Adding High Availability to Condor Matchmaker

Mark Silberstein, Gabi Kliot, Assaf Schuster
Technion – Israel Institute of Technology
Highly Available Matchmaker

- Matchmaker is a **single-point-of-failure**
  - Negotiator’s failure - No additional matches will be possible
  - Collector’s failure – negotiator is out of job, tools querying collector won’t work, etc.

- Our goal
  - **Negligible throughput degradation** in case of failure
Highly Available Matchmaker

- Our solution - Highly Available Matchmaker
  - **Automatic** failure detection
  - **Transparent** failover to backup matchmaker (no global configuration change for the pool entities)
  - **State replication** between primary and backups
  - “**Split brain” reconciliation** after network partitions
How it works – Election

Collector

Election msg

Collector

HAD

Collector

Election msg

Collector

Election msg

HAD

Collector

Election msg

HAD
How it works – basic scenario

Collector

Leader HAD

Active Server

Backup Server

Workstation – Startd and Schedd

I’m alive

I’m alive

Negotiator

HAD

Collector

Backup Server

Workstation – Startd and Schedd

HAD

Collector
How it works – crash event

How it works

- crash event

Collector

Leader HAD

Collector

I’m alive

I’m alive

HAD

Collector

Backup Server

Active Server

Backup Server

Workstation – Startd and Schedd

Workstation – Startd and Schedd
How it works – crash event

Election

HAD

Collector

Backup Server

Active Server

Backup Server

Collector

Leader HAD

Collector

Negotiator

Workstation – Startd and Schedd

Workstation – Startd and Schedd
How it works – crash event

LEADER HAD
Collector
Backup Server
Active Server
Backup Server
Collector
Workstation – Startd and Schedd
Workstation – Startd and Schedd

HAD
Collector

Negotiator

I’m Alive

©Gabriel Kliot, Mark Silberstein, Technion

8 Edinburgh Condor week – October 2004
Status

- Prototype was released at April 2004
- Currently finished all implementation and in the advanced stages of debugging
- More information:
  - [http://dsl.cs.technion.ac.il/projects/gozal/project_pages/ha/ha.html](http://dsl.cs.technion.ac.il/projects/gozal/project_pages/ha/ha.html)
In case of time
High Availability Daemon
State machine

- **PRE**
  - Self ID < Other ID
    - Kill Negotiator
  - Self ID > Other received ID

- **PASSIVE**
  - I am alive
  - Timeout with no I'm Alive message
  - Self ID > Other received ID

- **LEADER**
  - I am the HAD with highest ID after election timeout
  - Run Negotiator

- **ELECTION**
  - I am alive
How it works – basic scenario

Startd and Schedd

Collector

HAD

Leader HAD

“Hello”

Collector

Machine A
backup server

Machine B
active server

Machine C
backup server

Workstation
Startd and Schedd
How it works – crash event