Adding High Availability to Condor Central Manager

Artyom Sharov
Technion – Israel Institute of Technology, Haifa
Condor Pool without High Availability
Why Highly Available CM?

- Central Manager is a **single-point-of-failure**
  - No additional matches are possible
  - Condor tools do not work
  - Unfair resource sharing and user priorities

- Our goal - **continuous pool functioning** in case of failure
Highly Available Condor Pool

Startd and Schedd

Highly Available Central Manager

Startd and Schedd

Startd and Schedd

Startd and Schedd

Startd and Schedd

Startd and Schedd

Startd and Schedd

Startd and Schedd
Solution Requirements

- Automatic failure detection
- Transparent failover
- “Split brain” reconciliation
- Persistency of CM state
- No changes to CM code
Condor Pool with HA
HA – Election + Main

Active

Backup 1
Backup 2
Backup 3

#1
I win
Raise Negotiator

#2
I lose
I am alive

Election message
Election message
Election message

I lose

Artyom Sharov, Technion, Haifa
HA – Crash

Active

Backup 1

Backup 2

#3

#4

Failure detection

Election messages

I win

I am alive

I loose

I am alive

Raise Negotiator

Artyom Sharov, Technion, Haifa

Condor week – April 2006
Replication – Main + Joining

#1
State update

#2
Solicit version
Solicit version reply

#3
State update

Artyom Sharov, Technion, Haifa

Condor week – April 2006
Replication – Crash

State update

Failure detection

Active

Backup 1

Backup 2

#4

#5
Configuration

- **Stabilization time**
  - Depends on number of CMs and network performance
  - HAD_CONNECT_TIMEOUT – upper bound on the time to establish TCP connection
  - Example: HAD_CONNECT_TIMEOUT = 2 and 2 CMs - new Negotiator is guaranteed to be up and running after 48 seconds

- **Replication frequency**
  - REPLICATION_INTERVAL
Testing

- **Automatic distributed testing framework:**
  simulation of node crashes, network disconnections, network partition and merges

- **Extensive testing:**
  - distributed testing on 5 machines in the Technion
  - interactive distributed testing in Wisconsin pool
  - automatic testing with NMI framework
HA in Production

- Already **deployed and fully functioning** for more than a year in
  - Technion
  - GLOW, UW
  - California Department of Water Resources, Delta Modeling Section, Sacramento, CA
  - Hartford Life
  - Cycle Computing
  - Additional commercial users
Usability and Administration

- HAD Monitoring System
- Configuration/administration utilities
- Detailed manual section
- Full support by Technion team
Future Work

- HA in WAN
- HAIFA – High Availability Is For Anyone
  - HA for any Condor service (e.g.: HA for schedd)
  - More consistency schemes and HA semantics
  - Dynamic registration of services requiring HA
  - Dynamic addition/removal of replicas
Collaboration with Condor Team

- Ongoing collaboration for 3 years
- Compliance with Condor coding standards
- Peer-reviewed code
- Integration with NMI framework
- Automation of testing
- Open-minded attitude of Condor team to numerous requests and questions
- Unique experience of working with large peer-managed group of talented programmers
Collaboration with Condor Team

This work was a collaborative effort of:

- **Distributed Systems Laboratory in Technion**
  - Prof. Assaf Schuster, Gabi Kliot, Mark Zilberstein, Artyom Sharov

- **Condor team**
  - Prof. Miron Livny, Nick, Todd, Derek, Greg, Anatoly, Peter, Becky, Bill, Tim
You Should Definitely Try It

- Part of the official 6.7.18 development release
- Will soon appear in stable 6.8 release
- More information:
  - more details + configuration in my tutorial
- Contact:
  - {gabik, marks, sharov}@cs.technion.ac.il
  - condor-users@cs.wisc.edu

You Should Definitely Try It
In case of time
Replication – “Split Brain”

Active 1

Merge of networks

Active 2

Decision making:
my ID > ‘Active 2’ ID, I am a leader

Decision making:
my ID < ‘Active 1’ ID, give up

I am alive, Active 2

I am alive, Active 1

HAD

Replication

HAD

Replication

Artyom Sharov, Technion, Haifa

Condor week – April 2006
**Active Backup**

- **Merge of networks**
  - **HAD Replication**
  - **HAD Replication**

You’re leader

‘Active 2’ last version before merge

State update

**Replication**

- **Active**
- **Backup**

Artyom Sharov, Technion, Haifa

Condor week – April 2006
HAD State Diagram
RD State Diagram

JOINING – consists of two sub-states, see the next diagram/ Trying to download "Accountantnew.log" version from the pool

After downloading/several attempts of receiving any version

BACKUP/ Downloads "Accountantnew.log" versions from the leader and uploads it to joining machines

Receiving update from bad leader

Detects that had leader stopped sending updates

REPLICATION LEADER/ Synchronizes local version against real "Accountantnew.log" and uploads it to backup