Management Overlay Networks (MON)

Jin Liang, Indranil Gupta, and Klara Nahrstedt

Motivation

Managing Large Dist. App.

Challenges: support the queries dealing with
- Scale: 1000s or even 10000s of nodes
- Failures: various node/network failures...

Existing vs. Our Approach

Centralized | | Persistent

- Centralized Monitor/Control (CoMon, GIS, PSSH...)
  - Not scalable!
  - No in-network agg.!

- Persistent Overlay Structure
  (Astrolabe, SDIMS, ...)
  - Failure repair complex
  - Latent failure unavoidable

Distributed & On-demand

- Distributed Query Execution
  Overlays (trees, DAGs) for
  - Command propagation
  - Result aggregation

- On-demand Overlay Creation
  Create, use and discard
  - Light-weight and simple
  - More resilient to failures

Architecture

Distributed App Management

- Status query and control
- Agg query (avg, top-k,...)
- Software push

On-Demand Construction

- Tree and DAG overlays
  - Locality awareness
  - Probabilistic coverage

Membership Management

- Light-weight gossip
- Failure detection
- Proximity measurement

Overlay Construction

Phase 1: membership overlay
- Random graph (gossip)
- Loosely structured overlay

Membership list at one node

Phase 2: on-demand creation
- Based on membership info
- Controlled overlay shape

Commands
Data

MON Performance

- Membership dissemination for 1024 nodes (simulation)
  - Generalization of failed nodes
  - Membership expiration time 1.35 sec

- Tree construction on 330 nodes (PlanetLab)
  - Coverage: 321.59 nodes
  - Creation: 2.79 sec
  - Count time: 1.35 sec

Reliability of On-demand Overlay

Observations:
1. On-demand overlay can have good session reliability, lasting 10s of minutes w/o repair, esp. with redundant links (DAG).
2. Command re-try can improve task reliability, at the expense of execution time.
3. Essentially, command reliability depends on dealing with forward path and return path failures

http://cairo.cs.uiuc.edu/mon/

www.iti.uiuc.edu
University of Illinois at Urbana-Champaign