Applying Recursive Restartability to Real Systems

George Candea, James Cutler, Armando Fox
&M Mercury dev team
Stanford University

Motivation to Reboot

- Reactive restarts: quickly and effectively recover from trouble
- Prophylactic restarts: run once for 365 days vs. 365 times for one day
- Oracle is perfect
- Failures are detectable by OS
- Nodes in a subtree get restarted together
- Strong fault isolation between groups
- Going up increases MTTR as well as restart confidence
- Concept: tree tree

Core Message

Structuring along MTTF/MTTR boundaries enables
the improvement of system availability w/out
rewriting code or “rewiring” the infrastructure.

- Restart tree = hierarchy that captures
  restart dependencies of system
  (not functional deps, not decision tree)
- Restart group (analogous to UNIX): nodes in a subtree get restarted together
- Strong fault isolation between groups
- Going up increases MTTR as well as restart confidence

Recovery Oriented Computing Retreat
Subtree Depth Augmentation

- fedrcom: high MTTR / low MTTF due to disparate ratio
- Split component along MTTR/MTTF boundaries (we rewired)
- Better MTTR because of fine granularity
- Assumptions:
  - Concurrent restarts
  - Oracle is perfect
  - Restarting does not induce failures elsewhere

Consolidation

- Cascading failures in ise and istr due to synchronization
- Useless overhead every time resulting from doomed restart
- Encode knowledge in restart tree
- Dual of depth augmentation
- Assumptions:
  - Concurrent restarts
  - Oracle is perfect
  - No longer assumed:
    - Independently restartable components

Node Promotion

- Oracle mistakes: guess-too-low and guess-too-high
- Most problematic: widely different MTTRs (fedr: 2 sec, pbcom: 19 sec)
- Push high-MTTR up, low-MTTR down
- Side effect: free fedr rejuvenation
- Assumptions:
  - Concurrent restarts
  - No longer assumed:
    - Independently restartable components

Lessons and Discussion

- MTTF/MTTR-based boundary (re)definition instead of "traditional" ways (e.g., fedrcom → fedr + pbcom)
- Transform restart tree post deployment (addresses most "expensive" time to fail in product's life -- the later you discover a bug, the more expensive it is)
- Not all downtime is the same (e.g., satellite pass)... would you rather have high MTTF or low MTTR?
- Need knowledge of distribution to use MTTF/MTTR in making predictions (typically low coeff of variation assumed)
- Restart group boundaries should not intersect existing failure isolation boundaries

Ongoing Work

- Collect more precise numbers
- Apply RR to Interactive Workspaces Room
- Improve fault detection and logging in ground station
- Design a "RR object" to be inherited by all sw components in system (e.g., a RR EJB in a J2EE-compliant application server)

More...

http://RR.stanford.edu