

*Adding
High Availability
to Condor Central Manager*
Tutorial

Artyom Sharov
Computer Sciences Department
Technion - Israel Institute of Technology



Outline

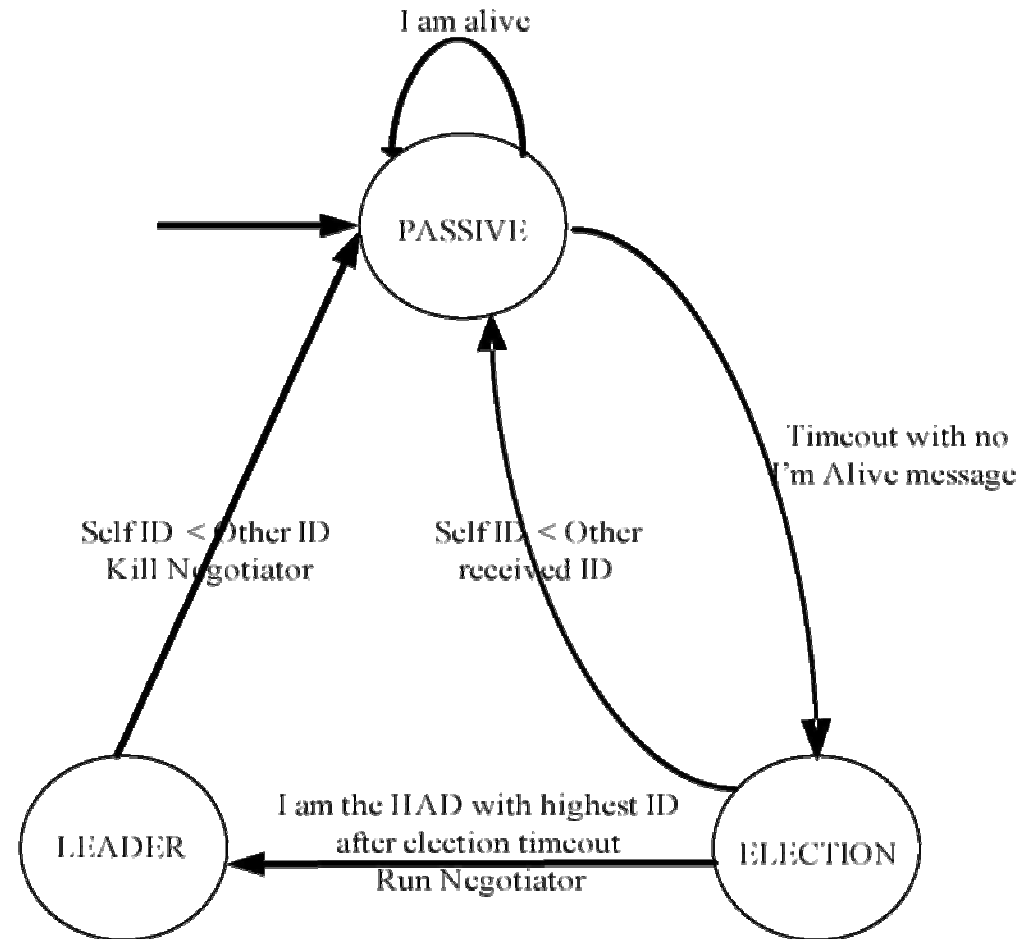
- > Overview of HA design
- > Configuration parameters
- > Sample configuration files
- > Miscellaneous

Overview of HA design

Design highlights (HAD)

- > Modified version of Bully algorithm
 - For more details: H. Garcia-Molina. Elections in a Distributed Computing System., *IEEE Trans. on Computers*, C-31(1):48.59, Jan 1982.
- > One HAD leader + many backups
- > HAD as a state machine
- > "I am alive" messages from leader to backups
 - Detection of leader failure
 - Detection of multiple leaders (split-brain)
- > "I am leader" messages from HAD to replication

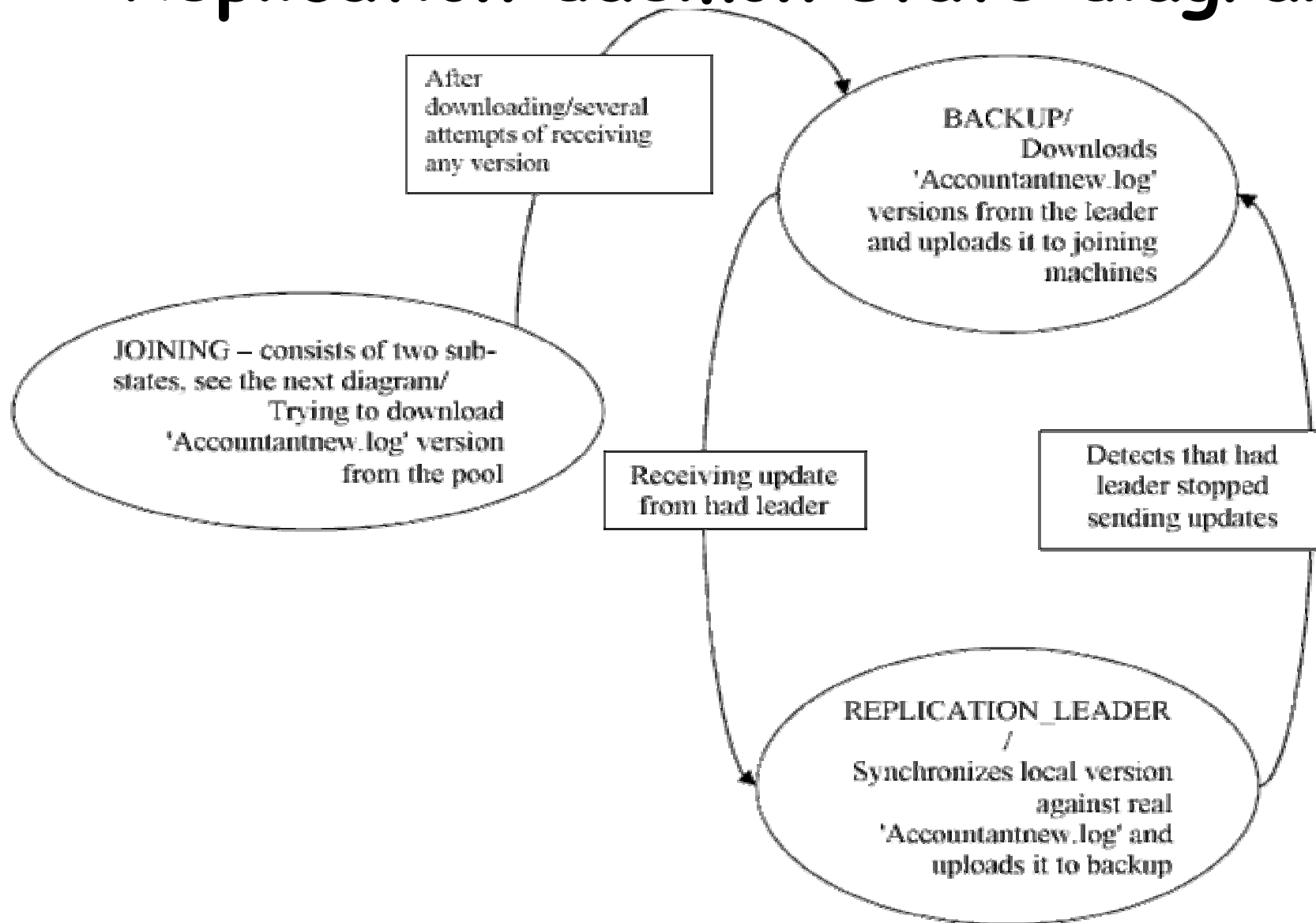
HAD state diagram



Design highlights (replication)

- > Replication daemon **must** have a matching HAD
- > Loose coupling between replication and HAD
- > Separation between a **replication mechanism** and a **consistency policy**
- > Default replication mechanism
 - Transferers
 - File transfer integrity (MAC)
 - Transfer transactionality
- > Default consistency policy
 - Replication daemon as a state machine
 - Version numbers + version file
 - "Split brain" reconciliation support
- > Treating the state file as a black box

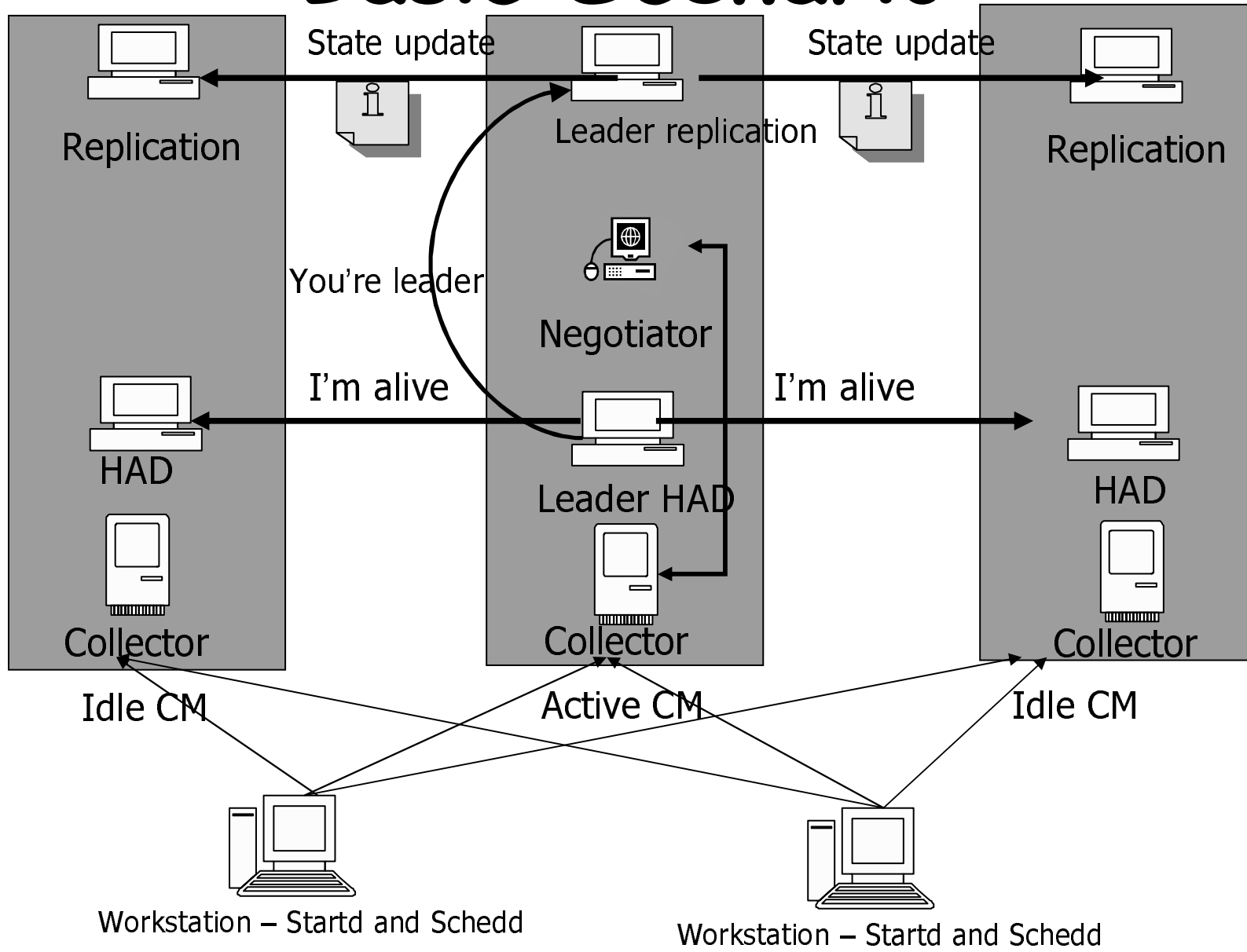
Replication daemon state diagram



HAD-enabled pool

- > Multiple Collectors run simultaneously on each CM machine
- > All submission and execution machines must be configured to report to all CMs
- > High Availability
 - HAD runs on each CM
 - Replication daemon runs on each CM (if enabled)
- > HAD makes sure a single Negotiator runs on one of the CMs
- > Replication daemon makes sure the up-to-date accountant file is available

Basic Scenario



Enablements

- > HA mechanism must be explicitly enabled
- > Replication mechanism is optional and might be disabled

Configuration variables

HAD_LIST

- > List of machines, where the HADs are installed, configured and run
- > Each entry is either IP:port or hostname:port, optionally embraced in <>. The entries are comma-separated
- > Should be identical on all CM machines
- > Should be identical (ports excluded) to the COLLECTOR_HOST list, and in the same order

HAD_USE_PRIMARY

- > One HAD could be declared as primary
- > Primary HAD is always guaranteed to be elected as active CM, as long as it is alive
- > After primary recovers, it will become active CM, substituting one of its backups
- > In case HAD_USE_PRIMARY =true the first element in the HAD_LIST will be the primary HAD. In that case, the rest of the daemons will serve as backups
- > Default is false

HAD_CONNECTION_TIMEOUT

- > An upper bound on the time (in seconds) it takes for HAD to establish a TCP connection
- > Recommended value is 2 seconds
- > Default is 5 seconds
- > Affects stabilization time - the time it takes for HA daemons to detect failure and fix it
- > Stabilization time =
 $12 * \#CMs * HAD_CONNECTION_TIMEOUT$

HAD_USE_REPLICATION

- > Allows administrator of the machine to disable/enable the replication feature on Condor machine configuration level
- > Default is no

REPLICATION_LIST

- > List of machines, where the replication daemons are installed, configured and run
- > Each entry is either IP:port or hostname:port, optionally embraced in <>. The entries are comma-separated
- > Identical on all CM machines
- > In the same order as HAD_LIST

STATE_FILE

- > This file is protected by the replication mechanism. Replicated between all the replication daemons of REPLICATION_LIST
- > Default is
\$(SPOOL)/Accountantnew.log

REPLICATION_INTERVAL

- > Determines how frequently the RD wakes up to do its periodic activities: probing for update of the state file, broadcasting the update to backups, monitoring and managing the downloading/uploading process by transferer processes etc.
- > Since the accounting information file normally changes, as negotiator daemon wakes up, then REPLICATION_INTERVAL value must be like UPDATE_INTERVAL
- > Therefore the default is 300

HAD_ARGS/REPLICATION_ARGS

- > HAD_ARGS = -p <HAD_PORT>
- > REPLICATION_ARGS = -p <REPLICATION_PORT>
- > HAD_PORT/REPLICATION_PORT should be identical to the port defined in HAD_LIST/REPLICATION_LIST for that host
- > Allows master to start HAD/replication on a specified command port
- > No default value. This one is a must

Regular daemon configuration

- > `HAD/REPLICATION` - path to `condor_had/condor_replication` binary
- > `HAD_LOG/REPLICATION_LOG` - path to the respective log file
- > `MAX_HAD_LOG/MAX_REPLICATION_LOG` - maximum size of the respective log file
- > `HAD_DEBUG/REPLICATION_DEBUG` - logging level for `condor_had/condor_replication`

Influenced configuration variables

- > On both client (schedd + startd) and CM machines:
 - COLLECTOR_HOST- list of CM machines
 - HOSTALLOW_NEGOTIATOR - must include all CM machines

Influenced configuration variables

- > Only on Schedd machines:
 - `HOSTALLOW_NEGOTIATOR_SCHEDD` - must include all CMs, because negotiator might theoretically raise on any of CMs
- > Only on CM machines:
 - `HOSTALLOW_ADMINISTRATOR` - CM must have administrative privileges in order to turn Negotiator on and off
 - `DAEMON_LIST` - must include Collector, Negotiator, HAD and (optionally) RD
 - `DC_DAEMON_LIST` - must include Collector, Negotiator, HAD and (optionally) RD

Sample configuration files

Deprecated variables

- > #unset these variables - they are deprecated
- > NEGOTIATOR_HOST=
- > CONDOR_HOST=

condor_config.local. ha_central_manager

- > `CENTRAL_MANAGER1 = cm1.wisc.edu`
- > `CENTRAL_MANAGER2 = cm2.wisc.edu`
- > `COLLECTOR_HOST =`
`$(CENTRAL_MANAGER1),$(CENTRAL_MANAGER2)`

condor_config.local. ha_central_manager (cont.)

- > HAD_PORT = 51450
- > HAD_LIST = \$(CENTRAL_MANAGER1):\$(HAD_PORT),
\$(CENTRAL_MANAGER2):\$(HAD_PORT)
- > HAD_ARGS = -p \$(HAD_PORT)
- > HAD_CONNECTION_TIMEOUT = 2
- > HAD_USE_PRIMARY = true
- > HAD = \$(SBIN)/condor_had
- > MAX_HAD_LOG = 640000
- > HAD_DEBUG = D_FULLDEBUG
- > HAD_LOG = \$(LOG)/HADLog

condor_config.local. ha_central_manager (cont.)

- > HAD_USE_REPLICATION = true
- > REPLICATION_PORT = 41450
- > REPLICATION_LIST =
\$(CENTRAL_MANAGER1):\$(REPLICATION_PORT),
\$(CENTRAL_MANAGER2):\$(REPLICATION_PORT)
- > REPLICATION_ARGS = -p \$(REPLICATION_PORT)
- > REPLICATION = \$(SBIN)/condor_replication
- > MAX_REPLICATION_LOG = 640000
- > REPLICATION_DEBUG = D_FULLDEBUG
- > REPLICATION_LOG = \$(LOG)/HADLog

condor_config.local. ha_central_manager (cont.)

- > DAEMON_LIST = MASTER, COLLECTOR, NEGOTIATOR, HAD, REPLICATION
- > DC_DAEMON_LIST = MASTER, COLLECTOR, NEGOTIATOR, HAD, REPLICATION
- > HOSTALLOW_NEGOTIATOR = \$(COLLECTOR_HOST)
- > HOSTALLOW_ADMINISTRATOR = \$(COLLECTOR_HOST)

condor_config.local. ha_client

- > CENTRAL_MANAGER1 = cm1.wisc.edu
- > CENTRAL_MANAGER2 = cm2.wisc.edu
- > COLLECTOR_HOST =
\$(CENTRAL_MANAGER1),\$(CENTRAL_MANAGER2)
- > HOSTALLOW_NEGOTIATOR =
\$(COLLECTOR_HOST)
- > HOSTALLOW_NEGOTIATOR_SCHEDD =
\$(COLLECTOR_HOST)

Miscellaneous

HAD Monitoring System

- > Analyzes daemons logs
- > Detects failures of the HA mechanism itself
- > Announces about failures to the administrators
- > Runs as a batch job once in some period of time

Disabling HA mechanism

- > Dynamically disabling HA - DisableHAD Perl script
- > Remove HAD, REPLICATION and NEGOTIATOR from DAEMON_LIST on all machines
- > Leave one NEGOTIATOR in DAEMON_LIST on one machine
- > condor_restart CM machines
- > Or turn off running HA mechanism:
 - condor_off -all -negotiator
 - condor_off -all -subsystem replication
 - condor_off -all -subsystem had
 - condor_on -negotiator on one machine

Configuration sanity check script

- > Checks that all HA-related configuration parameters of RUNNING pool are correct
 - HAD_LIST consistent on all CMs
 - HAD_CONNECTION_TIMEOUT consistent on all CMs
 - COLLECTOR_HOST consistent on all machines and corresponds to HAD_LIST
 - DAEMON_LIST contains HAD, COLLECTOR, NEGOTIATOR
 - HAD_ARGS is consistent with HAD_LIST
 - HOSTALLOW_NEGOTIATOR and HOSTALLOW_ADMINISTRATOR are set correct
 - REPLICATION_LIST is consistent with HAD_LIST and REPLICATION_ARGS is consistent with REPLICATION_LIST

Backward Compatibility

- > Non-upgraded client machines will run fine as long as the machine that served as Central Manager before the upgrade is configured as primary CM
- > Non-upgraded client machines will of course not benefit from CM failover

FAQ

- > Reconfigure and restart all your pool nodes, not only CMs
- > Run sanity check script
- > Condor_off -neg will actively shut down the Neg. No HA is provided
- > In case primary CM failed, it takes more time for tools to return results. This is since they query the Collectors in order of COLLECTOR_HOST
- > More than one Neg can be noticed at the beginning for very short time
- > Run monitoring system to track the failures
- > Collector can be queried about the status of HADs in the pool by condor_status utility