



Healthcheck script

“Better to avoid problems than deal with them
when they occur”

History

- Pbs
 - Very centralized
 - Not easily re-configured
- Nagios
 - Very centralized
 - Not easily re-configured

Lots of elaborate scripts to monitor stuff



Now

- HtCondor
 - Not at all centralized – distributed
 - Can be dynamically configured
- Nagios
 - Very centralized
 - Not easily re-configured



HtCondor

- Make a node check itself
- How to do this ...
- Condor CRON

```
STARTD_CRON_JOBLIST=$(STARTD_CRON_JOBLIST)  
WN_HEALTHCHECK
```



Condor CRON config

- STARTD_CRON_JOBLIST=\$(STARTD_CRON_JOBLIST) WN_HEALTHCHECK
- STARTD_CRON_WN_HEALTHCHECK_EXECUTABLE=/usr/local/bin/healthcheck_wn_condor
- STARTD_CRON_WN_HEALTHCHECK_KILL=true
- STARTD_CRON_WN_HEALTHCHECK_MODE=periodic
- STARTD_CRON_WN_HEALTHCHECK_PERIOD=10m
- STARTD_CRON_WN_HEALTHCHECK_RECONFIG=false



Condor CRON

Returns two values.

```
NODE_IS_HEALTHY = True
```

```
NODE_STATUS = "All_OK"
```

```
START=(NODE_IS_HEALTHY =?= True) && (StartJobs =?= True)
```



Condor self monitoring 1

Worked well

Questions about what constitutes a worker node, 'real machines', VMs, containers, ...



Condor self monitoring 2

- Many nagios checks in healthcheck script
 - read-only filesystem
 - ntp check
 - cvmfs checks
 - uptime
 - swap usage

Etc ...



- Note that healthcheck scripts runs as user Condor, so no root checks.
- possibly not a 'good thing', but it works for us.



Extras 1

- Selectively disable Vos

```
START=(NODE_IS_HEALTHY =?= True) && (StartJobs =?= True)
```

```
NODE_IS_HEALTHY = True
```

```
NODE_IS_HEALTHY = True && (regexp("atl", Owner) =?= False)
```



Extras 2

Report to nagios via send_nsca

Query nodes in condor

```
condor_status -constraint 'NODE_STATUS != "All_OK" &&  
partitionableslot == True' -autoformat Machine  
NODE_STATUS
```



Conclusions

- healthcheck_wn_condor works well.
- Uses nagios checks
- Very RAL specific

