GENERALIZING REQUEST-FLOW COMPARISON TO MORE SYSTEMS

Raja Sambasivan, William Wang, Greg Ganger (CMU)

OVERVIEW

- Request-flow comparison has been useful in helping diagnose real problems in Ursa Minor and select Google services
- **Goal:** Show it is a general approach useful in many systems
- Extensions needed to handle large graphs, contention, etc.
- Generality will be tested by diagnosing real HDFS problems

HANDLING VERY LARGE GRAPHS

Limitation:

- Spectroscope works best when graphs are small
  - Heuristics become expensive w/large graphs
  - Developers cannot understand very large graphs
- Some systems generate very large graphs
  - E.g., Graphs of HDFS writes contain over 15K nodes

Initial solution: Iterative zooming

- Limits graph sizes by "drilling down" on interesting changes
- Workflow:
  - Initially, very low granularity graphs are compared
  - Next, more trace points are exposed in changed areas and just those areas are re-compared

DIAGNOSING RESOURCE CONTENTION

Limitation:

- Contention is a significant source of performance changes in shared-macine environments
- Spectroscope cannot identify interfering interactions

Initial solution: Concurrent request analysis

- Technique Identifies whether a mutation is caused by interference w/concurrently executing interactions
- Workflow:
  - Identify if items responsible for a mutation have high latencies only when specific other items are executing
  - Report items if this conditional probability is high

HANDLING CATEGORY EXPLOSION

Limitation:

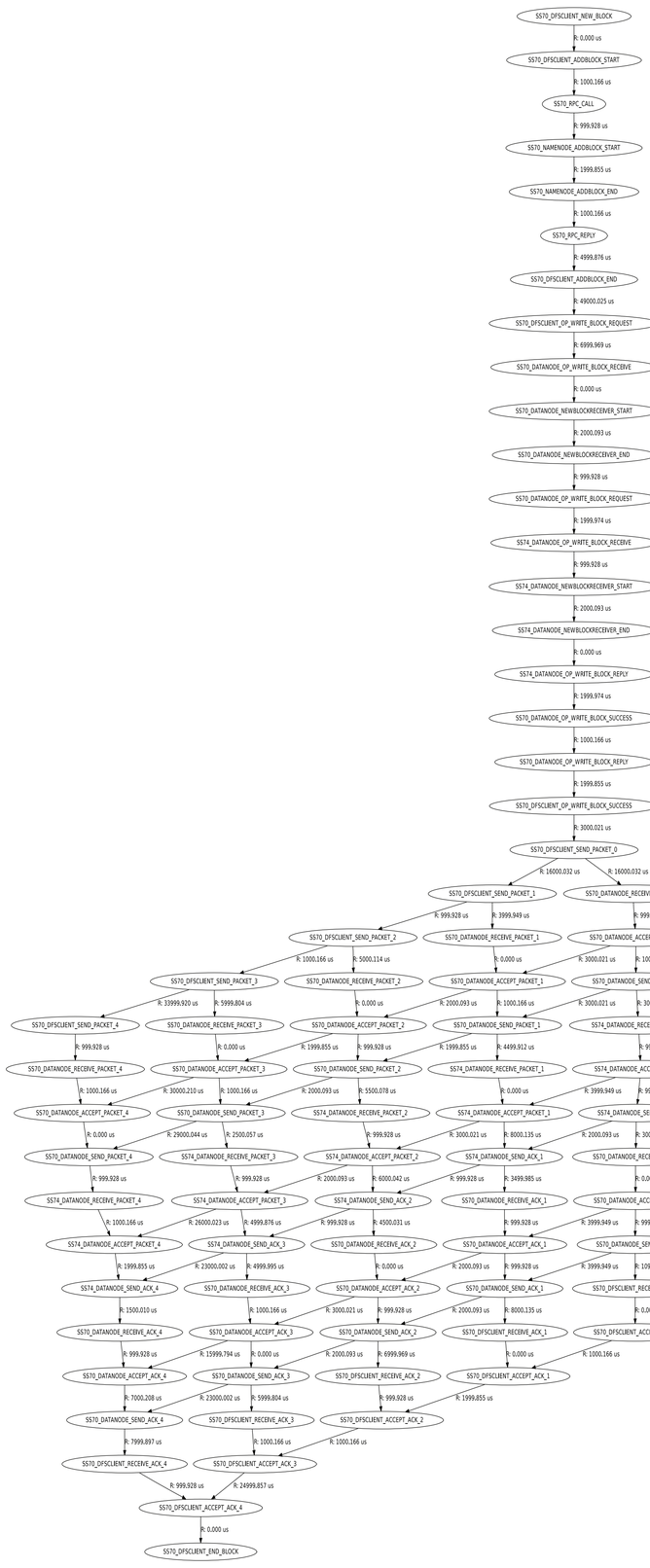
- Spectroscope creates categories of identically-shaped reqs
  - Mutations ID'd by comparing per-category distributions
- In some systems, too many categories will be created
  - In HDFS, up to $_{N}P_{R}$ categories can be generated by writes
    - N = # of datanodes, R = replication factor

Initial solution: machine clustering

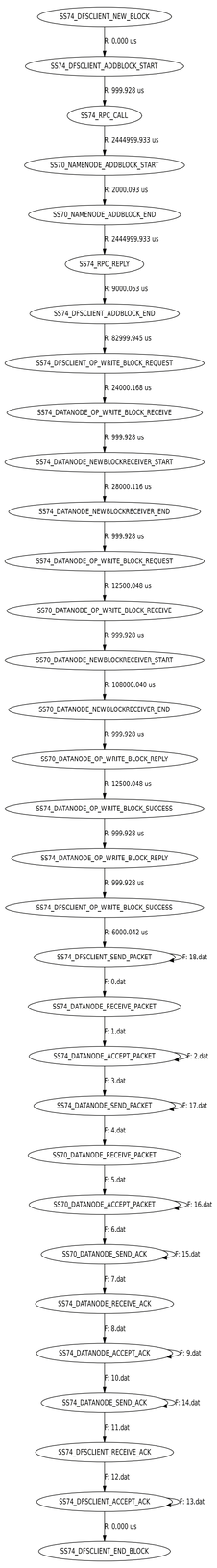
- Technique uses clustering to reduce number of machines
- Workflow:
  - Cluster individual machines based on perf. similarity
  - Use cluster ID instead of machine ID when creating graphs

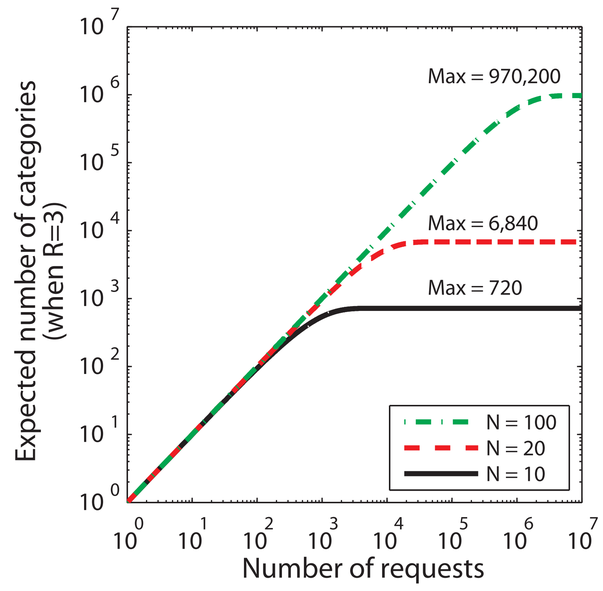
CREATING LOW-GRANULARITY GRAPHS

HDFS 320KB Write graph



Low-granularity version





Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University · Georgia Tech · intel · PRINCETON UNIVERSITY · UC Berkeley