

Using Proxies to Accelerate Cloud Applications

Siddharth Ramakrishnan
Jon Weissman

Department of CSE
University of Minnesota

Introduction

- Cloud ecosystem (Gannon 2009)
 - SAAS: (Google Spreadsheet, Gmail)
 - I/P-AAS: (Virt: EC2/S3, Azure), Google AppEngine
 - Parallel frameworks: (MapReduce cloud)
- Scale-up/Scale-down
- Remote execution/hosting
- Performance
- Transparency

Application View: Cloud Diversity

- Data clouds
 - S3, SkySurvey, GoogleHealth, ...
- Compute clouds
 - EC2, IronScale, ...
- Service clouds
 - Gmail, Gmaps, Google-earth

Trends

- Specialization and diversity
 - Functional and non-functional
 - Non-functional: security, reliability, SLAs, cost
 - Functional: type of data, type of services, ...
- Distributed clouds
 - Smaller footprint data center containers geographically dispersed
 - Logical cloud federation: OpenCirrus

Confluence

- Diversity of clouds + push for distribution
- (1) No single cloud model will rule
- (2) New distributed models are attractive
- (3) Emerging applications will utilize multiple clouds “multi-cloud” applications

An Aside: Edge Systems

- Edge systems
 - Compute-oriented: BOINC, @home, ...
 - Data-oriented: P2P, Bittorent, openDHT, ...

Appeal: scale, cost, *diversity*

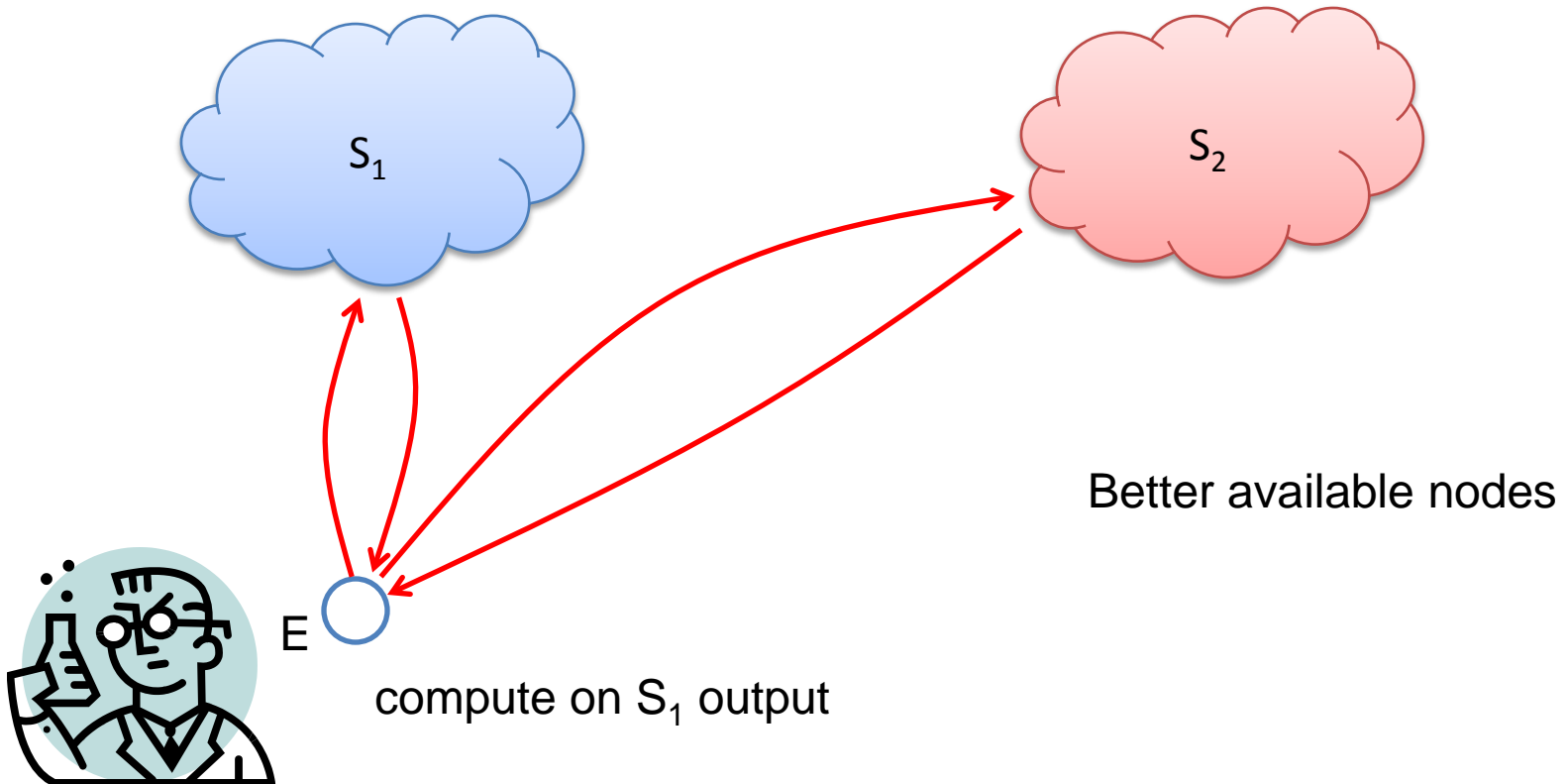
=> Edge computers can play an important role in multi-cloud applications

Multi-Cloud Applications

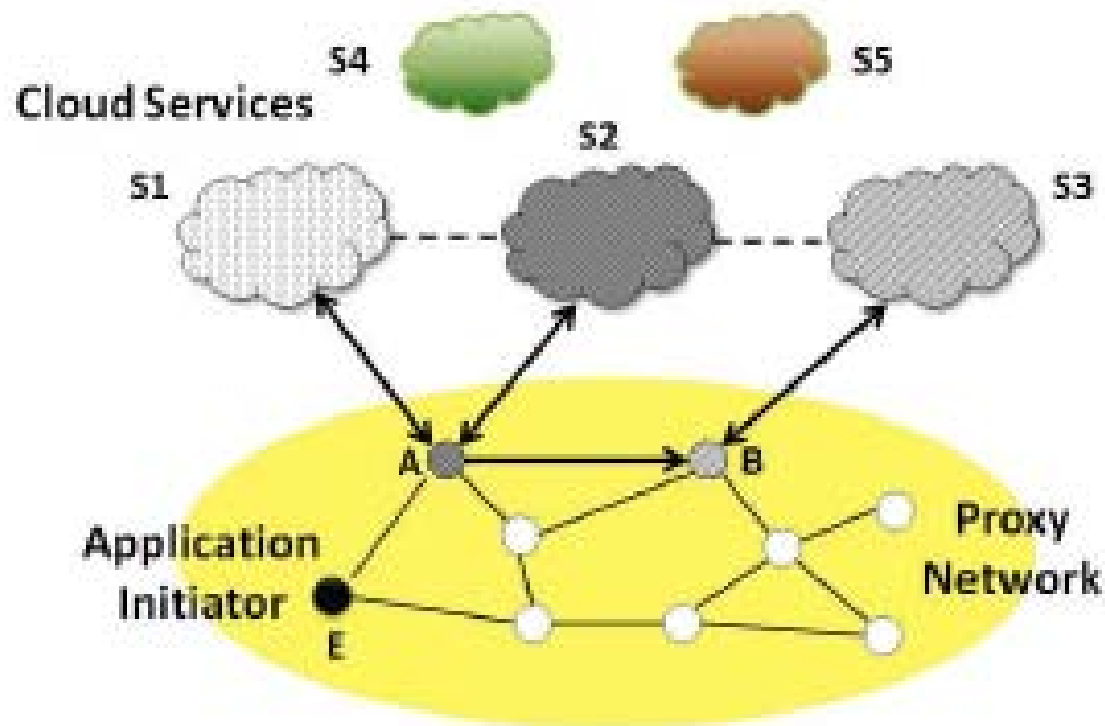
- Specialization => data-intensive applications will increasingly span multiple clouds
 - data is dispersed across multiple clouds
- Distributed data mining
 - Ex: weather data + commodity prices
- Scientific workflows
 - Ex: life science: GenBank<->BLAST<->PubMed, ...
- Mashups
 - Ex: GoogleEarth + CDC pandemic data
- Multi-cloud parallel frameworks
 - Ex: MapReduce, AllPairs, ...

The Problem

- Current cloud interaction paradigm is client-server
 - **Web Services or http**
- Data flows back and forth to end-client application

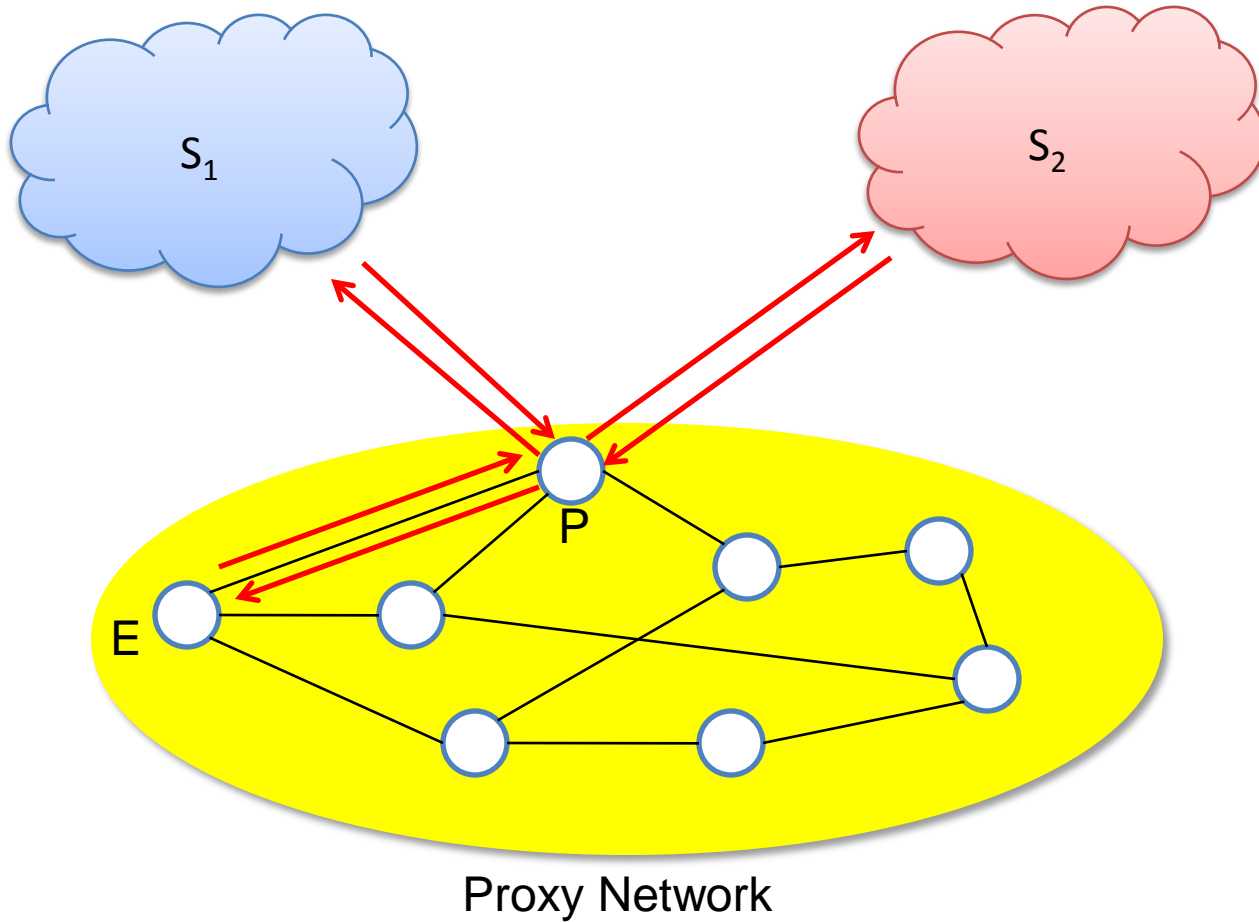


Solution: Proxy Architecture: 50K ft



Resource constrained

Exploit diversity
of proxy nodes



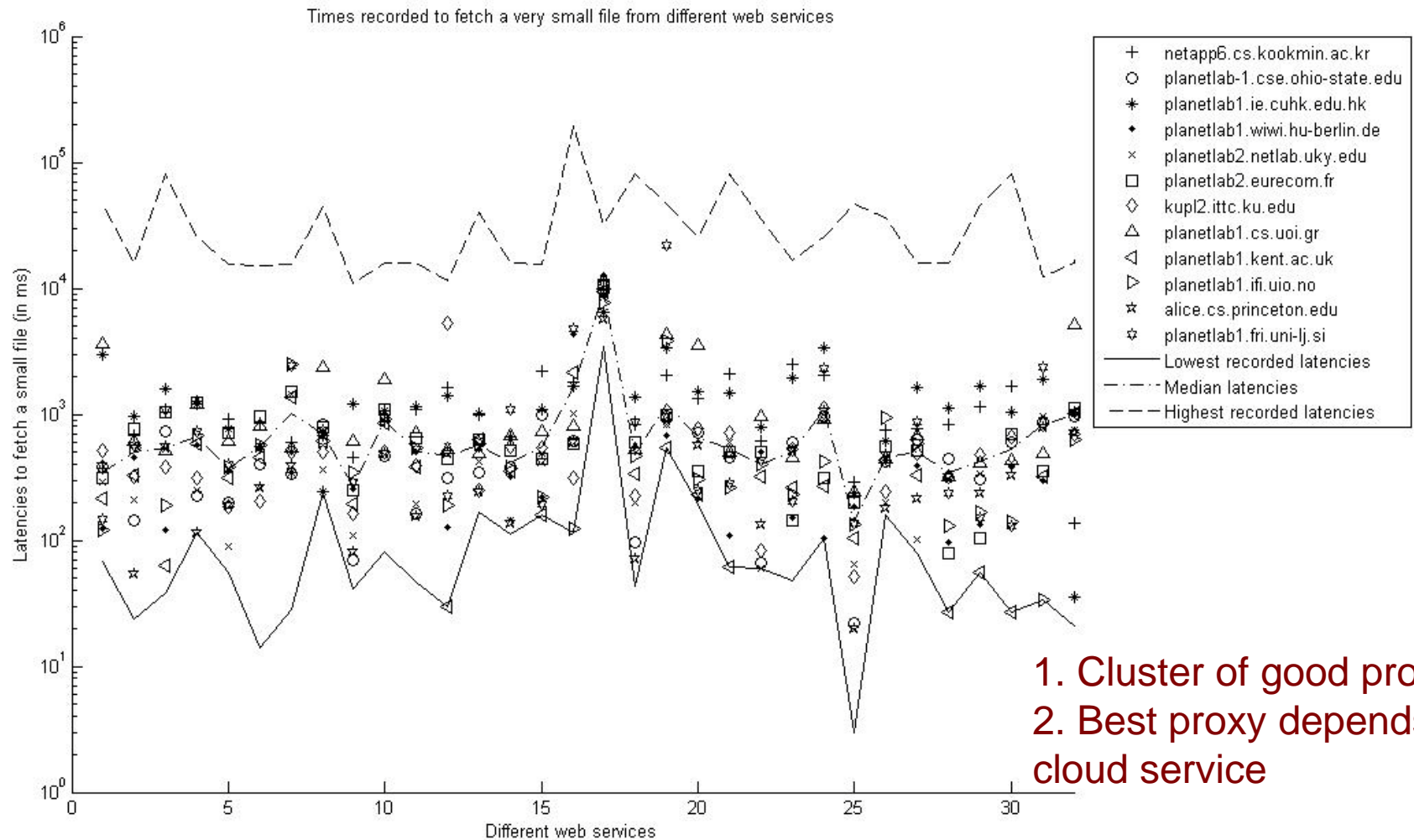
Data-oriented Proxy Roles

- Cloud service interaction
 - Proxy as a client
- Routing
 - Proxy routes data to other proxies
- Computing => Grids
 - Proxy computes data operators: compress, filter, merge, mine, ...
- Caching => P2P
 - Proxy caches data (from cloud, computations, ...)

Proxy Network

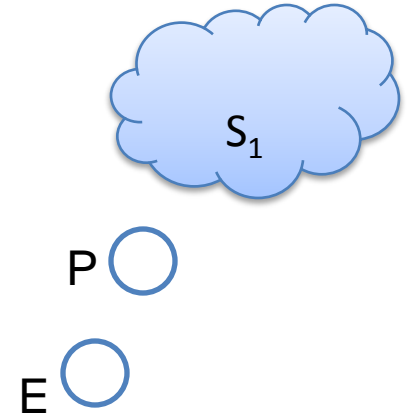
- Where do proxies come from?
 - volunteers, deployed CDNs, ...
- How do proxies form overlays?
 - is there a system-wide overlay and/or application-specific overlays?
 - need more experience with multi-cloud applications

How Much Network Diversity?

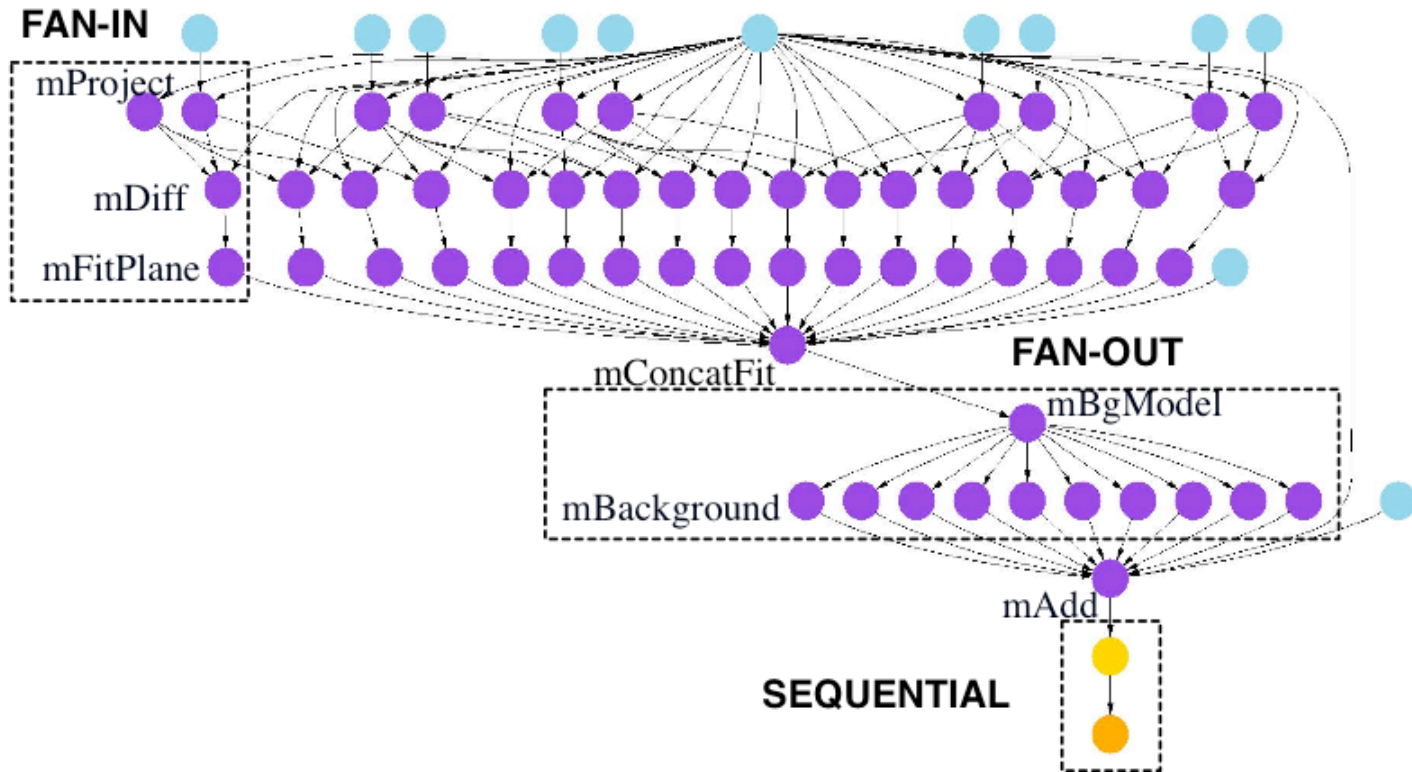


Proxy Hop Penalty?

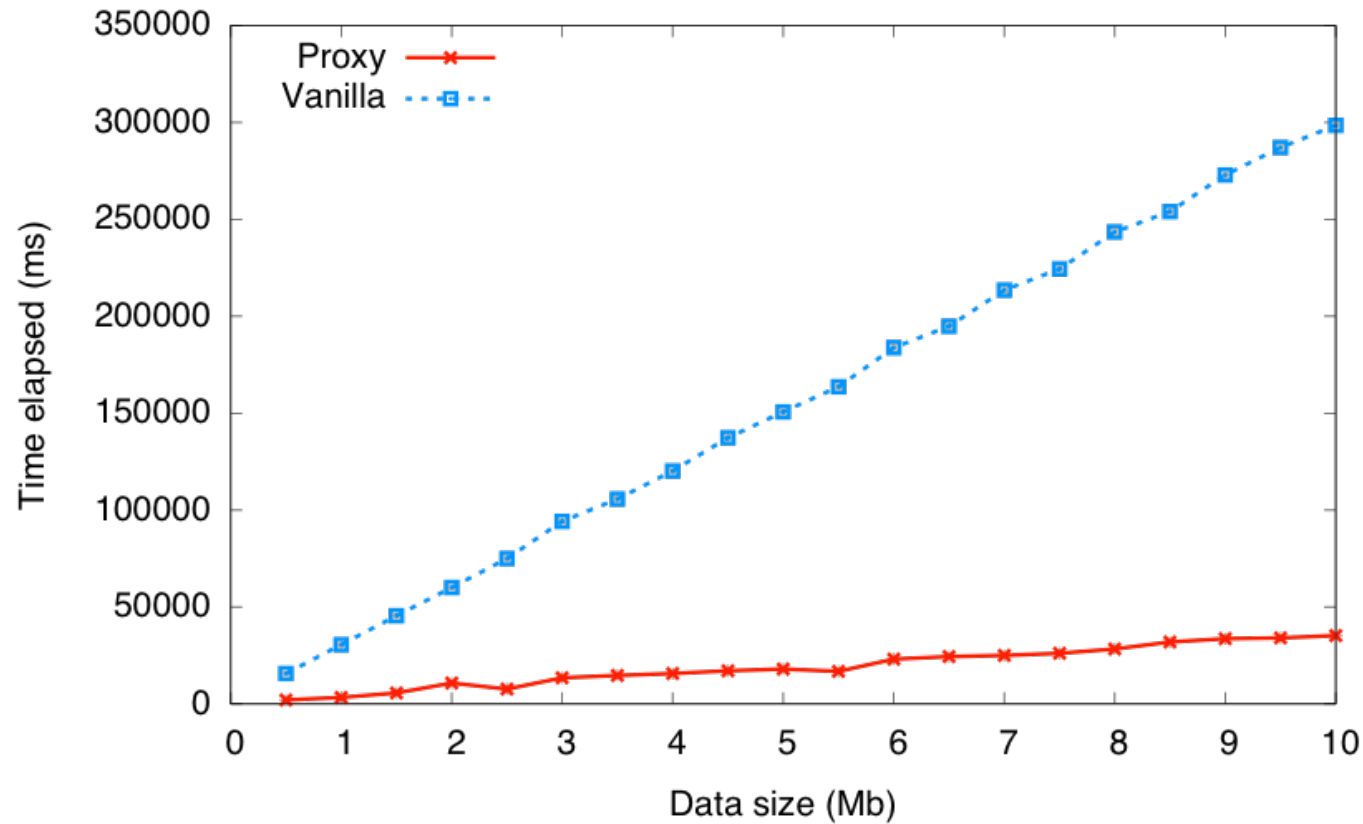
- Despite network proximity and data reduction, proxies may add a network hop
 - 1600 paths
 - Over 70% benefited by intermediary
 - Over 20% performance improvement



Example: Montage



Montage Speedup

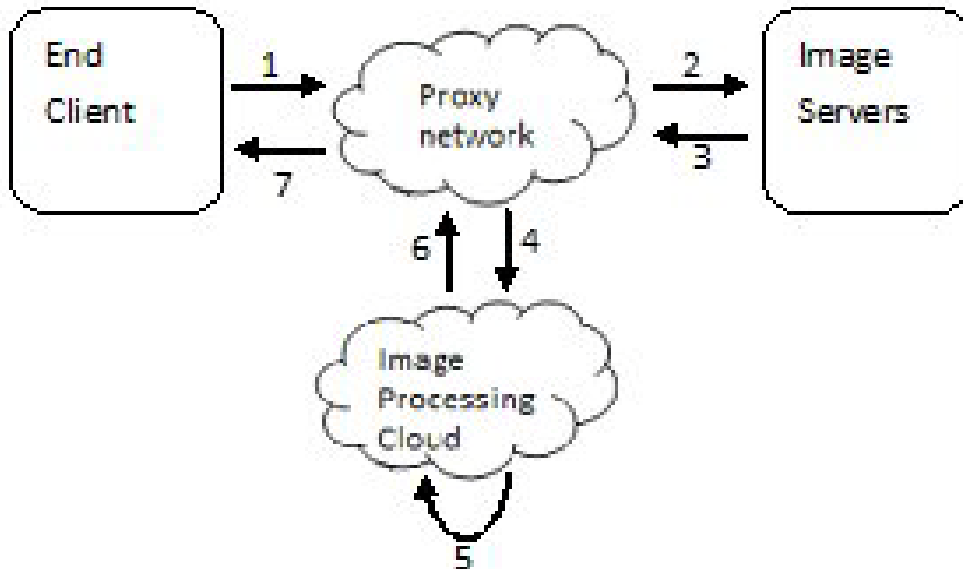


Initiator is the workflow engine, remote from Montage services
One proxy per Montage service, co-located

Example: Image Processing

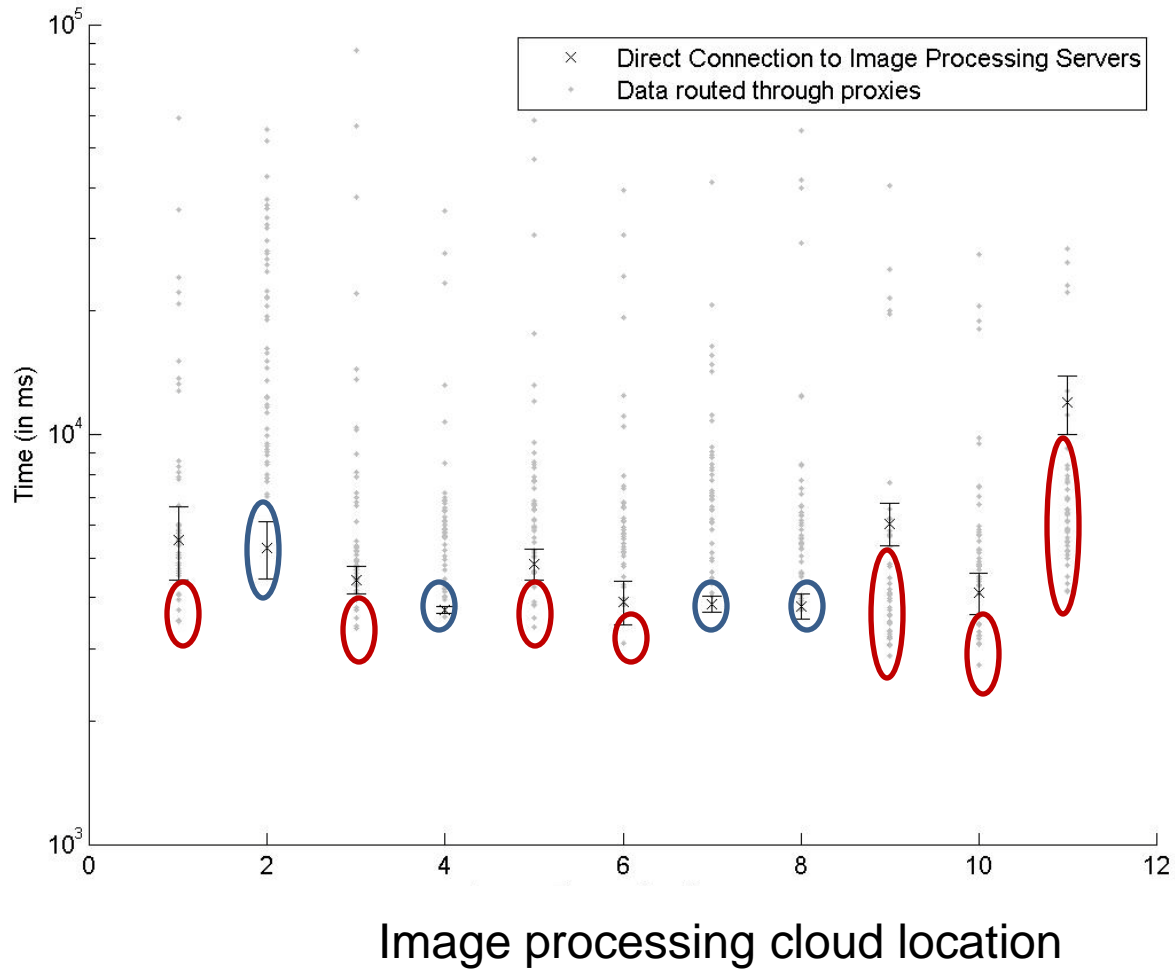


Basic workflow



Enhanced proxy workflow

Results



end-user
image server
location fixed

There exist many
proxies
that can
accelerate this
application

Summary

- Cloud specialization will trigger a new wave of multi-cloud applications
- Proposed a proxy network to “accelerate” these applications => bottleneck awareness
- Many research challenges
 - Proxy node selection
 - Proxy network configuration