

A Cloud Computing Approach to On-Demand and Scalable CyberGIS Analytics

Pierre Riteau (UChicago), Myunghwa Hwang (UIUC),
Anand Padmanabhan (UIUC), Yizhao Gao (UIUC),
Yan Liu (UIUC), Kate Keahey (Argonne),
Shaowen Wang (UIUC)

Presented by Kate Keahey

6/23/14

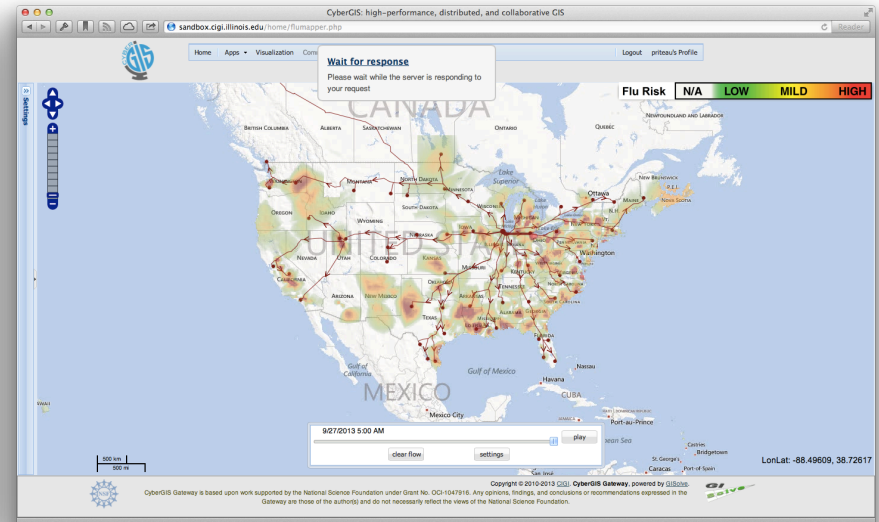
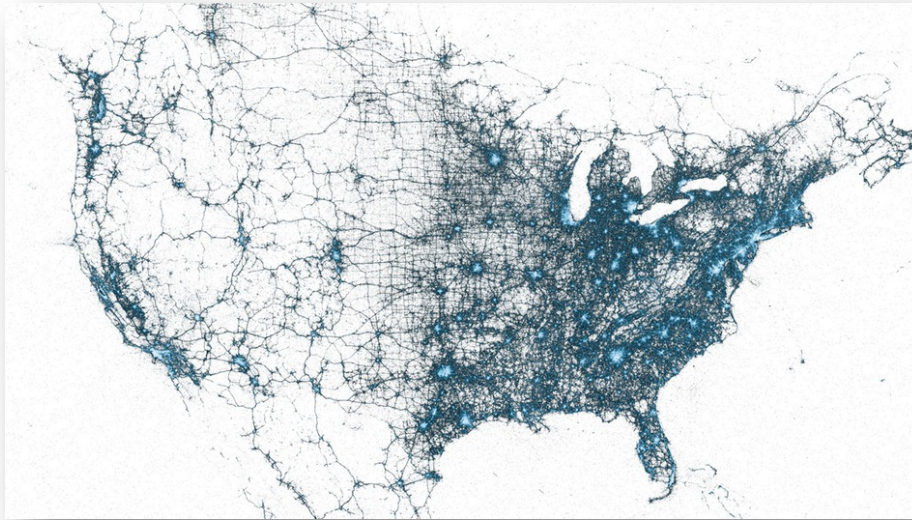


www.nimbusproject.org
www.cybergis.org



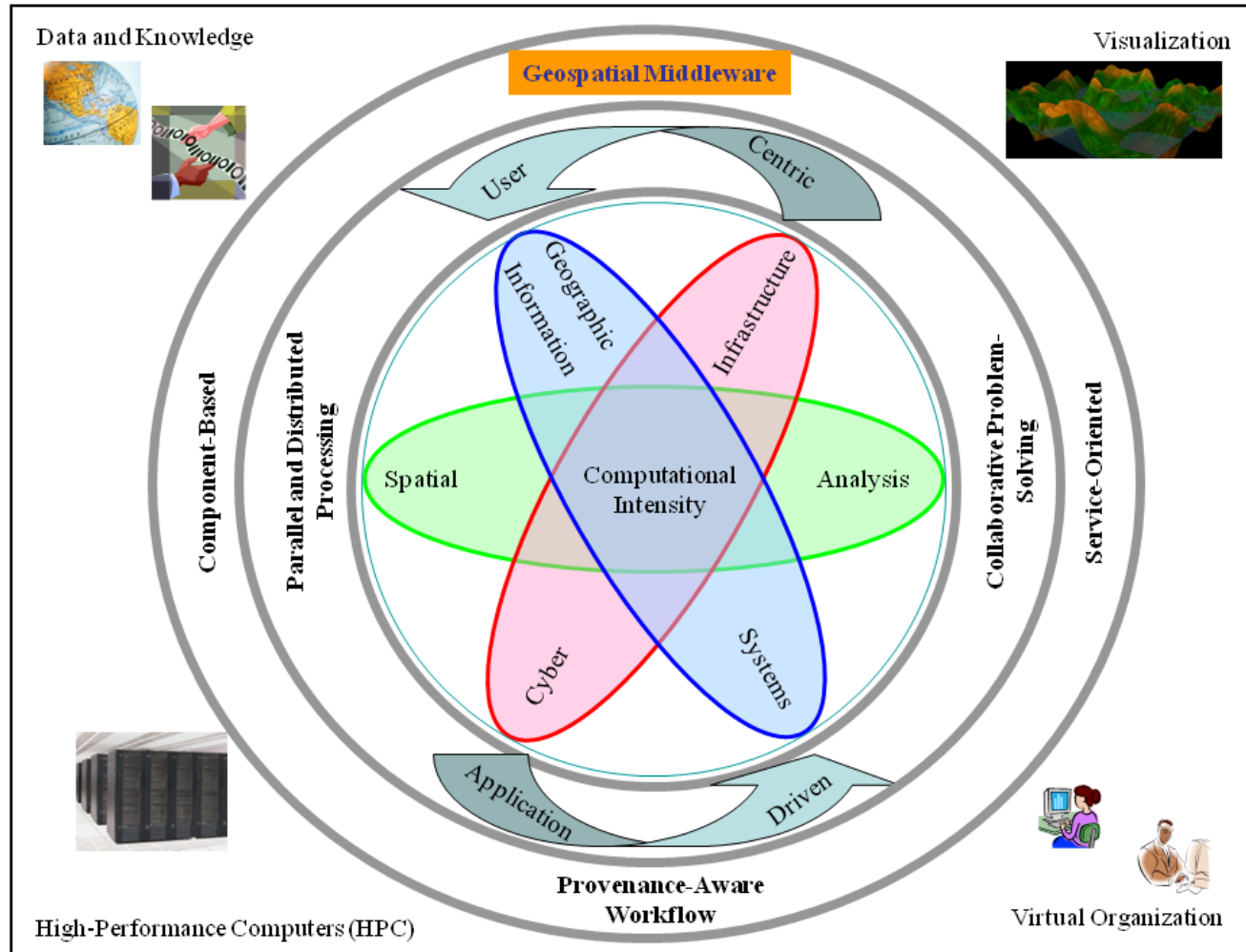
1

Geographic Information Systems (GIS)



- *“Geographic Information Systems (GIS) are simultaneously the telescope, the microscope, the computer, and the Xerox machine of regional analysis and synthesis of spatial data.” (Abler 1988)*

CyberGIS



Wang, S. 2010. "A CyberGIS Framework for the Synthesis of Cyberinfrastructure, GIS, and Spatial Analysis." *Annals of the Association of American Geographers*, 100(3): 535-557



www.nimbusproject.org
www.cybergis.org



The Problem

- Consistent response time in peak demand
 - Example: online education
 - Demand from many users varies over time and across tasks
 - Response time has critical impact on user experience
- Adaptation to varying sizes of analytical problems
 - Example: Problem Solving Environments
 - Real-time interaction, requests with potentially large spatial data

Using Cloud Resources

Potential

- On-demand provisioning of resources
- Pay-as-you-go cost model

Challenges

- Deploying spatial analytics modules on cloud resources
- Integrating cloud resources with existing CyberGIS infrastructure and middleware
- Balancing computational workload across resources
- Scaling resources dynamically so that acceptable quality of service can be maintained

PySAL on CyberGIS Gateway

The screenshot shows the 'Model Specification & Estimation' interface of the CyberGIS Gateway. The interface is divided into several sections:

- Navigation:** Home, Apps, Visualization, Community, Support, About, Feedback, Logout, shaowen's Profile.
- Model Specification & Estimation:**
 - Specification:** Drag and drop variables from the Variables panel. Double-click each variable to remove it. Once the model is specified, click Run on toolbar or Next at bottom.
 - Y (Required):** PRICE, YE
 - X (Required, at least one variable):** NBATH, DWELL, NROOM
 - Estimation:**
 - Model Type:** Standard, Spatial Lag, Spatial Error, Spatial Lag+Error (selected)
 - Method:** OLS, GMM (selected), ML
 - Standard Errors:** White, HAC, KP HET (checked)
- Variables:** Search: Filter variables. List of variables: STATION, PRICE, NROOM, DWELL, NBATH, PATIO, FIREPL, AC, BMENT, NSTOR, GAR, AGE, CITCOU, LOTSZ, SQFT, X, Y.



CyberGIS Gateway is based upon work supported by the National Science Foundation under Grant No. OCI-1047916. Any opinions, findings, and conclusions or recommendations expressed in the Gateway are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

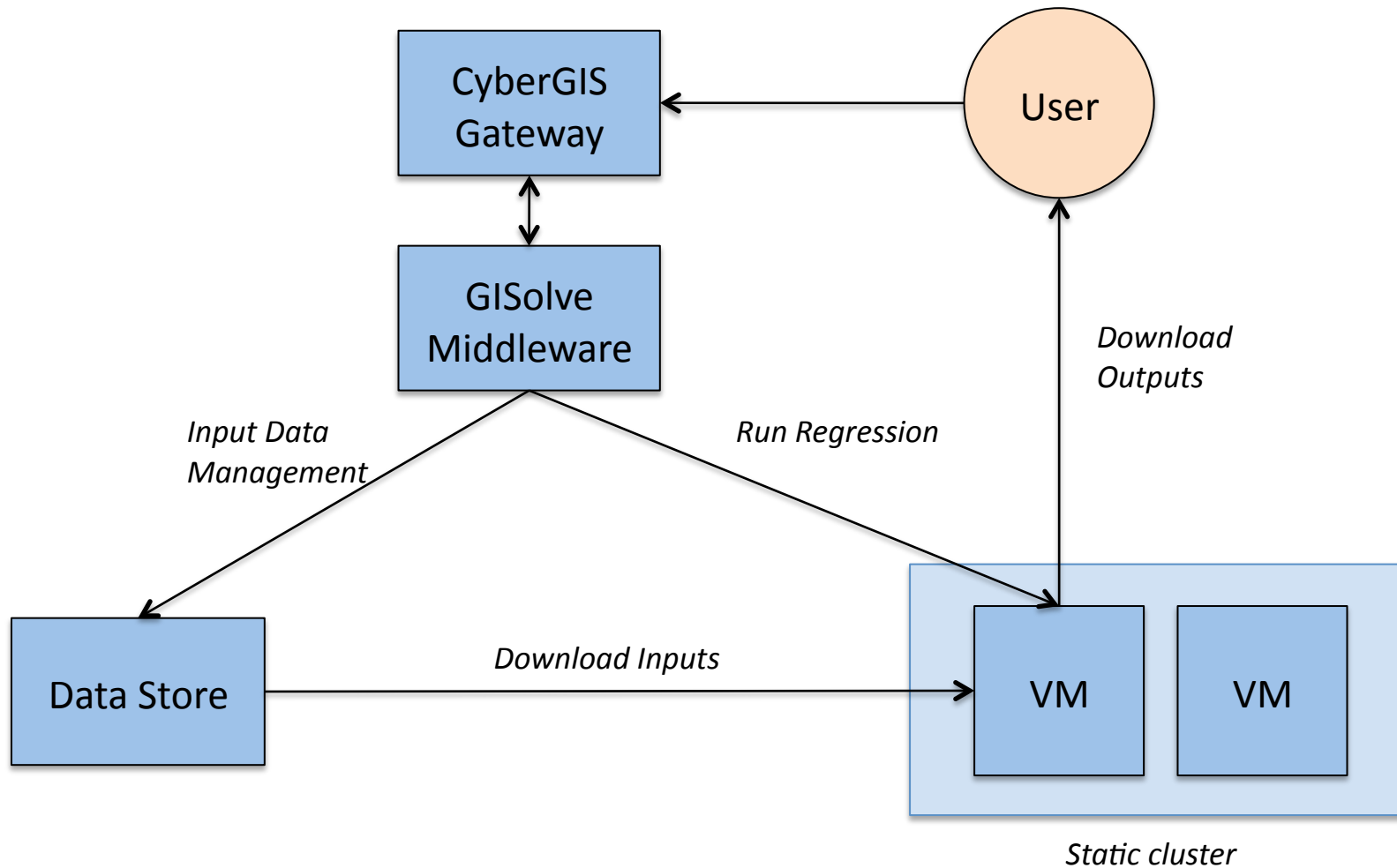
Copyright © 2010-2013 CIGI. CyberGIS Gateway, powered by GISolve, GeoDa, PySAL.



www.nimbusproject.org
www.cybergis.org



CyberGIS: Current Architecture



CyberGIS: Original Architecture (cntd)

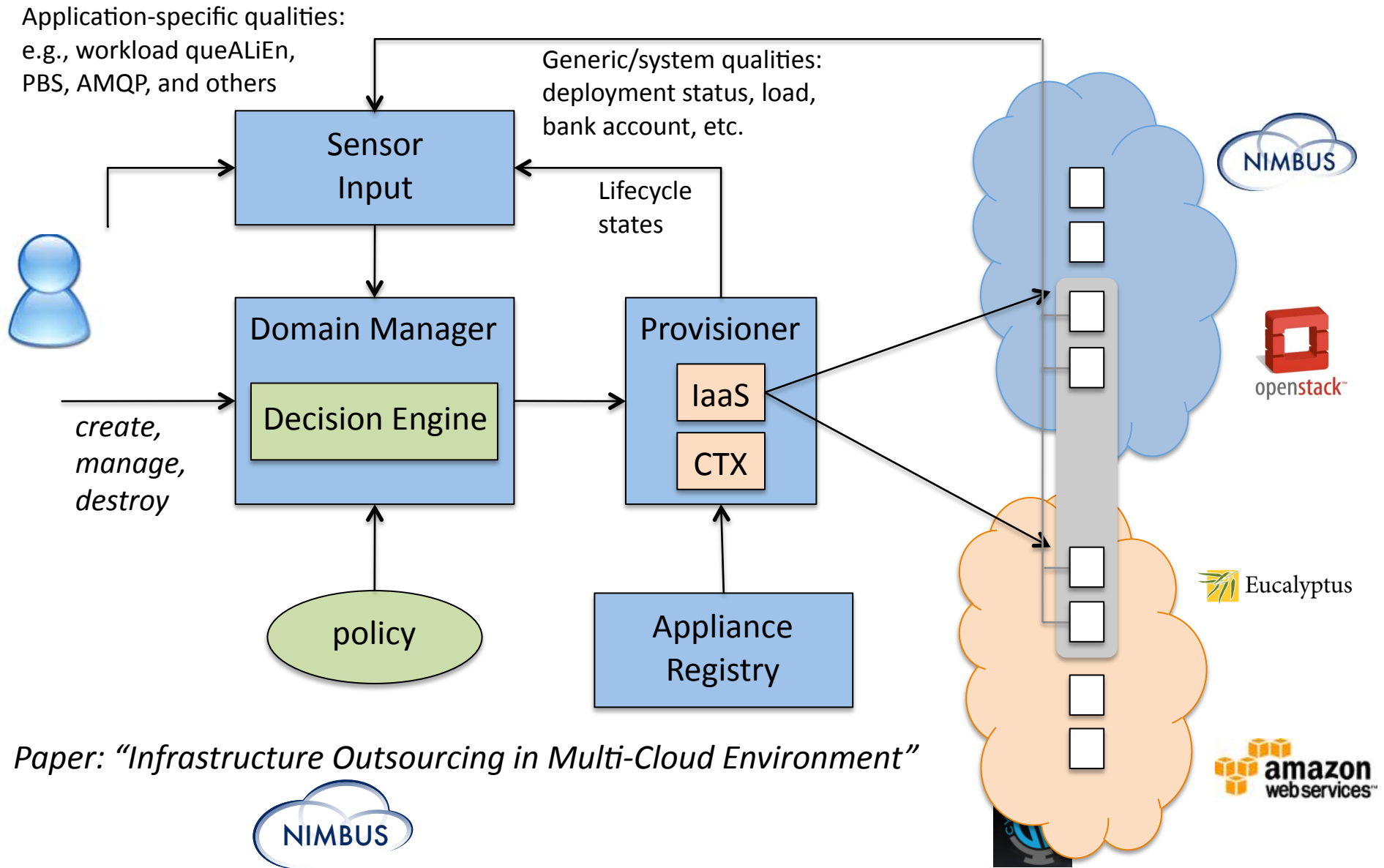
- Users submit jobs through the Gateway
- Input data uploaded to the Data Store
- GISolve middleware distributes requests in round robin to a static cluster of VMs with PySAL installed
- No queuing: extra requests rejected
- Output downloaded directly from VM
 - Assumes static deployment

Moving CyberGIS to a Cloud Platform

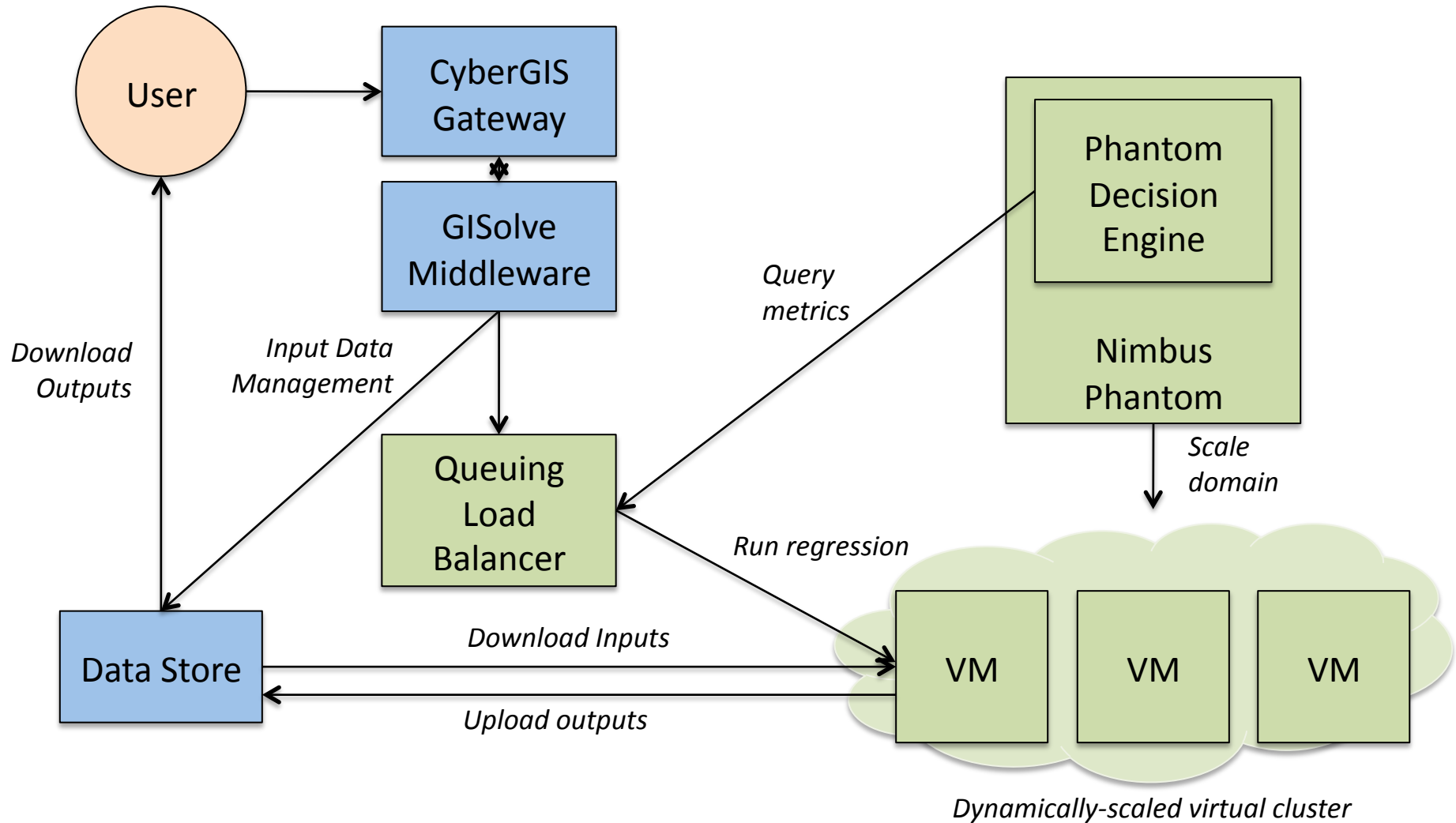
- Need to add/remove instances on the fly
- Our solution:
 - Add queuing load balancer behind GISolve
 - No need to modify GISolve middleware code
 - Use Nimbus Phantom and the load balancer information to implement auto-scaling



Nimbus Phantom



CyberGIS: Modified Architecture



Implementation

- HAProxy as load balancer
 - Metrics extracted using haproxyctl
- Custom Phantom decision engine
 - Tracks the number of connections to HAProxy
 - Requests changes in number of instances
- Policy
 - Requests new instances when VMs fill to capacity
 - “Lazy termination” based on history to avoid thrashing
- Instances are integrated in HAProxy when booted and removed when terminated
- Output files stored on data store
 - Instances can be terminated any time

Experimental Platform

- Used OpenStack Alamo on FutureGrid
- Dedicated instances for:
 - HAProxy (m1.tiny)
 - Data Store (m1.small)
 - Regression service (m1.small)
- Comparison of:
 - Static cluster (original architecture)
 - Static cluster + dynamically added instances



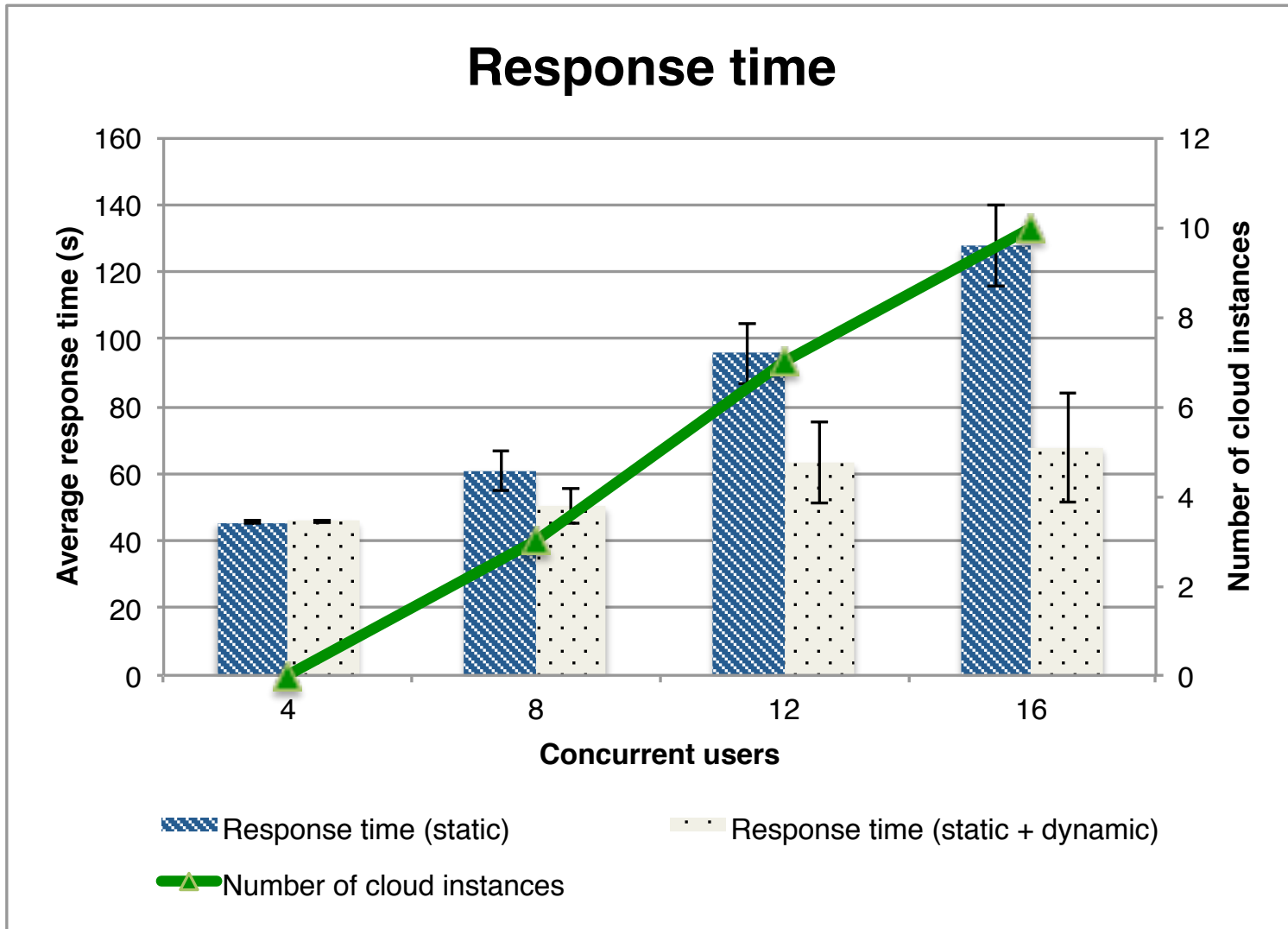
Experiments

- Two use cases scenarios
- Scenario 1
 - Small number of users
 - Large data files
 - Example: scientists conducting a study
- Scenario 2
 - Large number of users
 - Smaller data files
 - Example: labs conducted as part of a class
- Generated load with Apache JMeter

Scenario 1 (Large Requests)

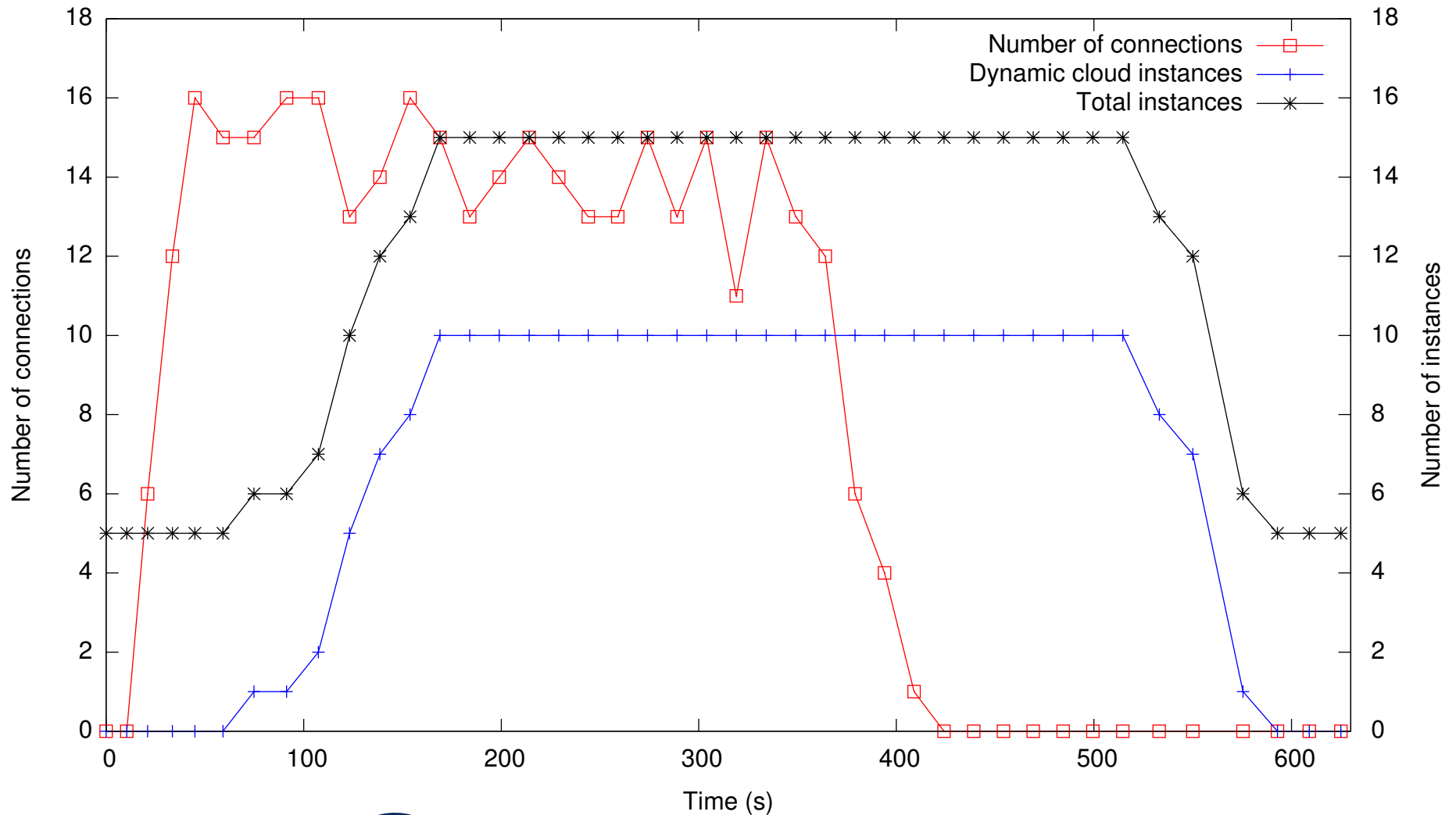
- Number of users varies from 4 to 16
- 5 requests per user
- 10 second pause between requests
- Static cluster of 5 VMs
- Maximum of 10 dynamic cloud instances
- 2 minutes auto-scaling history buffer
- Single request per VM (no concurrency)

Scenario 1 (cntd)



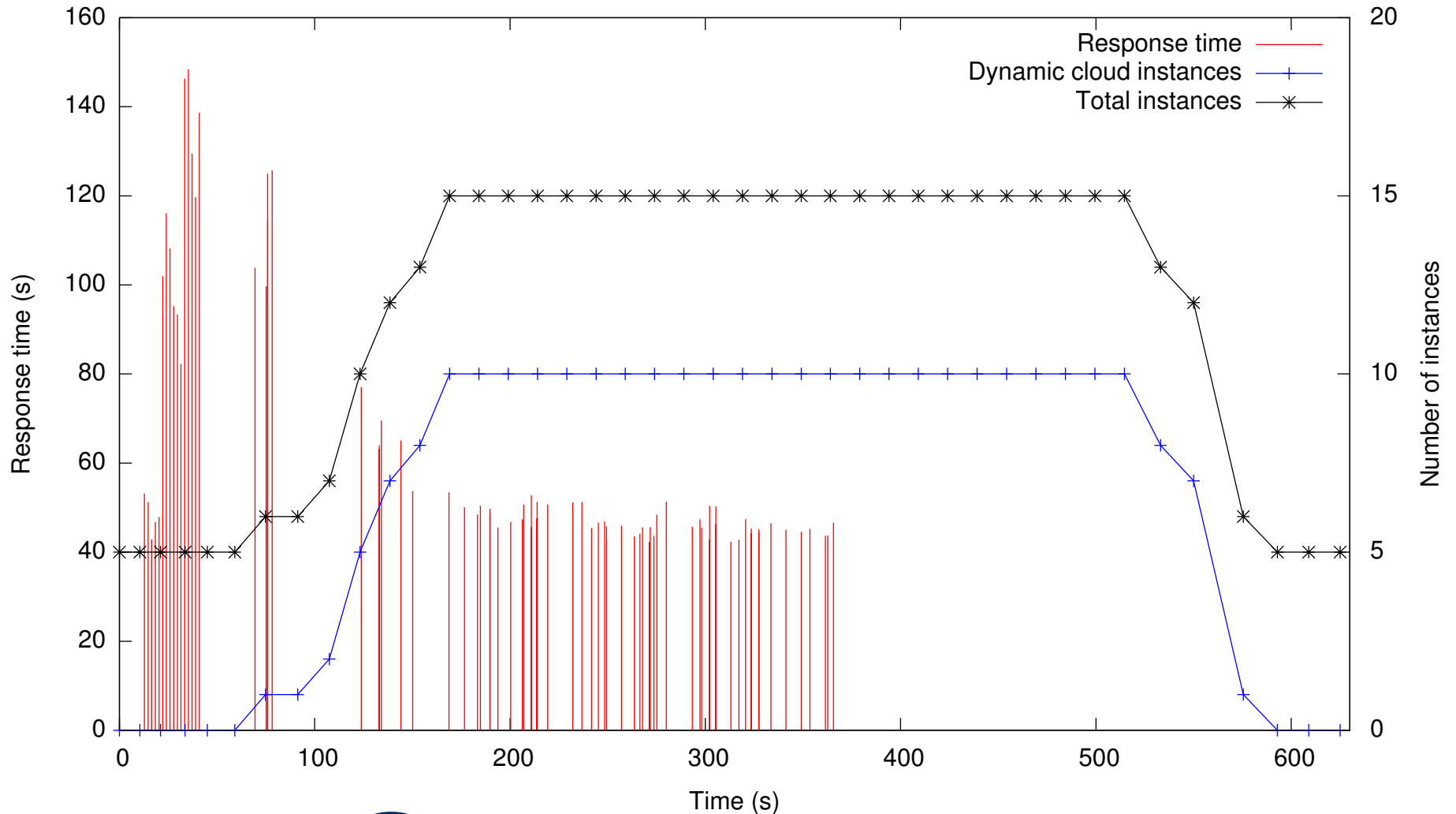
Auto-scaling with 16 users

Impact of concurrent requests



Auto-scaling with 16 users (cntd)

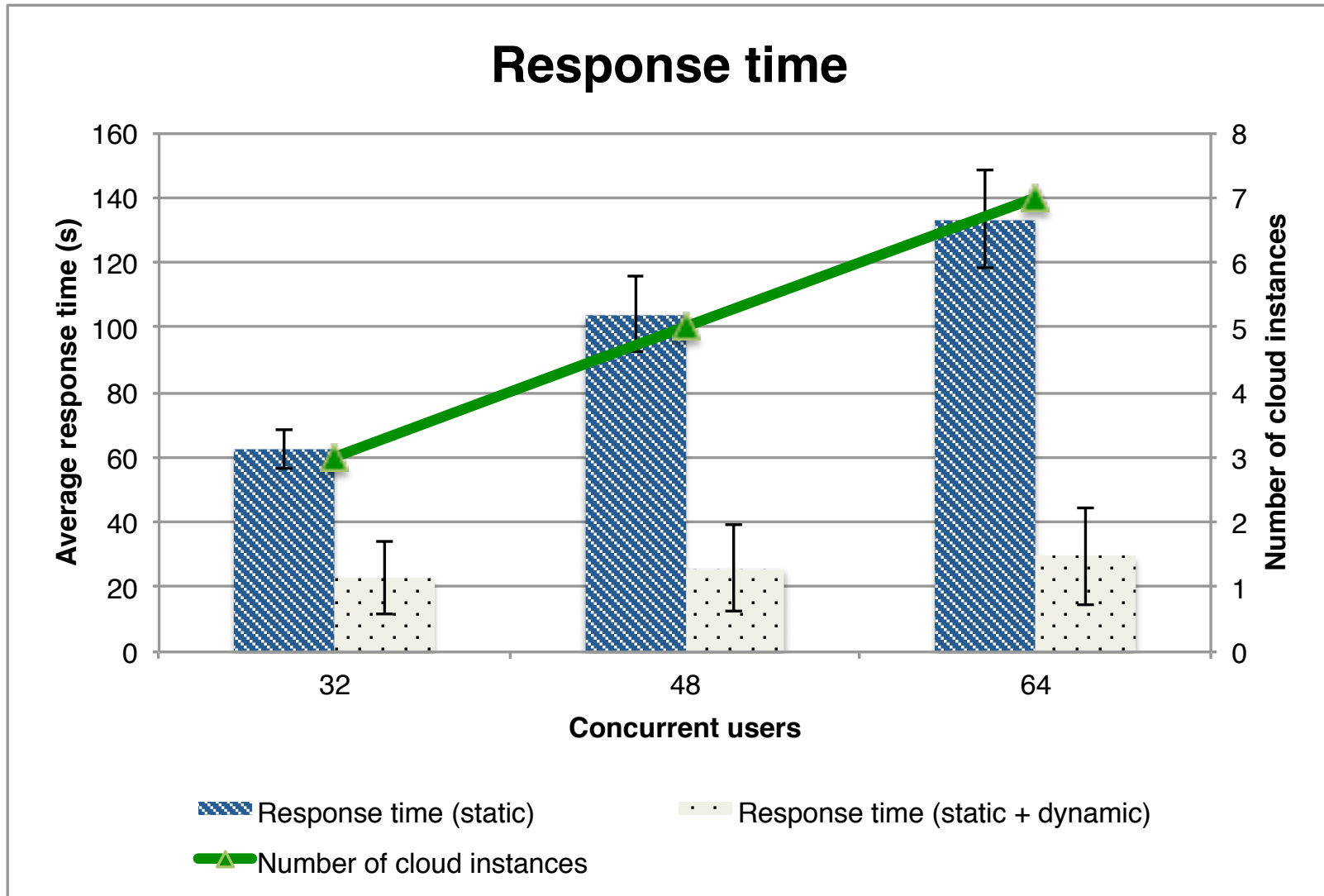
Impact of dynamic cloud instances over response time



Scenario 2 (Small Requests)

- Number of users from 32 to 64
- 5 requests per user
- 10 second pause between requests
- Single static VM
- Maximum of 10 dynamic cloud instances
- 2 minutes auto-scaling history buffer
- 8 concurrent requests per VM

Scenario 2 (cntd)



Summary

- Response time is critical for CyberGIS users
- Requirement for a system that can react to changes in demand
- Integrated Nimbus Phantom auto-scaling
- Maintains low response time
- Future work:
 - Better request management
 - Scaling policy improvements
 - Data storage scalability