

Hacking ARC

NeIC NT1 Operator
Christian Sørtrup
<soettrup@ndgf.org>

2019-04-01
ISGC, Taipei

Overview

- Warmup – improving performance on VMs
- Backends – the hard way (tm)
- Runtime environments – The right way
- Auth plugins – needed to get RTEs to work (ARC5)



Warmup - Shared mem for LDAP

- Heavy I/O load on UCPH frontends
 - Caused by the infosys writing to ldap repeatedly
 - Information recreated on restart
- Solution: tmpfs
 - Make sure you have a shmem partition:

```
cat /etc/fstab  
tmpfs /dev/shm tmpfs defaults,size=8G 0 0
```
 - Link the slapd tmp dir:

```
ln -s /dev/shm /var/lib/arc/bdii/db
```
 - At UCPH we also put sssd cache in shmem (slurm ldap user lookups)

```
ln -s /dev/shm /var/lib/sss/db
```



Hacking backends

- Backends consist of three shell scripts (Slurm):
 - submit-slurm-job: job script creation and preprocessing
 - scan-slurm-job: job post-processing
 - cancel-slurm-job: killing jobs
- Pro
 - Full control
- Con
 - Must be reapplied every update



Backends - Example

- UiO patch for `ce01.grid.uio.no`
 - Adds disk & env constraints
 - Edits job memory request
 - Adds accounting info
- Uses ansible to update ARC and reapply patch
 - Drop `<nordugrid-discuss@nordugrid.org>` a line if interested

```
--- /usr/share/arc/submit-SLURM-job      2018-08-10 04:16:22.000000000 +0200
+++ /usr/share/arc/submit-SLURM-job.patched  2018-08-10 13:22:08.093643986
+0200
@@ -94,6 +94,31 @@
+   echo "#SBATCH --nice=${priority}" >> $LRMS_JOB_SCRIPT
+ fi

+griduser=`whoami`
+case $griduser in
+  gridana)
+    ....
+  *)
+    joboption_rsl_project="grid-prod"
+    ;;
+esac
+
+if [ ! -z "$joboption_disk" ]; then
+  disk_req_in_gb=`echo "$joboption_disk / 1024 + 1" | bc`
+else
+  disk_req_in_gb=10
+fi
+
+echo "#SBATCH --gres=localtmp:${disk_req_in_gb} --constraint=grid&cvmfs" >>
$LRMS_JOB_SCRIPT
+
+ # project name for accounting
+if [ ! -z "$joboption_rsl_project" ]; then
+  echo "#SBATCH -U $joboption_rsl_project" >> $LRMS_JOB_SCRIPT
@@ -188,6 +213,9 @@
+set_req_mem

+if [ ! -z "$joboption_memory" ]; then
+  if [ $joboption_memory -lt 3936 ]; then
+    joboption_memory=3936
+  fi
+  echo "#SBATCH --mem-per-cpu=${joboption_memory}" >>
$LRMS_JOB_SCRIPT
+fi
```



RTE

- Run time environments
 - Advertise available environments
 - Setup environments
 - Modify jobs
- Requested by jobs and used for matchmaking
- Shell script or any executable
- Requested RTEs called at three stages with number as argument
 - 0: Called during job script creation. Allows for modifying job variables before creation.
 - 1: Called by job on the node before running jobs executable. Pre-processing
 - 2: Called as the last part of the job. Post-processing.
- Pro
 - Easy to write and manage
- Con
 - Can only modify known variables before job is written and at specific stages.
 - ARC5: Must be requested by job.



RTE - Example

- UiO revisited
- Create RTE in runtimedir
 - ENV/UIO/CE01
- Missing: modifying constraints
- ARC5: Job must request RTE (workaround exists)
- ARC6:
 - \$ arcctl rte default ENV/UIO/CE01

```
# description: CE01 tailoring of jobs

if [ "x$1" = "x0" ]; then
# add accounting info
griduser=`whoami`
case $griduser in
  gridana)
    joboption_rsl_project="atlas-ana"
    ;;
  *)
    joboption_rsl_project="grid-prod"
    ;;
esac

# add disk requirements
if [ ! -z "$joboption_disk" ]; then
  disk_req_in_gb=`echo "$joboption_disk / 1024 + 1" | bc`
else
  disk_req_in_gb=10
fi

# minimum memory for job 3936
if [ ! -z "$joboption_memory" ]; then
  if [ $joboption_memory -lt 3936 ]; then
    joboption_memory=3936
  fi
fi
fi
```



RTE - Example 2

- ARC Build System
- FEDORA-29-X86_64

```
#!/bin/sh -x

BUILD_CONFIG_DIR=/home/apps/mock/config
REPO_URL=ftp://ftp.example.com/myrepo

BUILD_CONFIG="${BUILD_OSNAME}-${BUILD_OSVERS}-${BUILD_OSARCH}"

case $BUILD_OSNAME in
    debian|ubuntu) BUILD_CMD="$
{BUILD_CONFIG_DIR}/build_command_pbuilder.sh";;
    suse) BUILD_CMD="$
{BUILD_CONFIG_DIR}/build_command_build.sh";;
    *) BUILD_CMD="${BUILD_CONFIG_DIR}/build_command.sh";;
esac

case "$1" in
    0)
        case $BUILD_OSNAME in
            debian|ubuntu) joboption_queue="pbuilder" ;;
            centos) joboption_queue="mock" ;;
            fedora) if test $BUILD_OSVERS = "12"; then
                joboption_queue="mock-el6"
            else
                joboption_queue="mock"
            fi
            ;;
            redhat) joboption_queue="mock" ;;
            suse) joboption_queue="mock" ;;
            *) joboption_queue="mock" ;;
        esac

        joboption_args="${BUILD_CMD} --uniqueext=$joboption_gridid
-r ${BUILD_CONFIG} -s ${REPO_URL}"

esac
```



AUTHPlugin

- Originally a method for authorization/authentication
- Will run an external program just before a-rex transfers job into next state
- Several can be configured
- Can be configured in arc.conf:
 - ARC5: “authplugin = state options plugin”
 - ARC6: “statecallout = state options plugin”
- Plugin: any external executable, must return 0 or the default is to cancel the job
- There are a number of runtime substitutions available
 - e.g. “authplugin= PREPARING timeout=50 /bin/lis %F” will run “/bin/lis /etc/arc.conf”



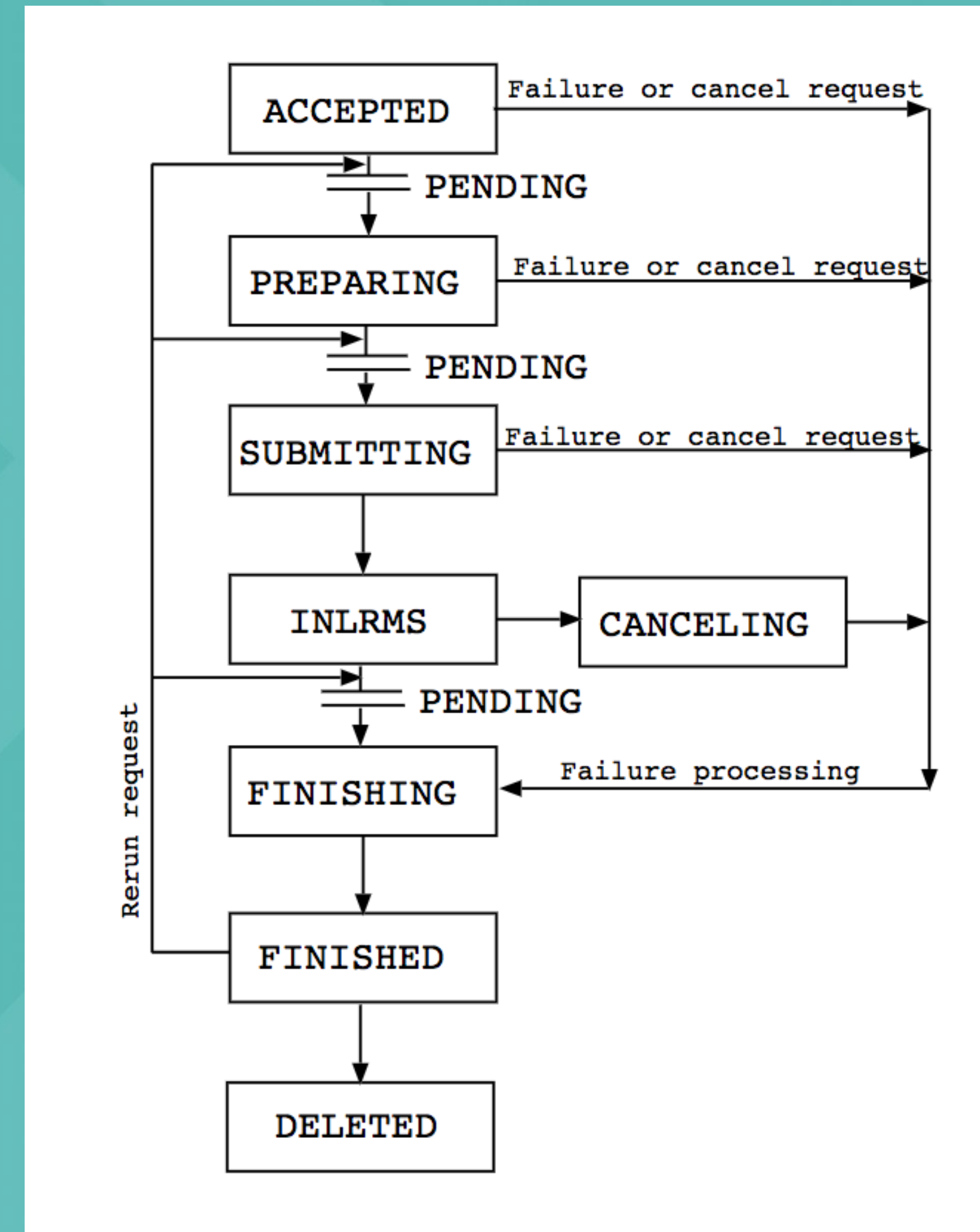
AUTHPlugin - Cont.

- State:

- ACCEPTED - before job is accepted by a-rex
- PREPARING - before downloading input files
- SUBMIT - before writing the job script
- FINISHING - after job has finished, before stage out
- FINISHED - after upload of output files
- DELETED - before all job files are removed from CE

- Options:

- timeout - specifies how long in seconds execution of the plugin allowed to last
- onsuccess, onfailure, ontimeout - action to take
 - Pass: continue execution
 - Log: write to log and continue
 - Fail: cancel job



AUTHPlugin - example

- UiO again. ARC5 default RTE

```
authplugin="PREPARING timeout=60,onfailure=pass,onsuccess=pass  
/usr/local/bin/default_rte_plugin.py %S %C %I ENV/UIO/CE01"
```

- %S - job state
- %C - control directory path
- %I - job ID

- Default_rte_plugin.py can be found here:

https://www.gridpp.ac.uk/wiki/Example_Build_of_an_ARC/Condor_Cluster



AUTHPlugin - default rte

```
#!/usr/bin/python
# Copyright 2014 Science and Technology Facilities Council
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing,
software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
# See the License for the specific language governing permissions
and
# limitations under the License.
```

```
"""Usage: default_rte_plugin.py <status> <control dir> <jobid>
<runtime environment>
"""
```

```
def ExitError(msg,code):
    """Print error message and exit"""
    from sys import exit
    print(msg)
    exit(code)
```

```
def SetDefaultRTE(control_dir, jobid, default_rte):
```

```
    from os.path import isfile
```

```
    desc_file = '%s/job.%s.description' %
(control_dir,jobid)
```

```
    if not isfile(desc_file):
        ExitError("No such description file: %s"%desc_file,1)
```

```
    f = open(desc_file)
    desc = f.read()
    f.close()
```

```
    if default_rte not in desc:
        if '<esadl:ActivityDescription' in desc:
            lines = desc.split('\n')
            with open(desc_file, "w") as myfile:
                for line in lines:
                    myfile.write( line + '\n')
                    if '<Resources>' in line:
                        myfile.write( '    <RuntimeEnvironment>\n')
                        myfile.write( '        <Name>' + default_rte + '</Name>\n')
                        myfile.write( '    </RuntimeEnvironment>\n')
        else:
            if '<jSDL:JobDefinition' not in desc:
                with open(desc_file, "w") as myfile:
                    myfile.write("( runtimeenvironment
= \"\" + default_rte + \"\" )")
```

```
    return 0
```

```
def main():
    """Main"""
```

```
    import sys
```

```
    # Parse arguments
```

```
    if len(sys.argv) == 5:
```

```
        (exe, status, control_dir, jobid, default_rte) = sys.argv
```

```
    else:
```

```
        ExitError("Wrong number of arguments\n"+__doc__,1)
```

```
    if status == "PREPARING":
```

```
        SetDefaultRTE(control_dir, jobid, default_rte)
```

```
        sys.exit(0)
```

```
    sys.exit(1)
```

```
if __name__ == "__main__":
    main()
```



Questions?