defined in the template "PRODUCTION_PULL_**UPS**_SHAREDFS_TEMPLATE" on GitHub already; the "queueStatus" is always online if configured on AGIS (If one wants it offline on harvester instance, modify "pilot manager" field to be something else than "Harvester")

After 20~30 minutes, remove the PQ in local queueconfig json file, and reload harvester service (or wait a few minutes more).

One can then check whether the PQ is still on the harvester node (coming from AGIS) with harvester-admin command. E.g.

```
[root@aipanda173 ~]# /usr/local/bin/harvester-admin qconf dump -J Taiwan-LCG2-HPC_Unified
{
"Taiwan-LCG2-HPC_Unified": {
        "allowJobMixture": false,
        "configID": 963,
        "ddmEndpointIn": null,
        "getJobCriteria": null,
        "mapType": "NoJob",
        "maxNewWorkersPerCycle": 200,
```

```
        "maxSubmissionAttempts": 3,
        "maxWorkers": 900,
        ...
}
```

If so, then done.

Configuration on Harvester

To make harvester work with auto queue configuration, one needs some lines in harvester.cfg:

```
[qconf]

configFromCacher = True

queueList =
 DYNAMIC

resolverModule = pandaharvester.harvestermisc.info_utils

resolverClass = PandaQueuesDict


[cacher]

data =
 ...
 panda_queues.json||(URL of remote schedconfig JSON)
 queues_config_file||(URL of remote queueconfig JSON)
```

On CERN_central_A,B these URLs are:

http://atlas-agis-api.cern.ch/request/ddmendpoint/query/list/?json&state=ACTIVE&site_state=ACTIVE&preset=dict&json_pretty=1&type[]=OS_LOGS&type[]=OS_ES

https://raw.githubusercontent.com/PanDAWMS/harvester_configurations/master/GRID/common_grid_queueconfig_template.json

One can skip the line of “queues_config_file” if no remote queueconfig needed.

FAQ

What if the same PQ set both on AGIS and in local queueconfig file?

Priority of location to define the parameters (descending):

1. PQ in Local queueconfig json file on harvester node (LQ)
2. Associated Params on AGIS (DQ)
3. PQ on remote URL source (RQ). So far no RQ is set on GitHub but still possible
4. Template in Local queueconfig json file on harvester node (LT)
5. Template on GitHub (common grid queueconfig template), or other remote URL source (RT)

See here for detailed explanation of how harvester works for auto queue configuration, which should cover all similar questions.

What if one inserts erroneous values in queue configuration (on AGIS or in local file)?

In principle, encountering invalid queue configurations of a PQ, harvester will:

- Drop the problematic PQ (offline and unavailable)
- Log error/warning message in panda-queue_config_mapper.log
- Keep running and serving the valid PQs and existing workers, without breaking anything

E.g.1

If “monitor” and “preparator” of Taiwan-LCG2-htcondor-score are not defined in AGIS, local file, or any templates, then this queue is consider invalid and will be unavailable on harvester:

```
# /opt/harvester/local/bin/harvester-admin qconf dump -J Taiwan-LCG2-htcondor-score

ERROR : Taiwan-LCG2-htcondor-score is not available
```

```
{}
```

And in panda-queue_config_mapper.log there can be:

```
2018-11-22 19:46:21,072 panda.log.queue_config_mapper: DEBUG
QueueConfigMapper.load_data : queue Taiwan-LCG2-htcondor-score comes from LQ

2018-11-22 19:46:21,076 panda.log.queue_config_mapper: ERROR
QueueConfigMapper.load_data : Missing mandatory attributes preparator,monitor .
Omitted Taiwan-LCG2-htcondor-score in queue config
```

which shows the queue comes from LQ (local queue). Thus, one should check if something is fishy in local config file.

E.g.2:

One inserts the name of a non-existing harvester template on AGIS, say:

PanDa Queue name: Taiwan-LCG2-htcondor-score

harvester template:		**xxxxxxxxxxxxxxxxxxxxx**

And assume nothing in local config file overrides it.

Then, this queue is consider invalid and will be unavailable on harvester, and in panda-queue_config_mapper.log there can be:

```
2018-11-22 19:46:20,935 panda.log.queue_config_mapper: WARNING
QueueConfigMapper.load_data : Invalid templateQueueName "xxxxxxxxxxxxxxxxxxxxx" for
Taiwan-LCG2-htcondor-score (DQ). Skipped
```

which shows the queue comes from DQ (dynamic queue). Thus, one knows they should check the setup of this PQ on AGIS.

If you find harvester gets broken with an erroneous queue configuration or error/warning message not informative enough, report the issue to developers.

Can I determine the PQ to be UCORE or UPS queue elsewhere than AGIS? Say, in local queueconfig file on Harvester?

Short answer: No.

To make UPS (unifiled pilot streaming) working on a PQ, both PanDA server and Harvester need to know the PQ is a UPS queue (so PanDA computes how many workers of each resources type to submit, and harvester only submits workers passively according to commands from PanDA). Thus, the information of UPS setup of the PQ must share across both PanDA and harvester: AGIS is ideal for this. It makes no sense to set a PQ to be a UPS queue unilaterally.

As to UCORE (unified), well, harvester can still to submit workers without knowing the queue is UCORE or not: UCORE information only influence the computation of resource requirement of a worker. But there is no reason not to keep information about UCORE of the queue on AGIS. Actually even for a MCORE or SCORE queue, Harvester does not need set the string “MCORE” or “SCORE”. All one can do to define the number of cores is:

- Set nCore explicitly (fixed value) of the PQ in queue configuration,
- Or take the resource requirement (corecount) of the job (default) in push,
- Or take the capacity (corecount) of the site from AGIS (default) in rest cases

And the same situation holds for a UCORE queue. (and similar for memory requirement etc.)
So UCORE can only be reasonably run in Push where ncore of workers defined by jobs, or Pull_UPS where ncore of workers decided by PanDA. (In the case a UCORE running in pure Pull mode, Harvester will end up submitting workers with fixed ncore, either according to explicitly set nCore, or site corecount from AGIS)
Since AGIS information (site corecount, etc.) can be referenced for a queue no matter what, set the UCORE tag queue on AGIS is natural.