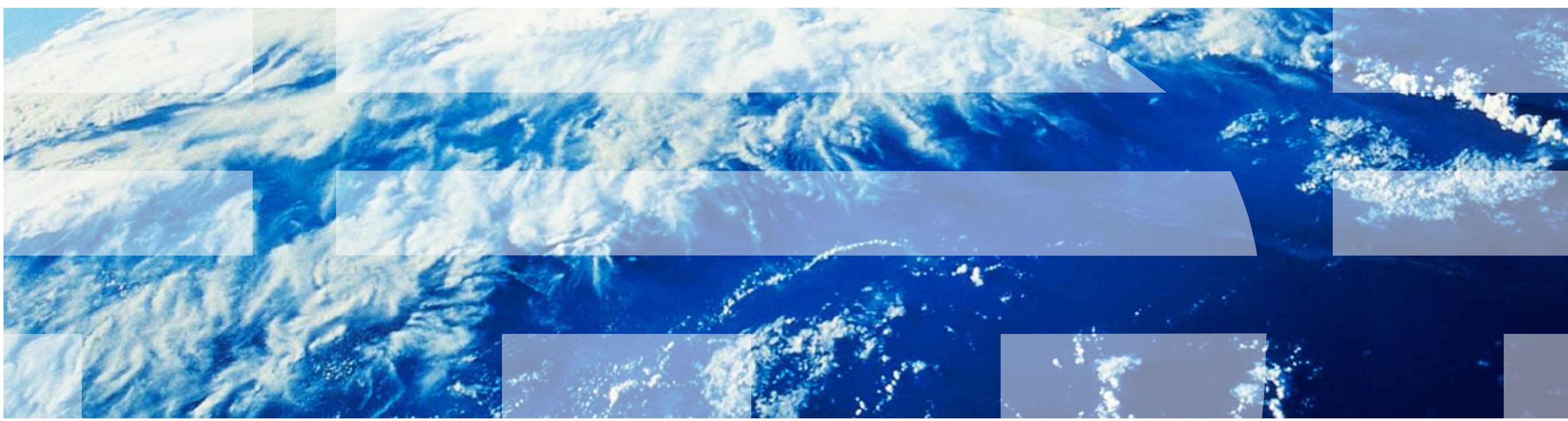
Peter Altevogt (IBM Systems Group Boeblingen)

Cloud Performance Simulations

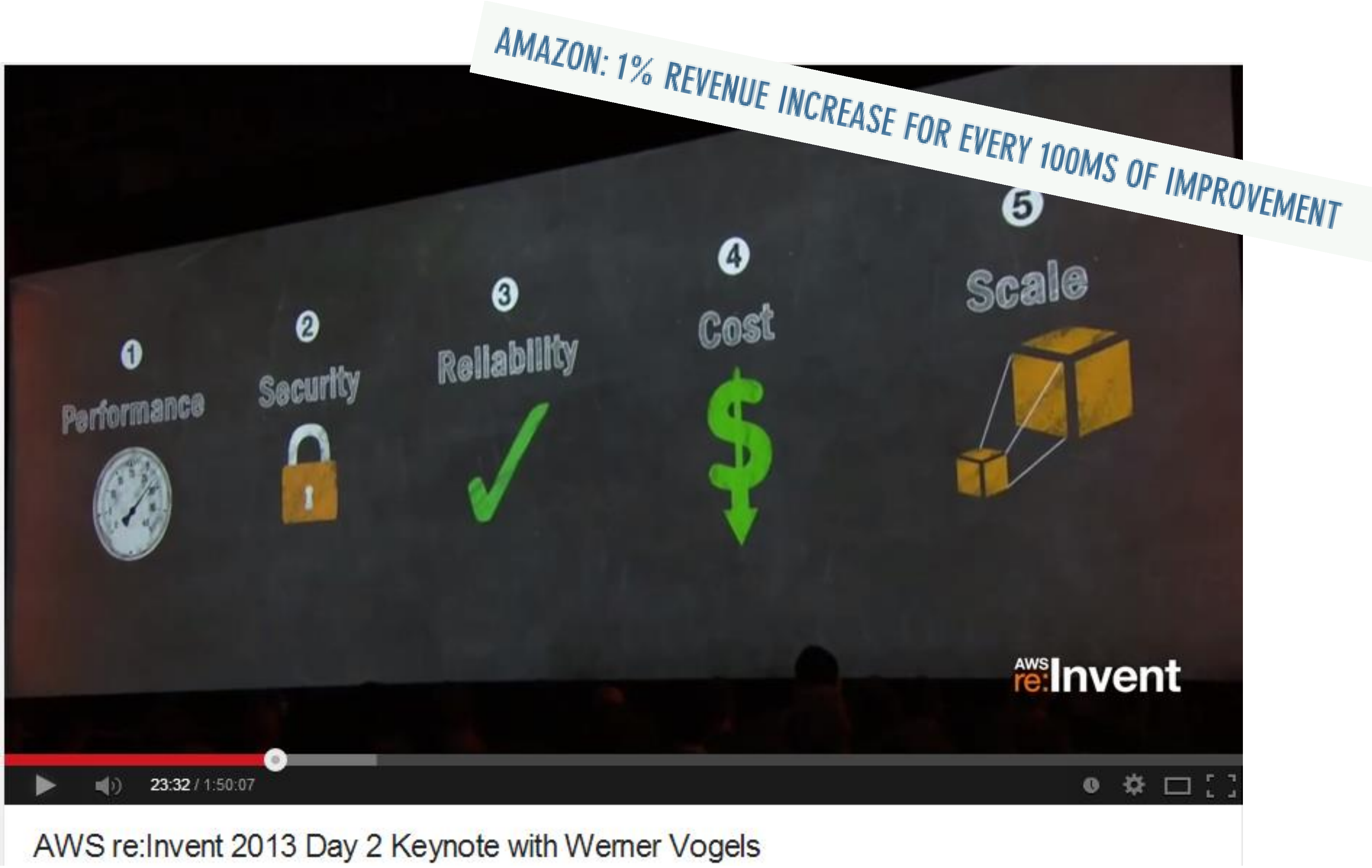


Contents:

- Cloud Performance & Scalabilty
- Cloud Performance Simulations
- The IBM Performance Simulation Framework for Cloud
- Performance Simulation of OpenStack Image Deployment
- Summary
- Outlook

Cloud Performance & Scalability (1/12):

- **Top non-functional requirements against clouds:**



Cloud Performance & Scalability (2/12):

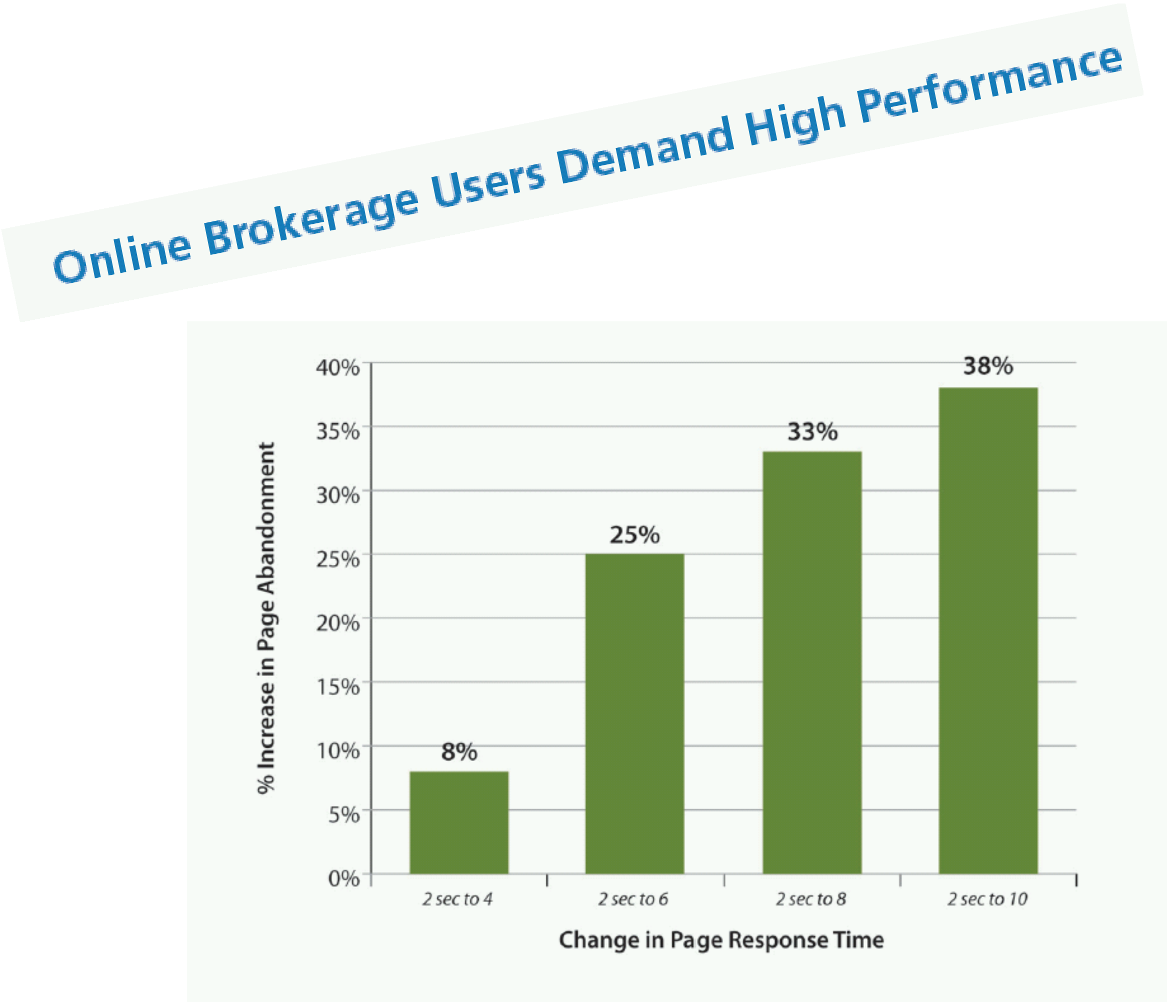
4 out of the top 10 obstacles to growth of cloud computing are performance and scalability related*:

Table 1: Quick Preview of Top 10 Obstacles to and Opportunities for Growth of Cloud Computing.

	Obstacle	Opportunity
1	Availability of Service	Use Multiple Cloud Providers; Use Elasticity to Prevent DDOS
2	Data Lock-In	Standardize APIs; Compatible SW to enable Surge Computing
3	Data Confidentiality and Auditability	Deploy Encryption, VLANs, Firewalls; Geographical Data Storage
4	Data Transfer Bottlenecks	FedExing Disks; Data Backup/Archival; Higher BW Switches
5	Performance Unpredictability	Improved VM Support; Flash Memory; Gang Schedule VMs
6	Scalable Storage	Invent Scalable Store
7	Bugs in Large Distributed Systems	Invent Debugger that relies on Distributed VMs
8	Scaling Quickly	Invent Auto-Scaler that relies on ML; Snapshots for Conservation
9	Reputation Fate Sharing	Offer reputation-guarding services like those for email
10	Software Licensing	Pay-for-use licenses; Bulk use sales

**Above the clouds: A Berkeley view of cloud computing.* (Armbrust, M., et al 2009)

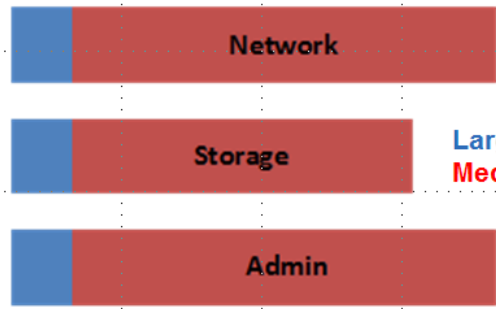
Cloud Performance & Scalability (3/12):



Cloud Performance & Scalability (4/12):

Economies of Scale *

- 2006 comparison of very large service with mid-size: (~1000 servers):



Large Service [$13/Mb/s/mth]: $0.04/GB
Medium [$95/Mb/s/mth]: $0.30/GB (7.1x)

Large Service: $4.6/GB/year (2x in 2 Datacenters)
Medium: $26.00/GB/year* (5.7x)

Large Service: Over 1.000 servers/admin
Enterprise: ~140 servers/admin (7.1x)

- Large block h/w purchases significantly more economic
  - Large weekly purchases offer significant savings
  - H/W Manufacturers willing & able to do custom designs at scale
- Automation & custom s/w investments amortize well at scale
- Summary: scale economics strongly in play

***James Hamilton, VP Amazon Web Services**

Cloud Performance & Scalability (5/12):

- **Some cloud size estimates (number of hosts)...***
  - **Google: > 900 000**
  - **Amazon: > 450 000**
  - **Microsoft: > 1000 000**
  - **Facebook: >> 100 000**

* http://www.datacenterknowledge.com/archives/2009/05/14/whos-got-the-most-web-servers/

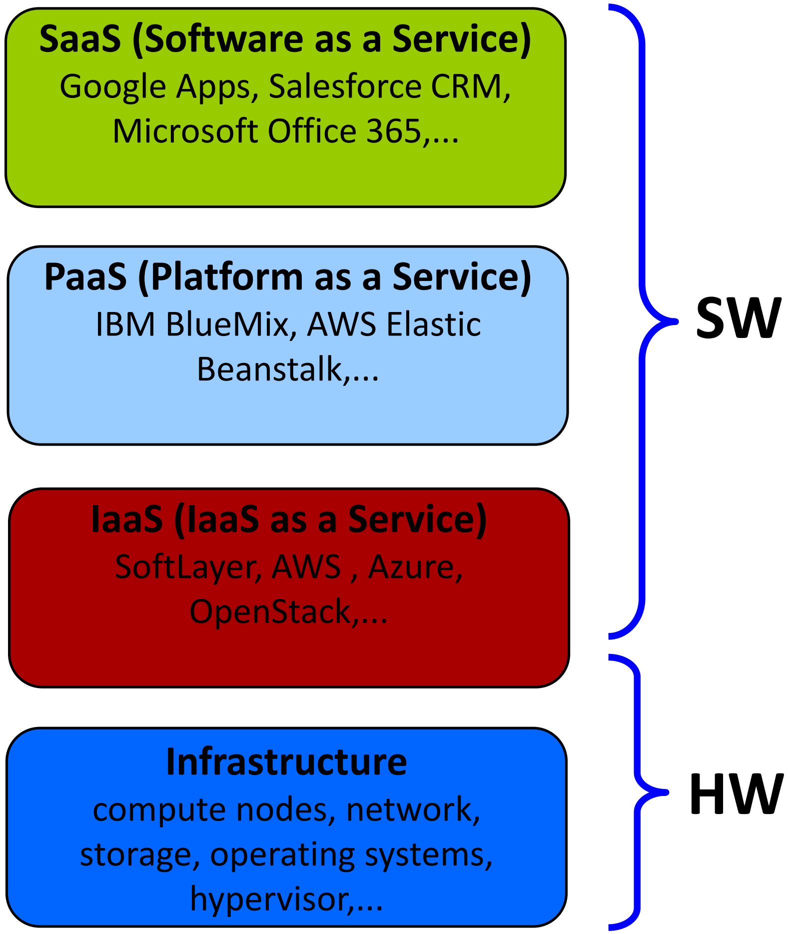
Cloud Performance & Scalability (6/12):

- **Some workload estimates...**
  - **Number of number of mobile system in 2016: 10^{10}**
  - **Concurrently active mobile system assessing the cloud: 10^{8}**
  - **IOPS per active mobile system: 5**

→ cloud will need approx. 10^{6} 15k SAS disks
... + compute nodes + network devices

How do we design and manage clouds at this scale?

Cloud Performance & Scalability (7/12):



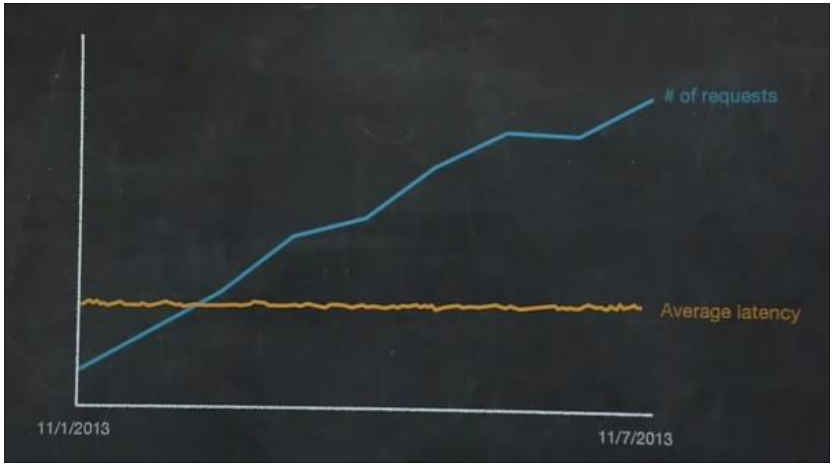
Cloud Performance & Scalability (8/12): Objectives

- **The user requires *consistent* performance**

  e.g. response times independent of time of day, load,...

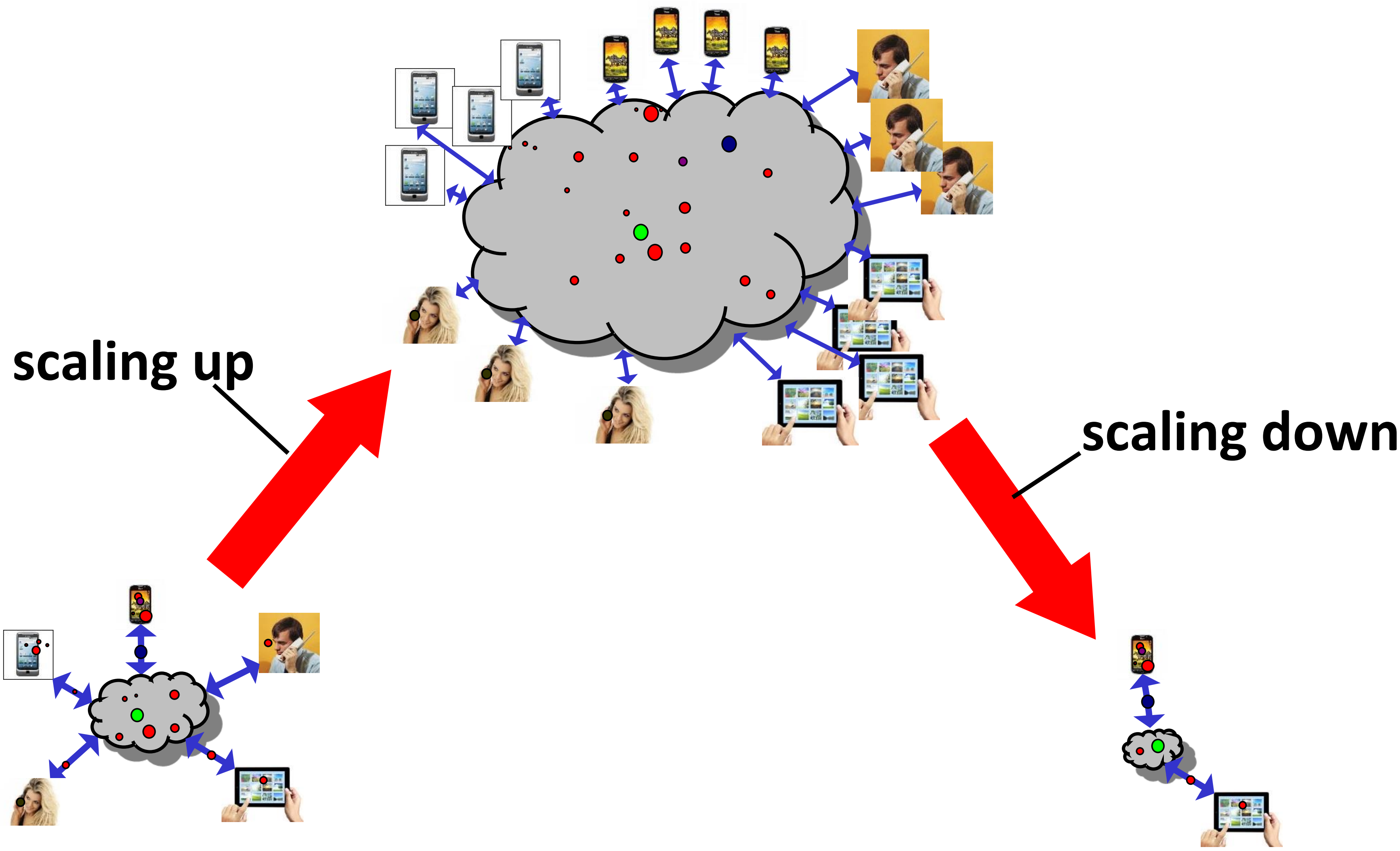
- **The provider needs to provide this with *minimal resources***

  → elasticity



Example: Dynamo DB

Cloud Performance & Scalability (9/12): Elasticity



Cloud Performance & Scalability (10/12):

- **Architectural Patterns:**
  - Horizontal scaling (cluster)
    - stateless autonomous compute nodes
  - Auto-Scaling
    - scale-up and ***scale-down*** of resources (elasticity)
  - Queue-centric workflow pattern
    - loose coupling, asynchronous delivery of requests,...
  - Eventual Consistency
    - CAP theorem, ACID vs. eventual consistent,...

...intricate interactions between HW and SW

Cloud Performance & Scalability (11/12):

- **Some key Software Performance issues:**

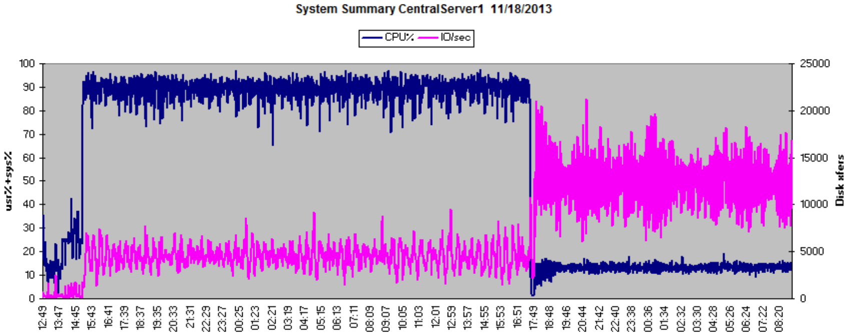
  – Scalability
    - support scale-out architectures
    - ensure consistent performance as a function of data cardinalities (users, number of VMs,...)

  – Data access
    - choice of database technology

      SQL vs. NoSQL, ACID vs. eventual consistent,...
    - database design

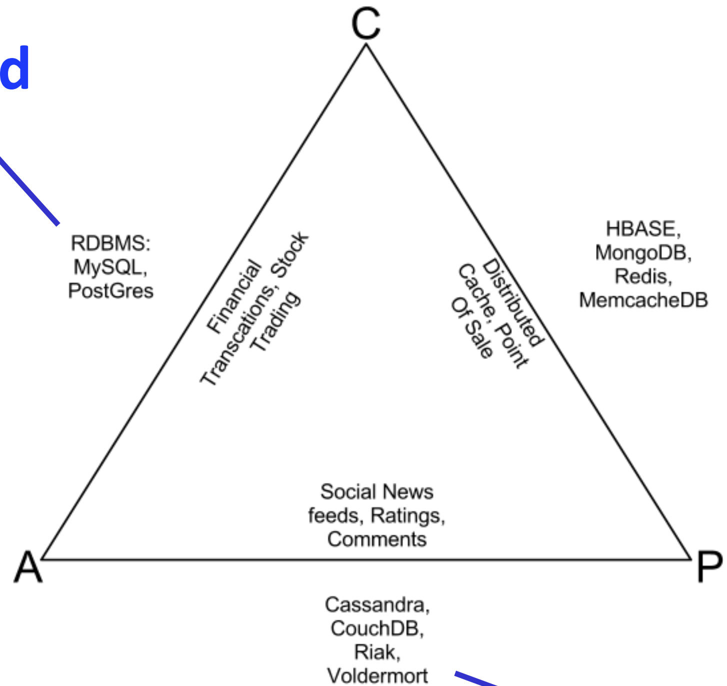
      schema, indices,...

  – Software resources
    - locks, tokens, connections



Cloud Performance & Scalability (12/12): CAP Theorem

Systems-of-record



Systems-of-engagement

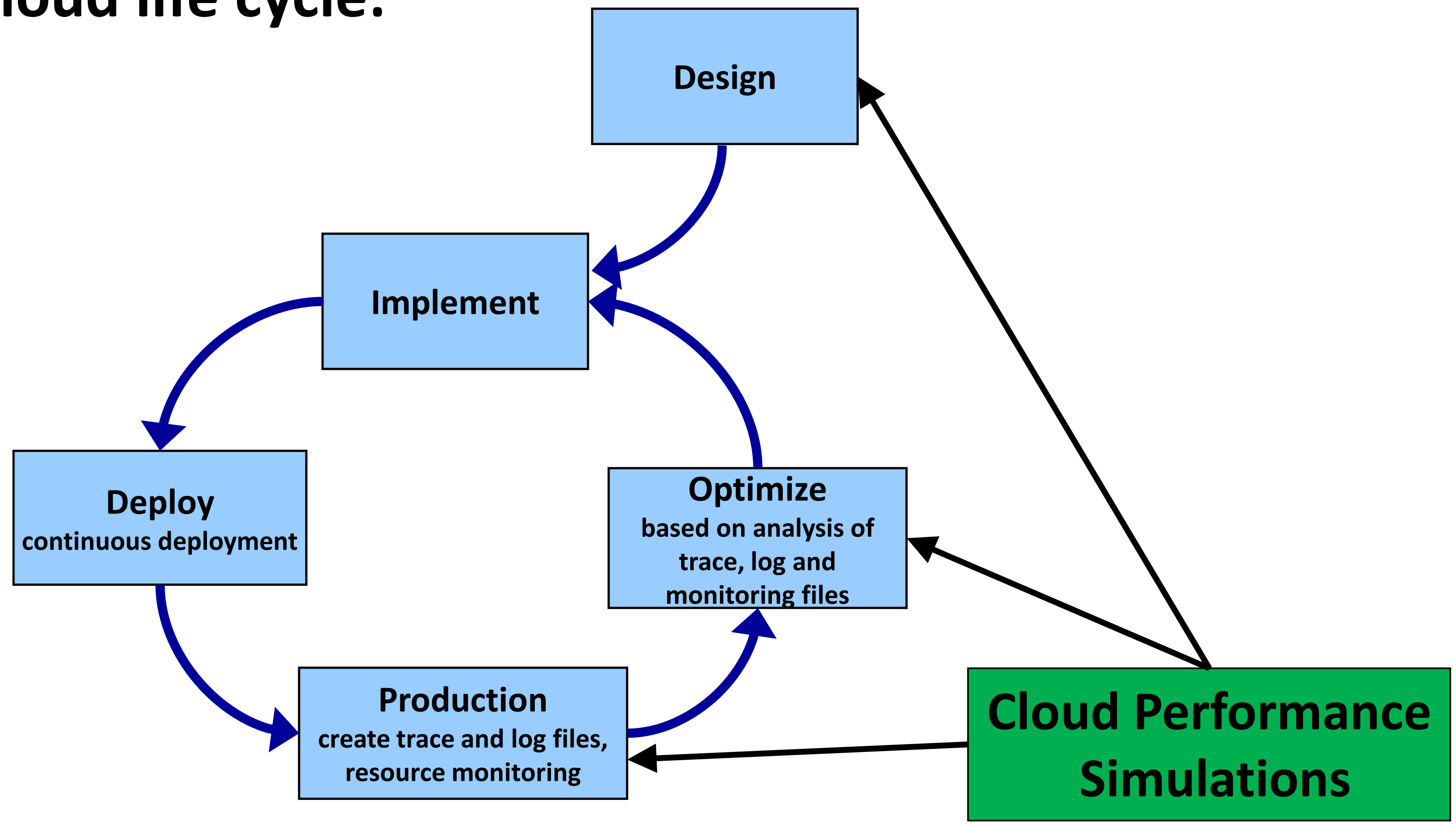
How to reconcile these 2 worlds in the cloud?

Contents:

- Cloud Performance & Scalabilty
- **Cloud Performance Simulations**
- The IBM Performance Simulation Framework for Cloud
- Performance Simulation of OpenStack Image Deployment
- Summary
- Outlook

Cloud Performance Simulations (1/3):

- **Cloud life cycle:**



Key issue: no performance testing at scale!

Cloud Performance Simulations (2/3): Design Time

- **Who?**
  - cloud designers, architects, consulants, researcher,…
- **Why?**
  - ensure good & consistent performance to end users
  - minimize resource provisioning for cloud provider
- **What?**
  - capacity planning
    - how many concurrently active VMs can be supported?
  - new algorithms
    - scalability of new deployment heuristics?
  - impact of new upcoming infrastructure
    - network, storage, processors,…
- **How?**
  - leverage appropriate frameworks for simulation, modeling resp. capacity planning

Cloud Performance Simulations (3/3): Run Time

- **Who?**
  - cloud administrators
  - autonomous (self-adaptive)
- **Why?**
  - ensure good & consistent performance to end users
  - minimize resource provisioning for cloud provider
- **What?**
  - adapt capacity
    - to support fluctuating number of concurrently active VMs can be supported
  - impact of new upgrading (downsizing) infrastructure
    - network, storage, processors,…
- **How?**
  - leverage appropriate frameworks

Contents:

The IBM Perf. Simulation Framework for Cloud (1/):

- **Objectives:**
  - leverage discrete-event simulation technologies for cloud computing to
    - support the design of scalable and performant clouds
    - extend scope of cloud performance engineering beyond measurements & tuning

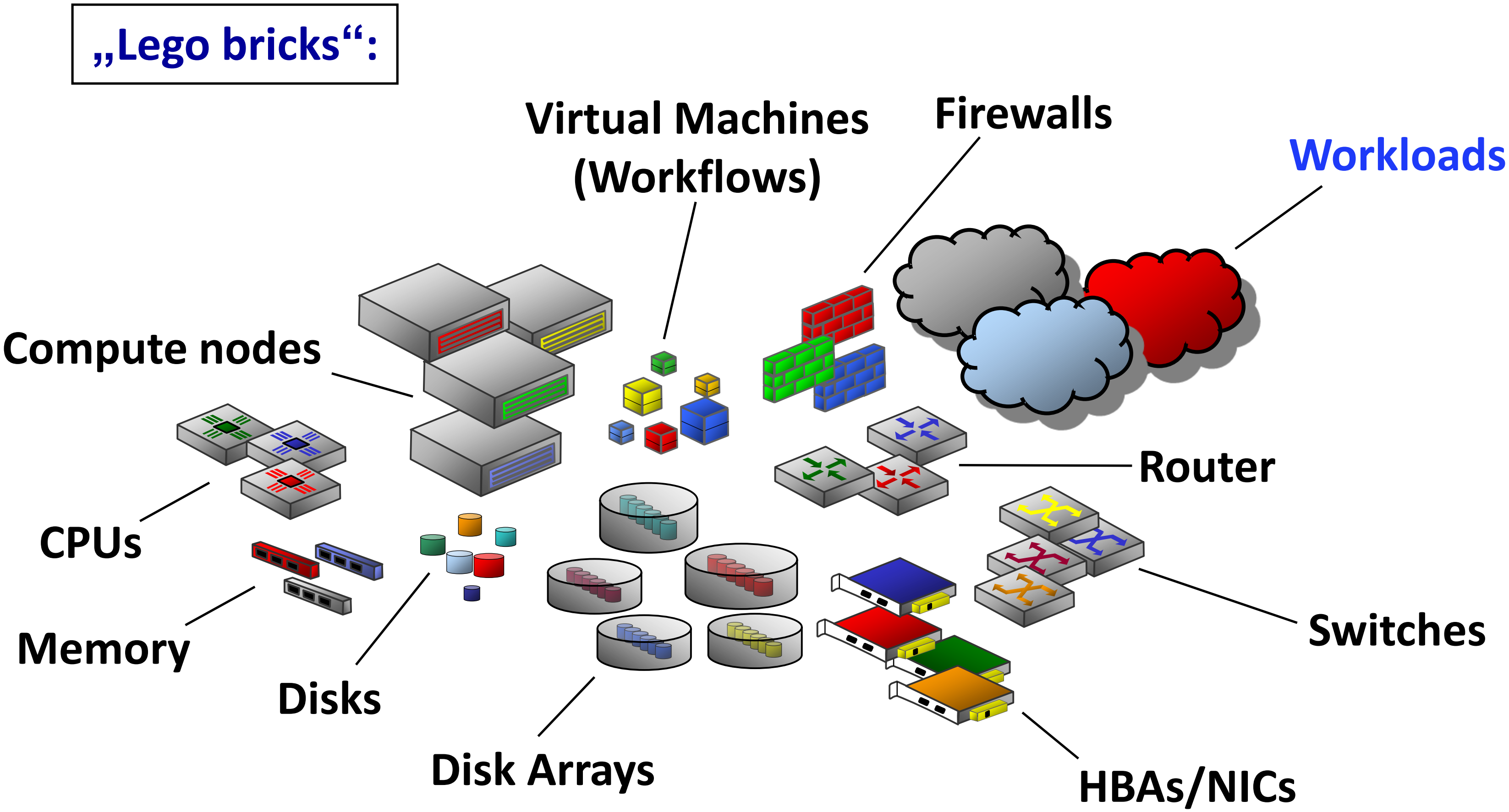
- **Current Status:**
  - IBM internal tool applied to various IBM cloud solutions

- **Collaborators:**
  - W. Denzel, T. Kiss (IBM Research Zurich)

- **Publications:**
  - Modular Performance Simulations of Clouds. *P. Altevogt, W. Denzel and T. Kiss, Proc. of the 2011 Winter Simulation Conference (WSC'11),* S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu, eds…
  - Cloud Modeling and Simulations. *P. Altevogt, W. Denzel and T. Kiss, Encyclopedia of Cloud Computing, 1 edition (Wiley, 2016),* S. Murugesan, I. Bojanova (eds)

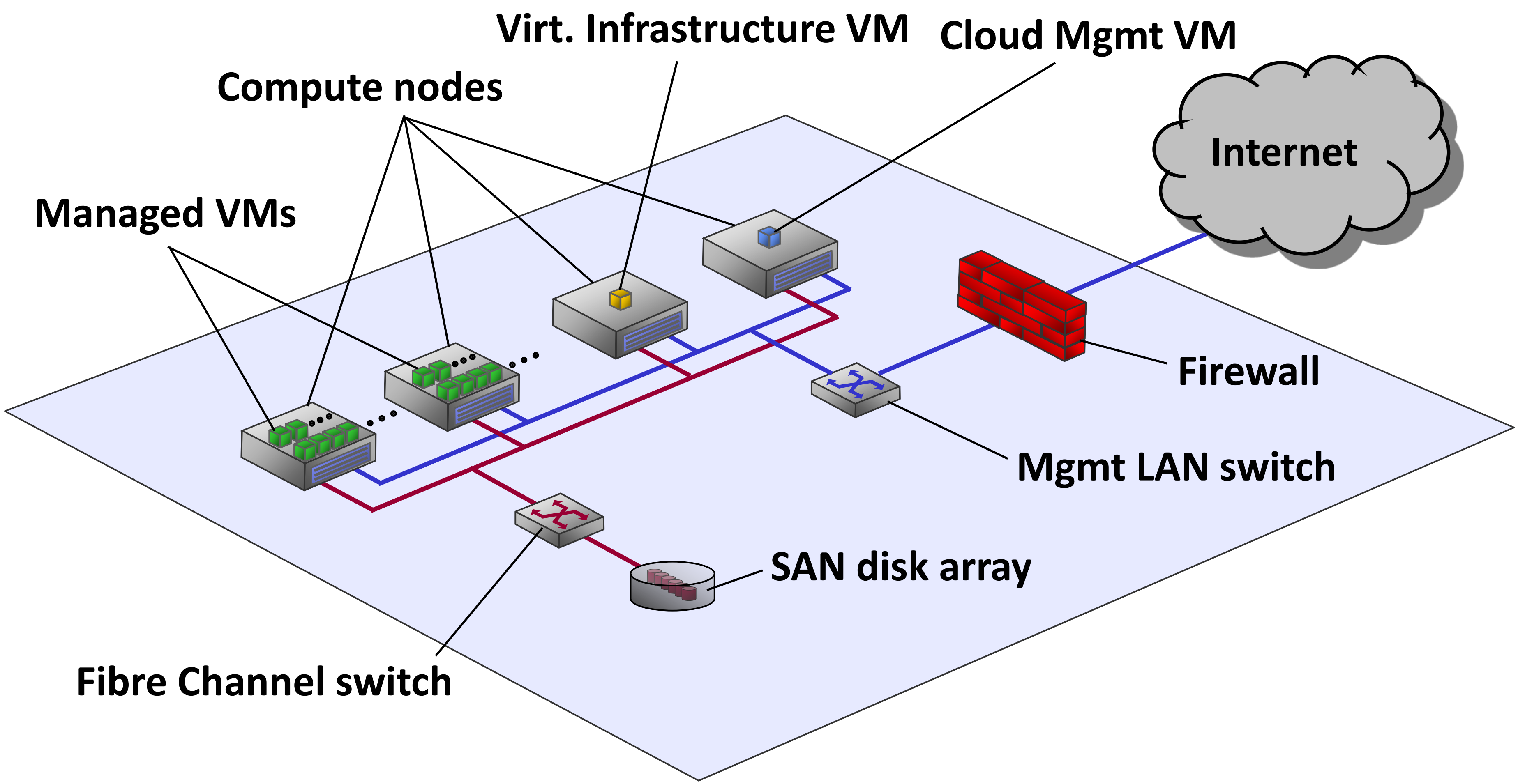
The IBM Perf. Simulation Framework for Cloud (2/):

- **<u>Step1</u>: create simulation models for cloud components**



The IBM Perf. Simulation Framework for Cloud (3/):

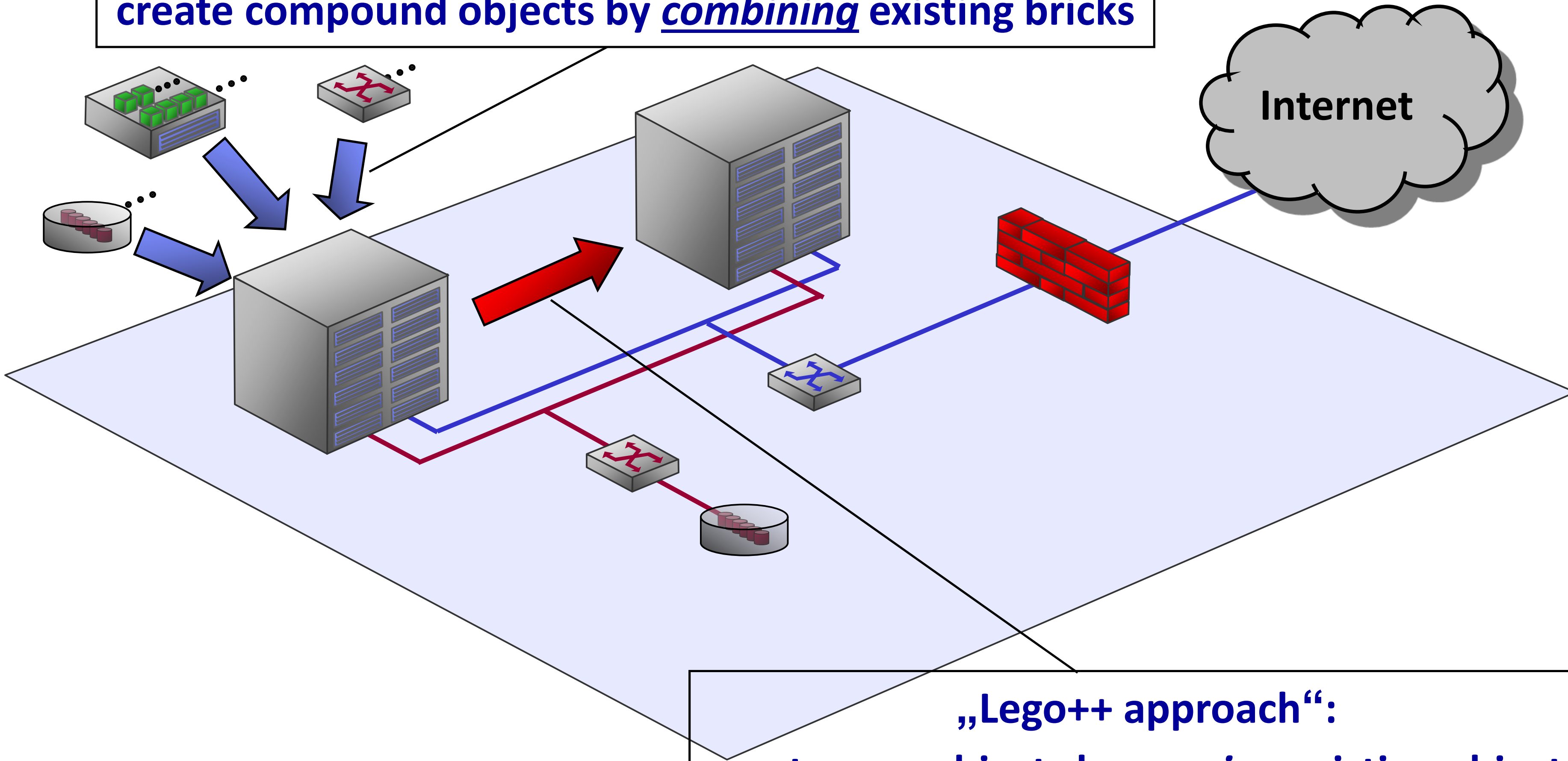
- **<u>Step 2</u>: combine „Lego bricks“ to create cloud models**



The IBM Perf. Simulation Framework for Cloud (4/):

- **<u>Step 2 (cont‘d):</u>**

„Lego approach“:
create compound objects by <u>*combining*</u> existing bricks

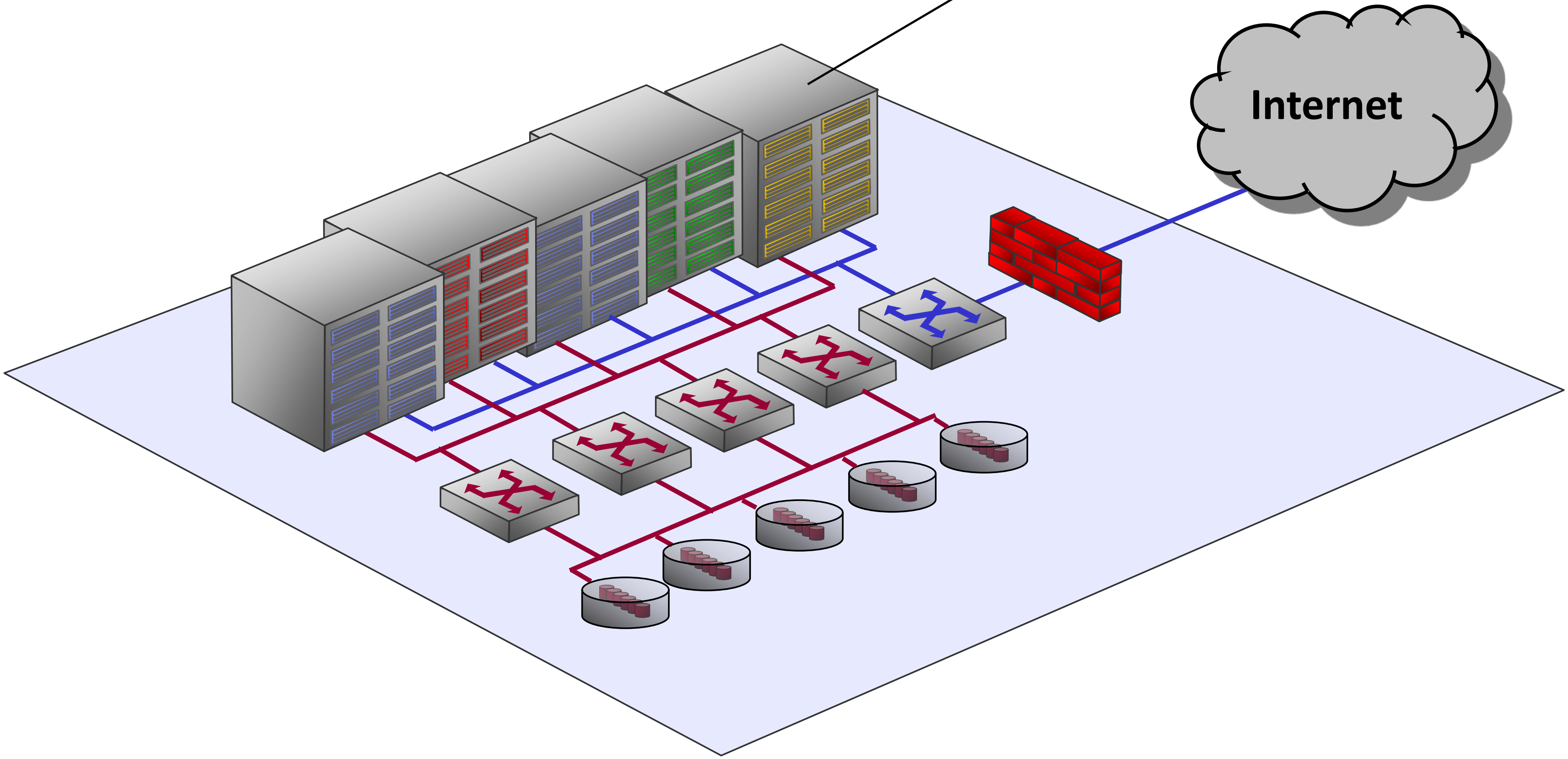


„Lego++ approach“:
create new objects by <u>*copying*</u> existing objects

The IBM Perf. Simulation Framework for Cloud (5/):

- **Step 2 (cont‘d):**

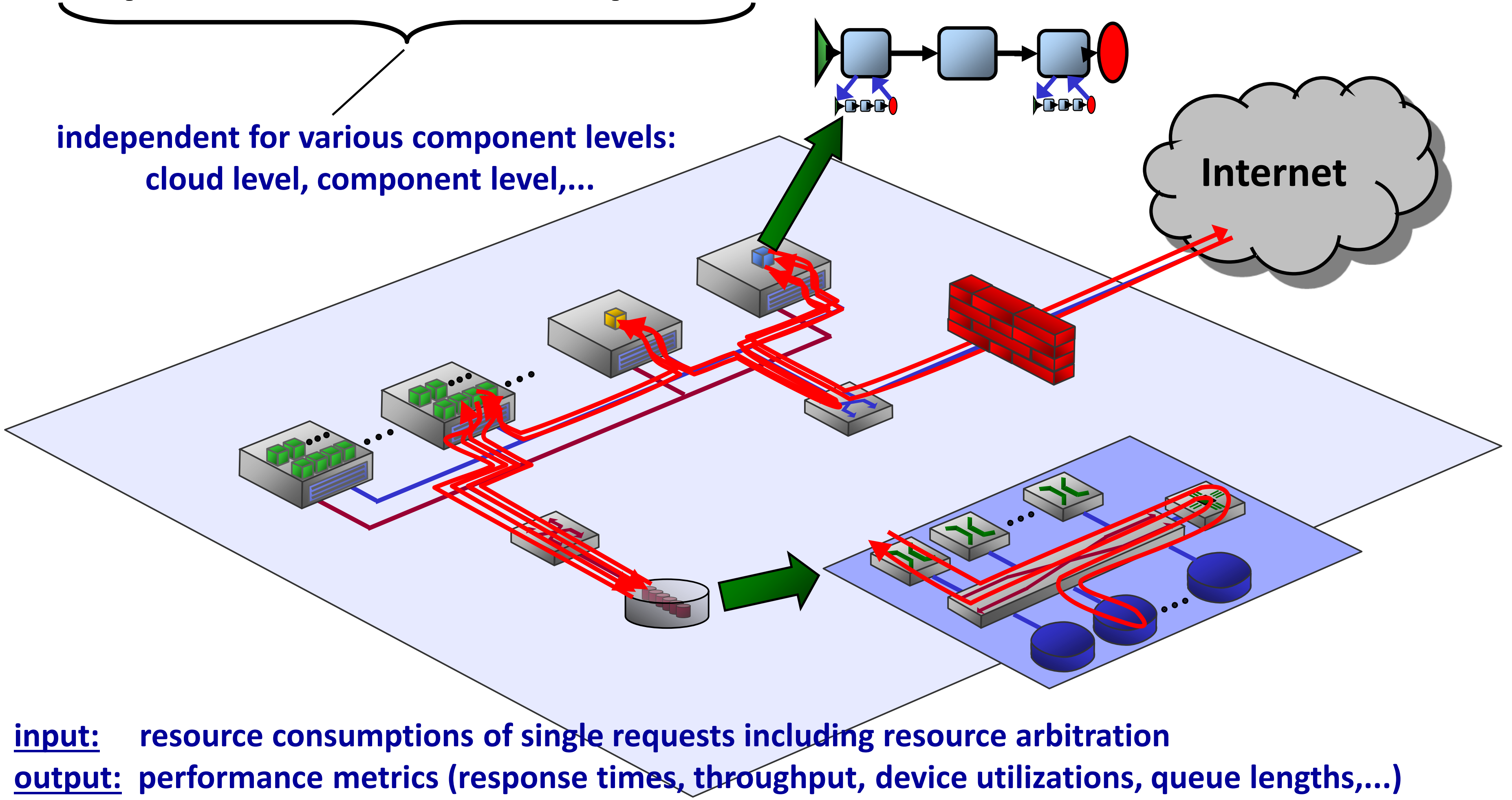
Example 1:
create data center cloud by „copy and paste“ of racks, switches, disk arrays,...



The IBM Perf. Simulation Framework for Cloud (6/):

- **Steps 3 & 4:**
  implement flow of requests and execute simulations...

independent for various component levels:
cloud level, component level,...

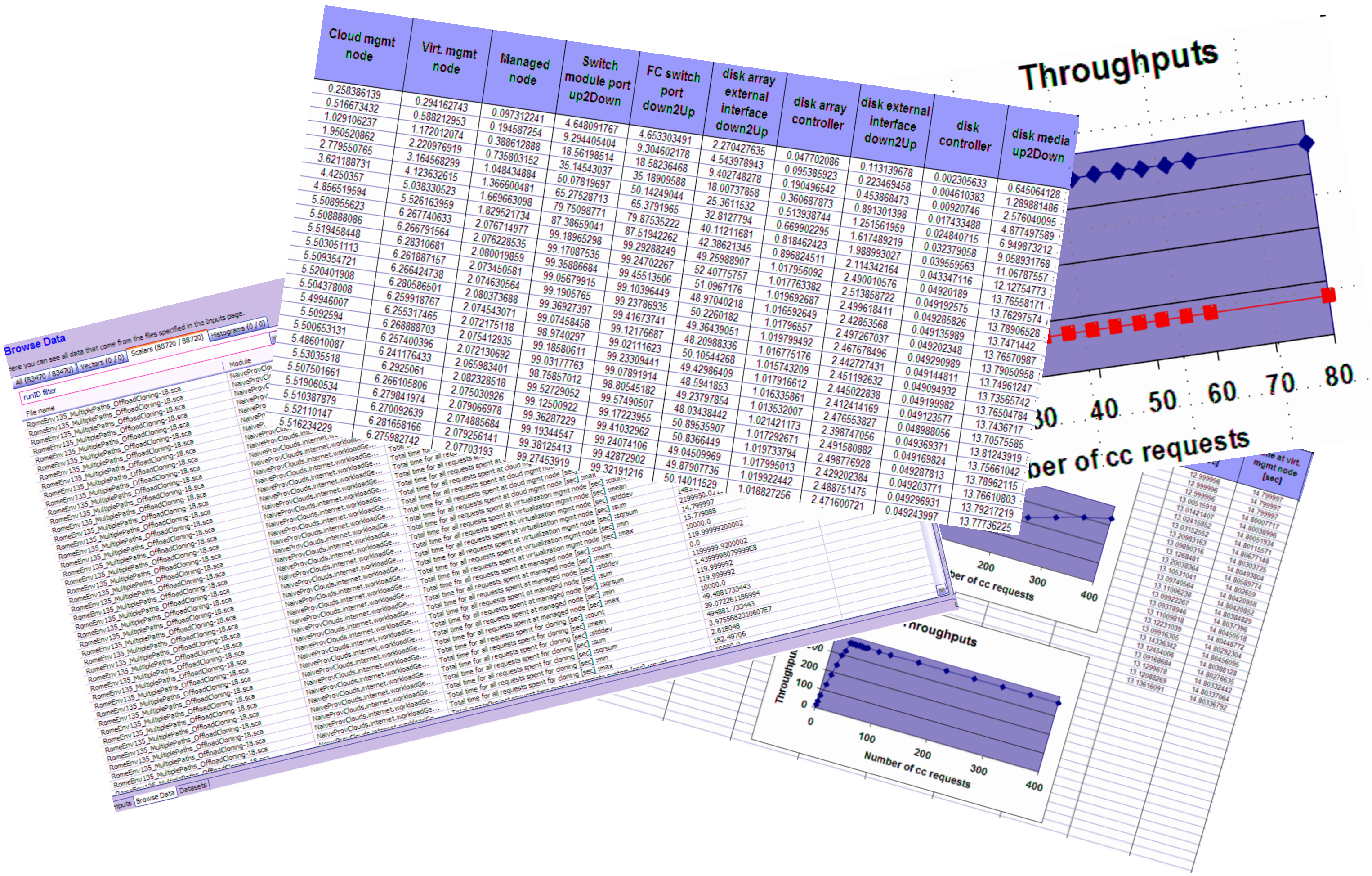


input: resource consumptions of single requests including resource arbitration

output: performance metrics (response times, throughput, device utilizations, queue lengths,...) for various request workloads and system architectures

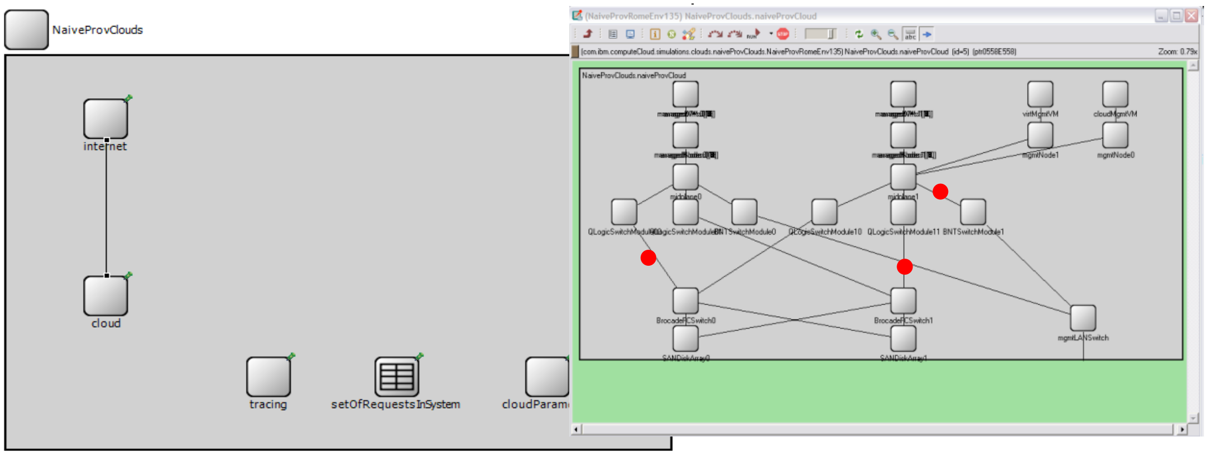
The IBM Perf. Simulation Framework for Cloud (7/):

- **<u>Step 5</u>: analyze and publish simulation results...**



The IBM Perf. Simulation Framework for Cloud (2/):

- **Simulation Technology:**
  - discrete-event simulations
- **Simulation Tooling:**
  - OMNEST Simulation Software:
    - widely used within IBM: Research and Systems Group
    - scalable: supports parallel simulations on a cluster
    - successfully applied to simulate very large systems (>> 65000 components)
    - public domain --> large user group and knowledge base



The IBM Perf. Simulation Framework for Cloud (2/):

- **Highlights:**
  - **outstanding scalability and performance**
    - core modules implemented in C++
    - script language for combining core modules and specifying complex networks algorithmically
    - supporting parallel simulation
  - **modularity (Lego++ approach):**
    - new cloud components may be easily added, existing ones changed
    - take some components from the shelf to start building clouds...
    - replicate...
  - **treats all cloud components as „first class citizens“**
  - **supports:**
    - high-level coarse-grain modeling
    - fast prototyping as well as <u>*arbitrary*</u> accuracy where required

Contents:

- Cloud Performance & Scalabilty
- Cloud Performance Simulations
- The IBM Performance Simulation Framework for Cloud
- **Performance Simulation of OpenStack Image Deployment**
- Summary
- Outlook

Introduction:

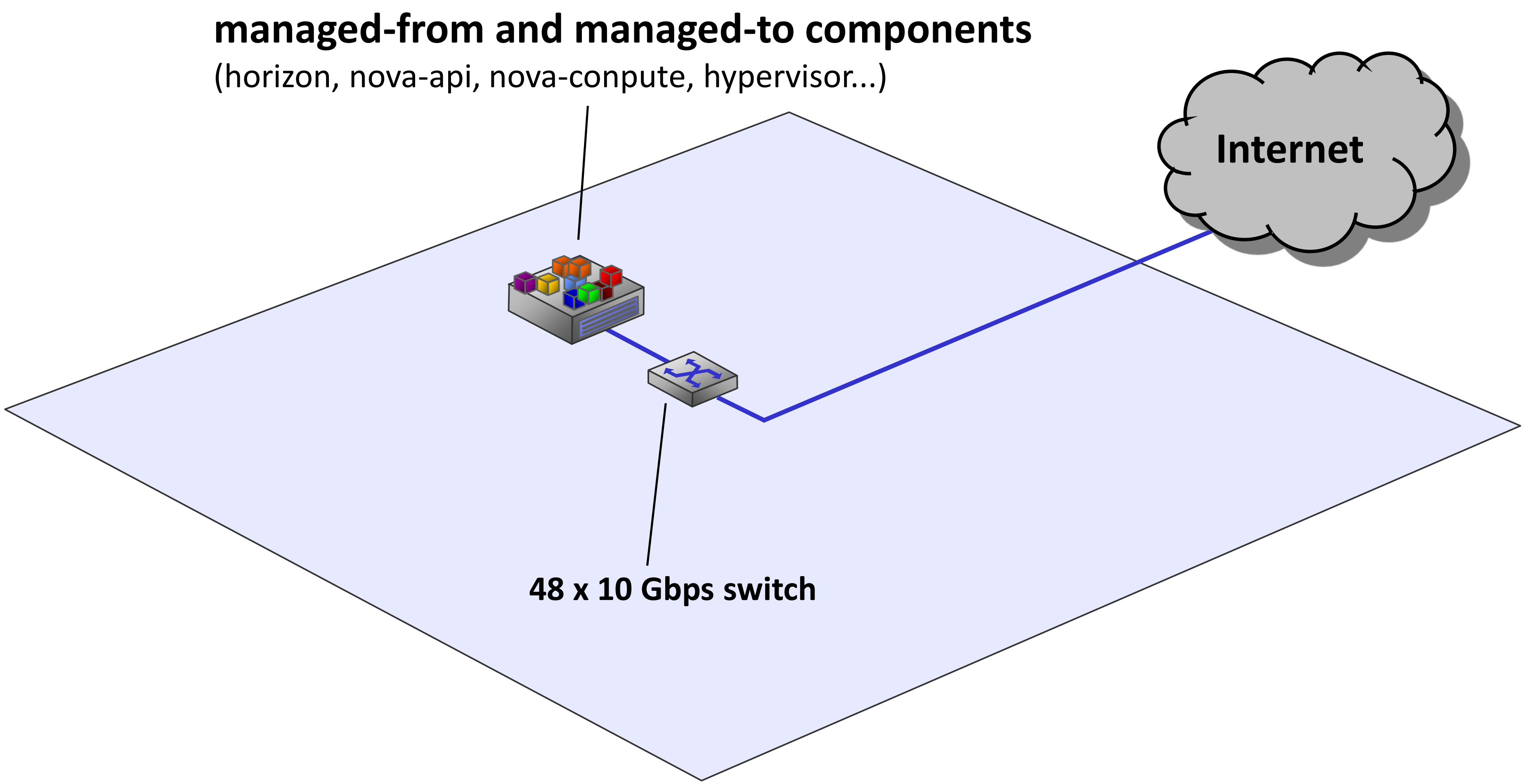
- **Objectives**
  - estimate end2end image deployment performance for various OpenStack cloud architectures
  - Identify bottlenecks for various architecture / load combinations
  - Provide some guidance for design of OpenStack clouds and performance work

- **Methodology**
  - discrete-event simulation parametrized and calibrated by measurements
  - currently focus on relative results versus a baseline

- **Tooling**
  - IBM Performace Simulation Framework for Cloud

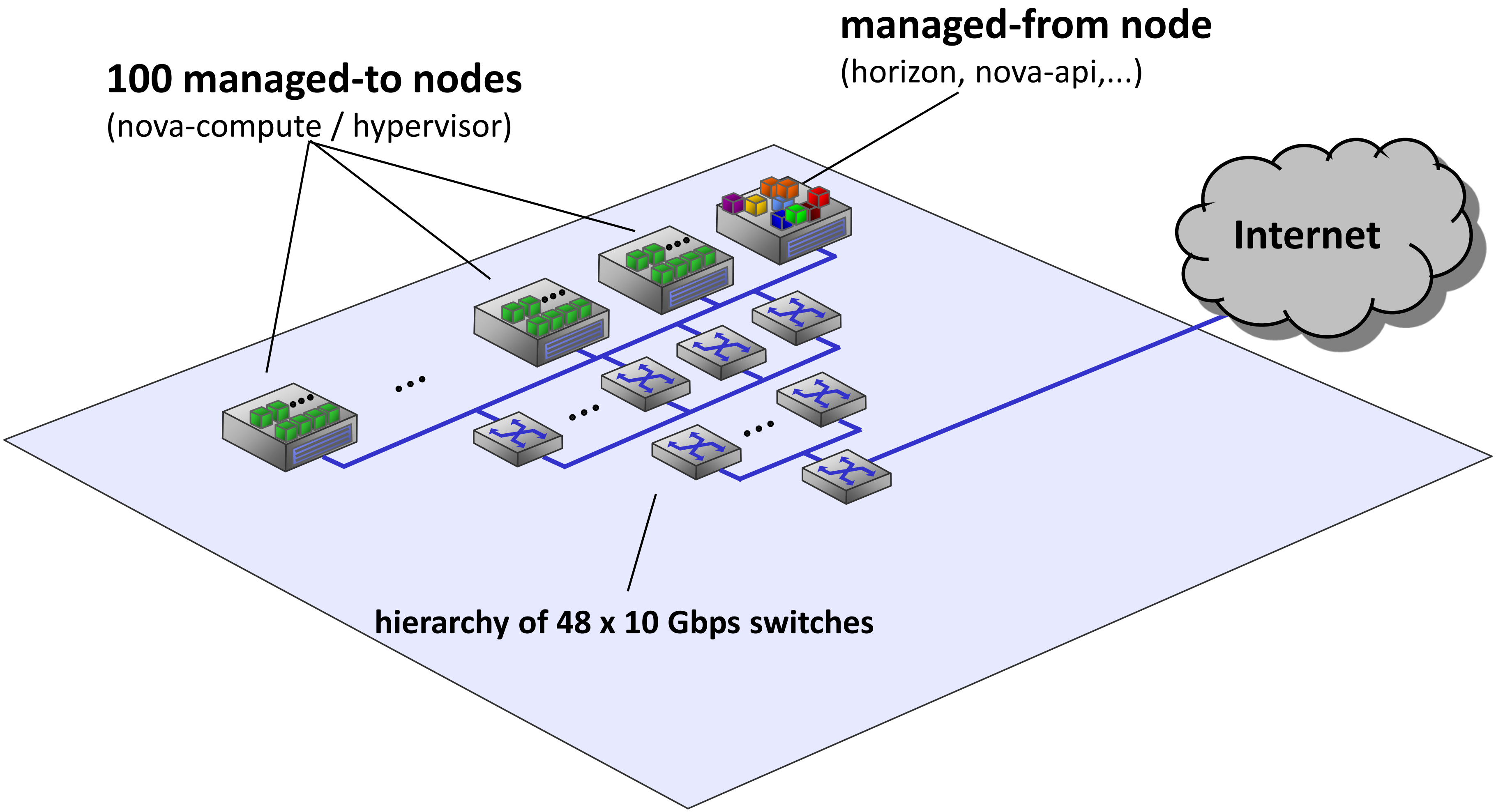
OpenStack Cloud Architectures (1/3):

- **One node for all components („S“):**



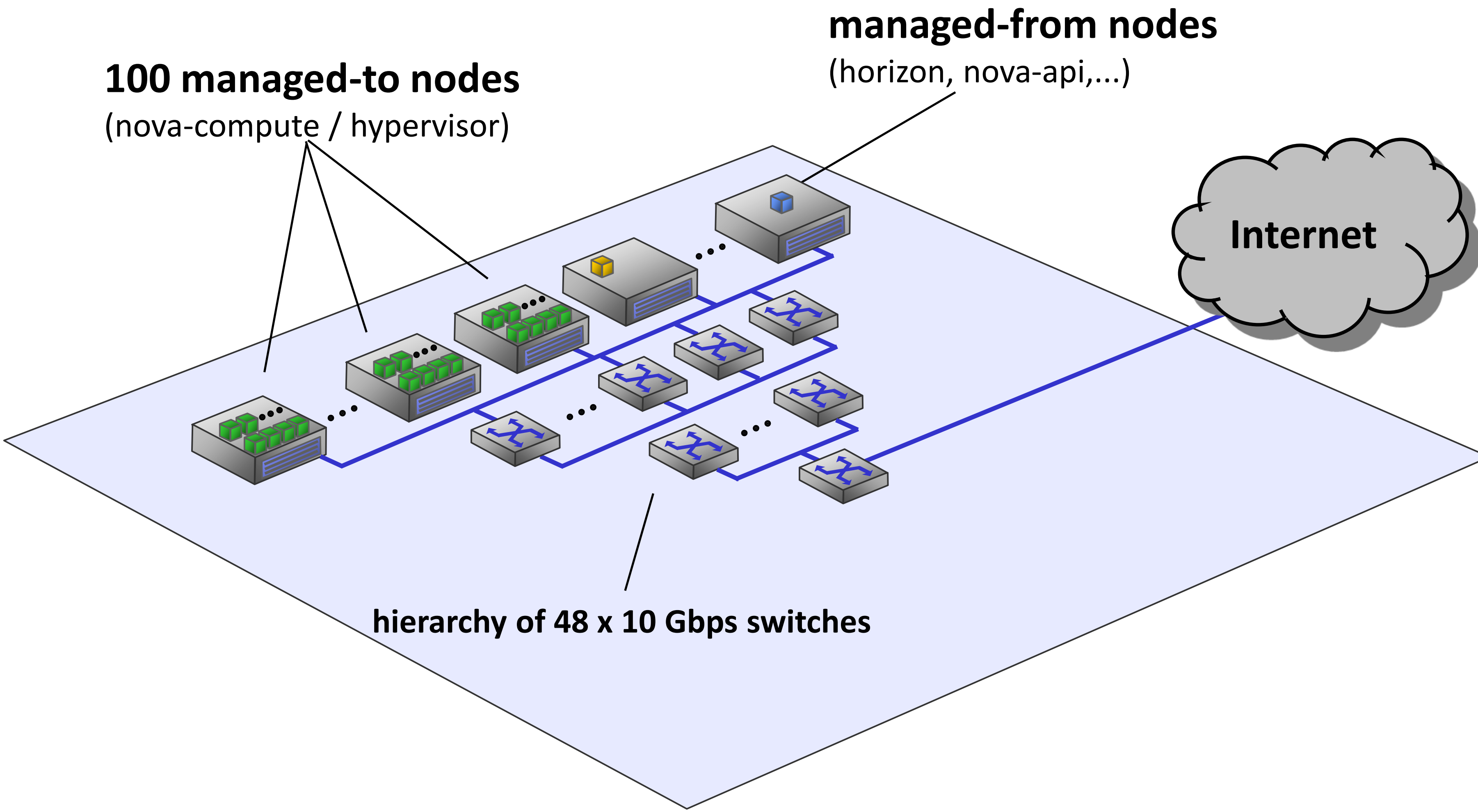
OpenStack Cloud Architectures (2/3):

- **Scale-out of compute nodes only („M“):**

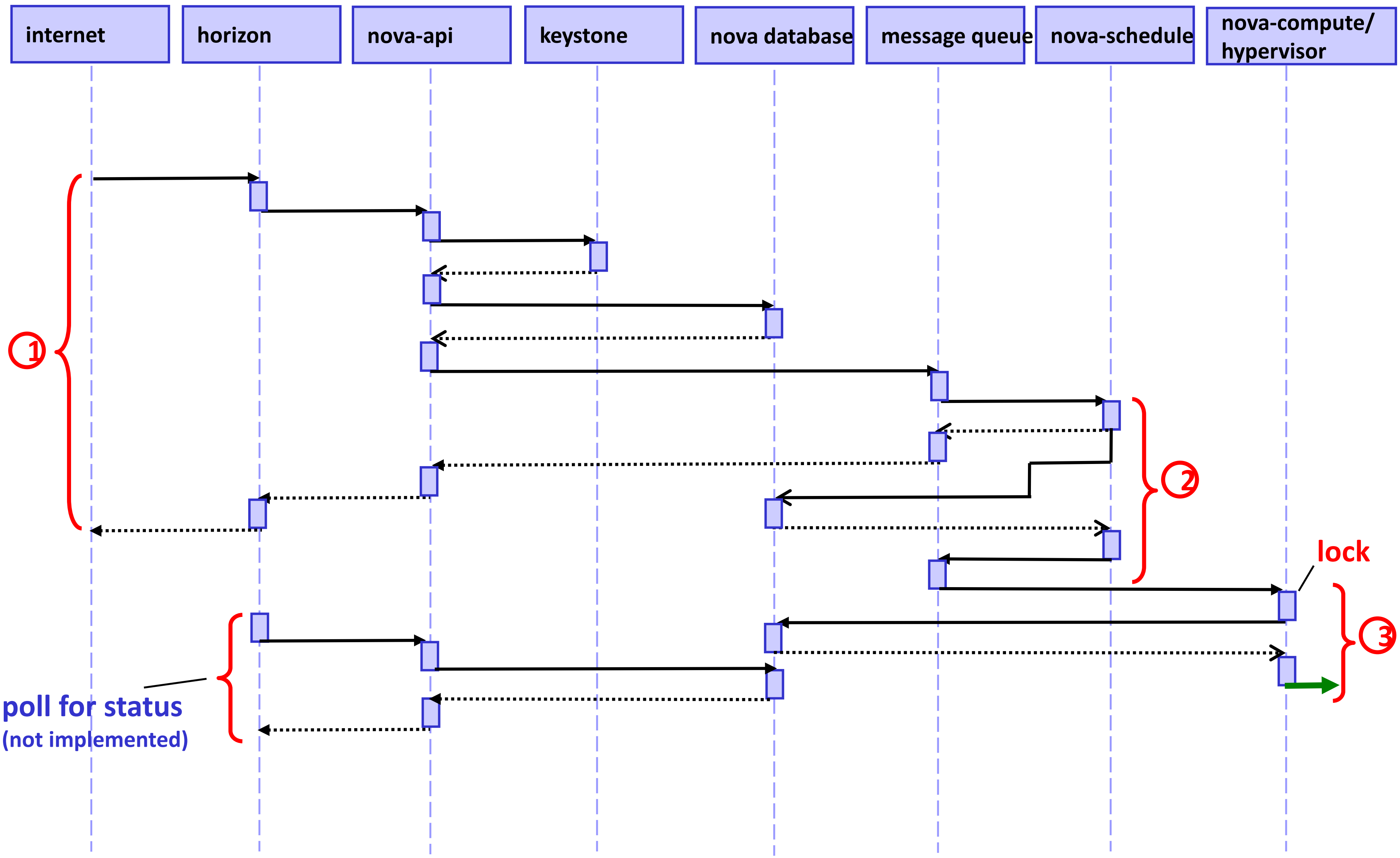


OpenStack Cloud Architectures (2/3):

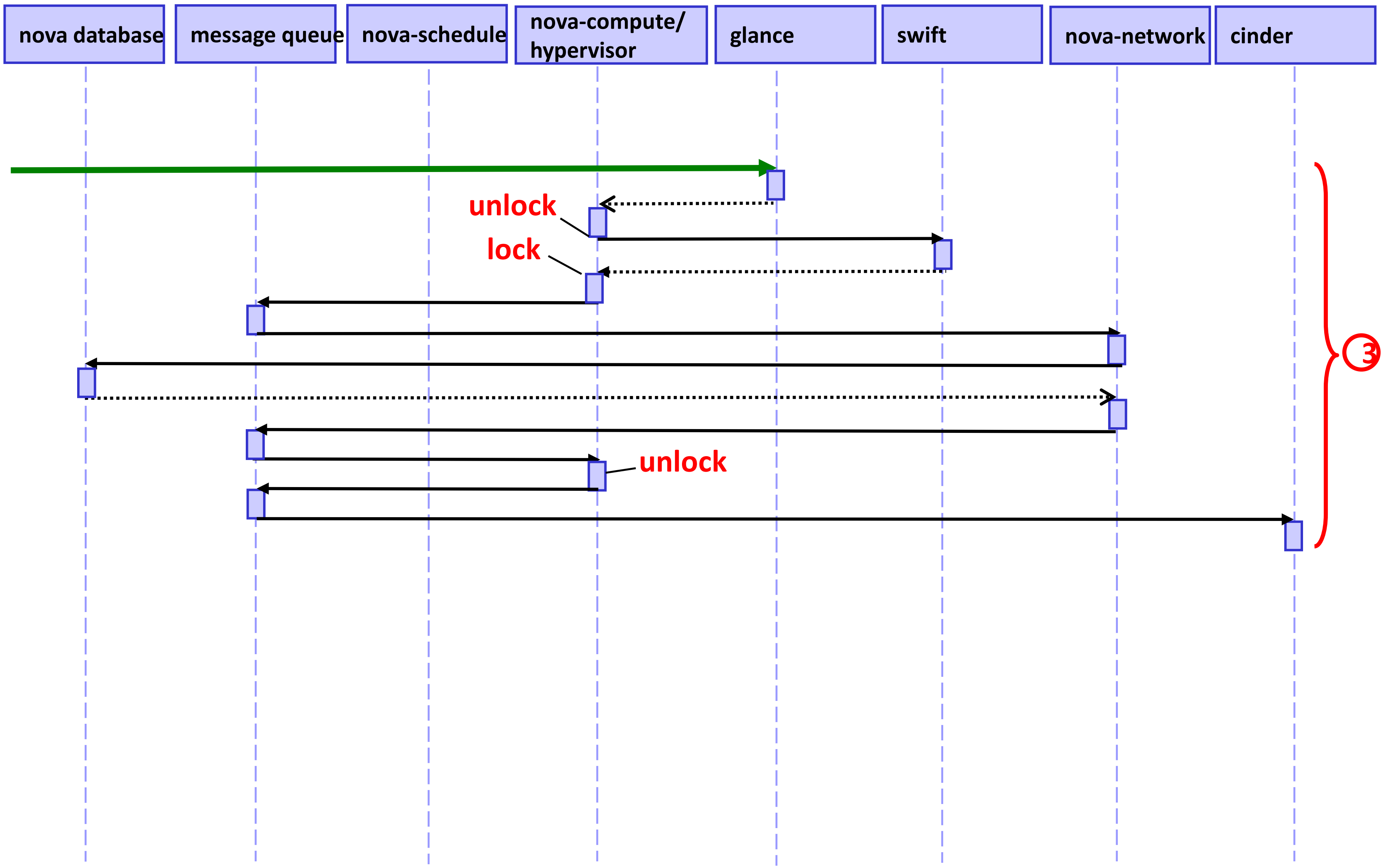
- **Scale-out of compute nodes only („L“):**



Deployment Workflow (1/2):



Deployment Workflow (2/2):



Workload Phases:

Phase	Components involved											Workflow details
	horizon	nova-api	keystone	nova database	message queue	nova-scheduler	nova-compute hypervisor	glance	swift	nova-network	cinder	
1	✓	✓	✓	✓	✓							- authentication - create instance entry in database - send ack to dashboard - forward request to scheduler
2				✓	✓	✓						- create schedule for instance - update entry in database - forward request to compute / hypervisor
3				✓	✓		✓	✓	✓	✓	✓	- retrieve instance infos from database - setup data for hypervisor - process instance request - get image URI from glance - get image from swift - reserve and allocate network - update instance entry in database - provision storage volume for instance - forward request to compute / hypervisor

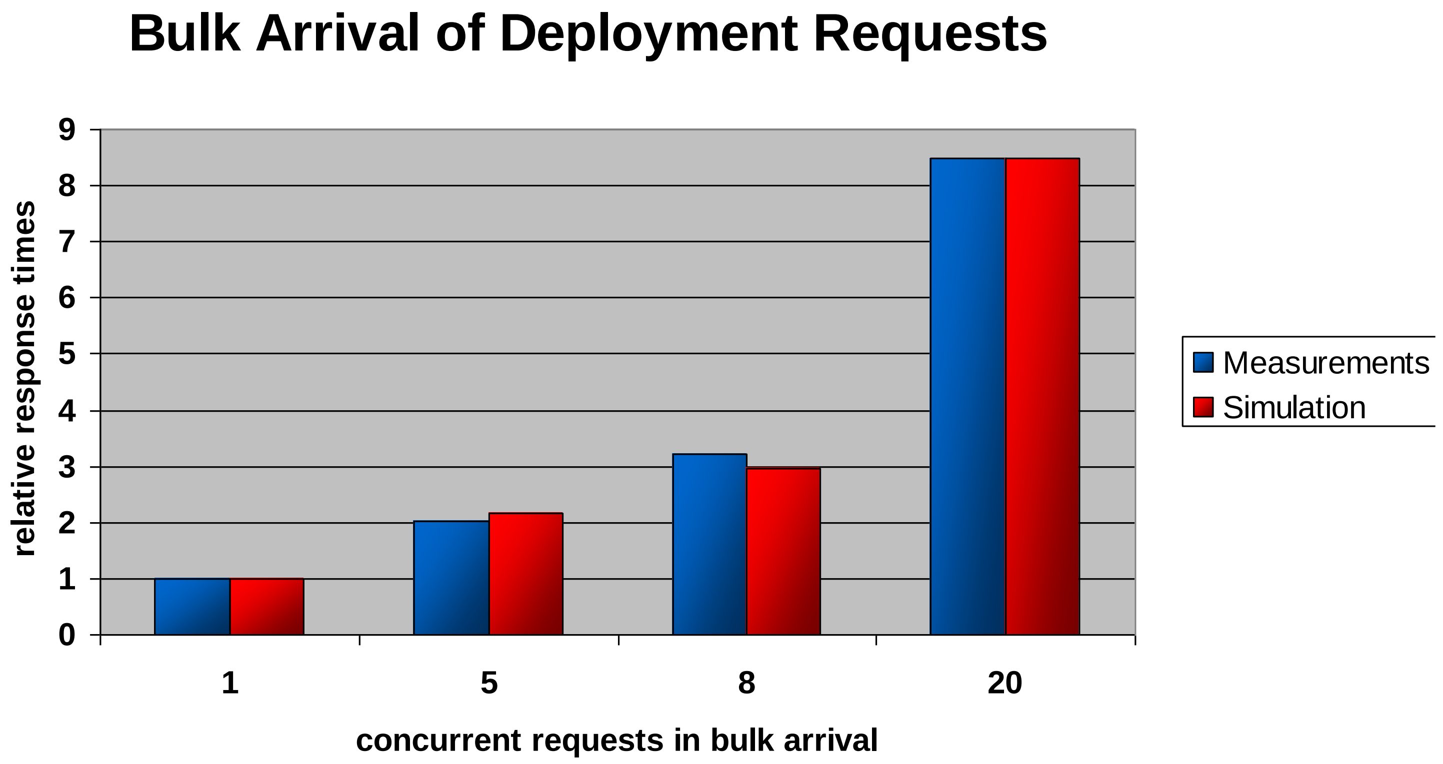
Parameterization and Calibration (1/2):

- **General approach:**
  - use single request measurement data used for parametrization
  - use concurrent requests measurement data used for calibration

- **Current methodology:**
  - start with available (single node) measurements
    - *Scaling the Boot Barrier*: P. Feiner, OpenStack Summit 2013
    - IBM internal data
  - focus on ***relative*** values vs. baseline to factor out specific system features
  - enhance (parametrization of) model using upcoming measurement data

Continuously increase accuracy of simulation by leveraging new measurement results to refine (parametrization of) model!

Parameterization and Calibration (2/2):



Simulation Scenarios:

Nr	Scenario	Cloud Architecture	Workflow	Objective
1	SingleNode	S	standard OS deployment workflow	set baseline
2	SingleNodeNoLocking	S	same as 1, but no locking at nova-compute and hypervisor	assess impact of locking at nova-compute and hypervisor
3	MultiComputeNodes	M	standard OS deployment workflow	Assess impact of scaling out deployment of VMs to various managed-to nodea
4	MultiNodes	X	standard OS deployment workflow	assess impact of scaling out deployment of VMs to various managed nodes and distributing all management components

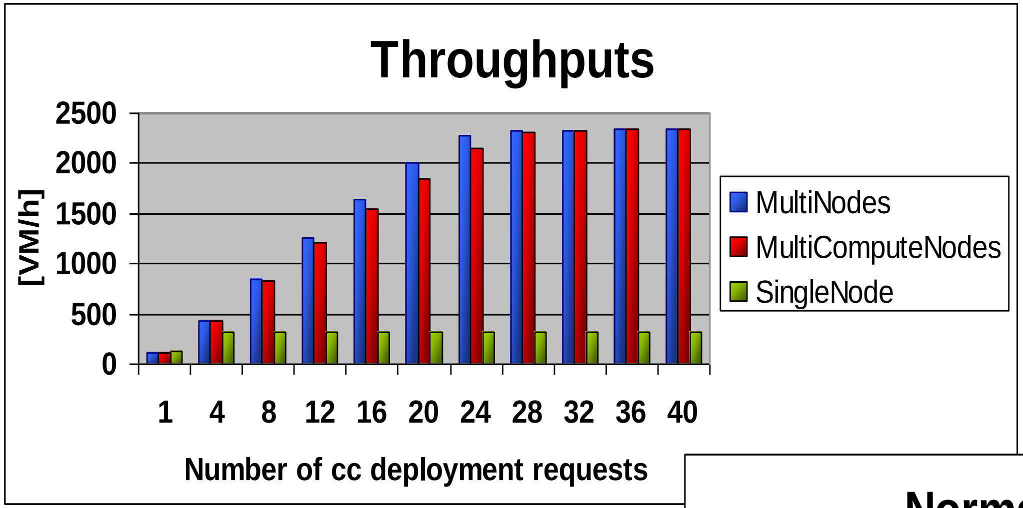
Simulation Execution Characteristics:

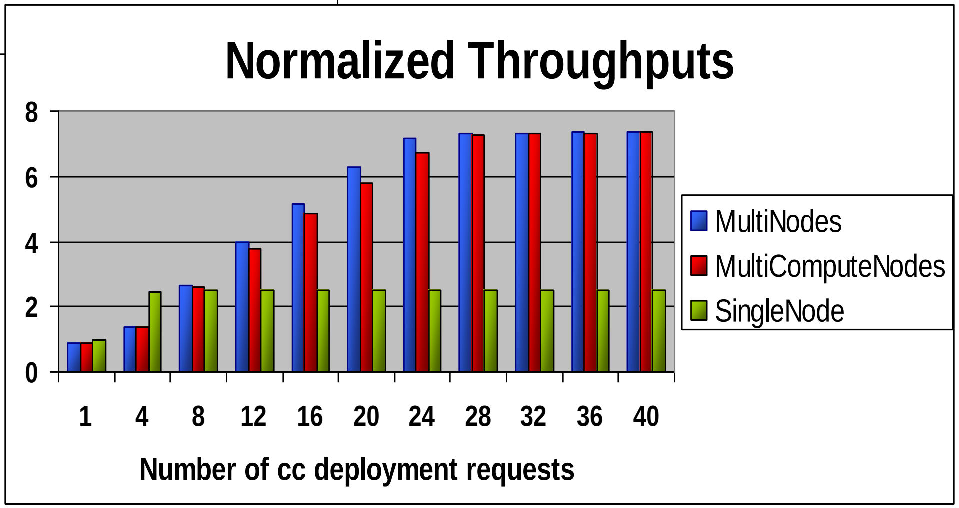
- **Scenario 1:**
  - CentOS VM with 2 Intel Xeon 5140 and 8 GB memory
  - cloud architecture „L“:
    - 100 managed-to nodes, 10 managed-from nodes, 22 switches, 1 disk array
  - 300 concurrent deployment requests per iteration

Metric	Value
events/sec	600,000*
total number of events for 1 iteration	1.6×10^9
Execution time for 1 iteration [sec]	3000
Memory consumption [GB]	2.6
CPU utilization [%]	99% of 1 core

* **approx. 1.2×10^6 events/sec on a W530 ThinkPad**

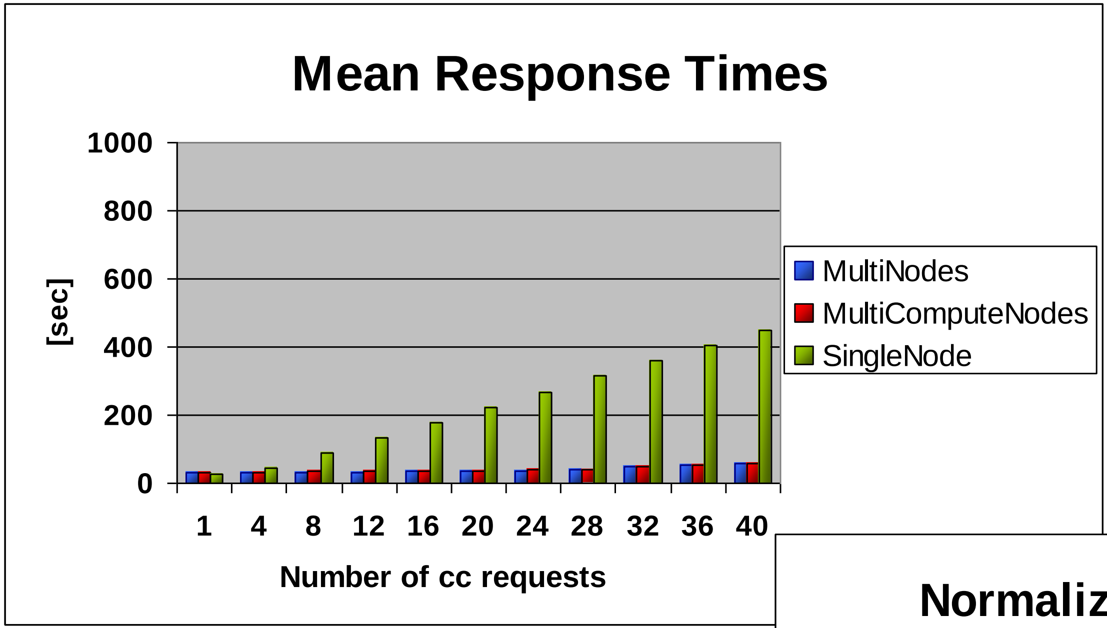
Simulation Results (1/7):

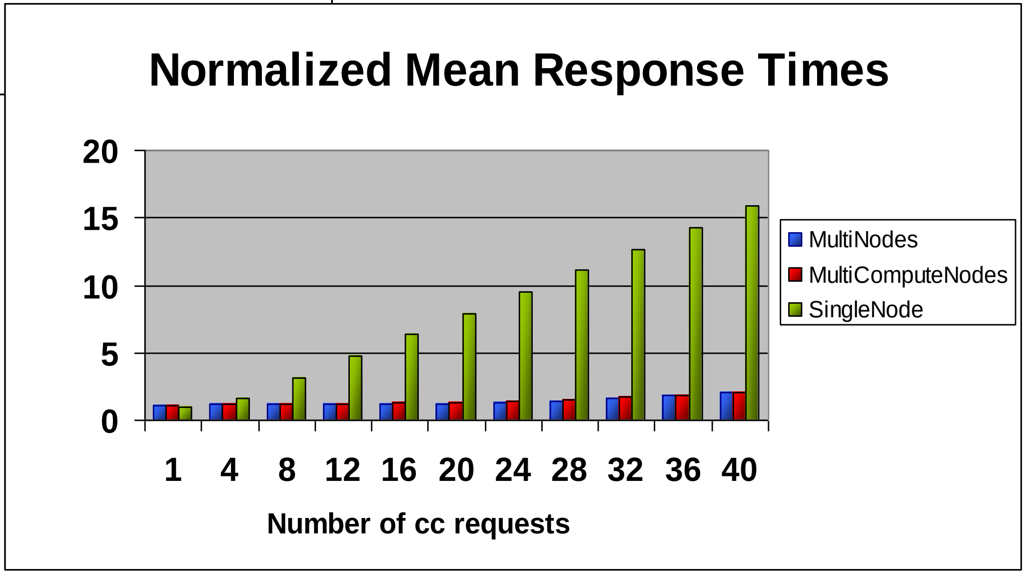




...performance increase when moving from 1 node to 100 managed-to nodes disappointing: where are the problems?

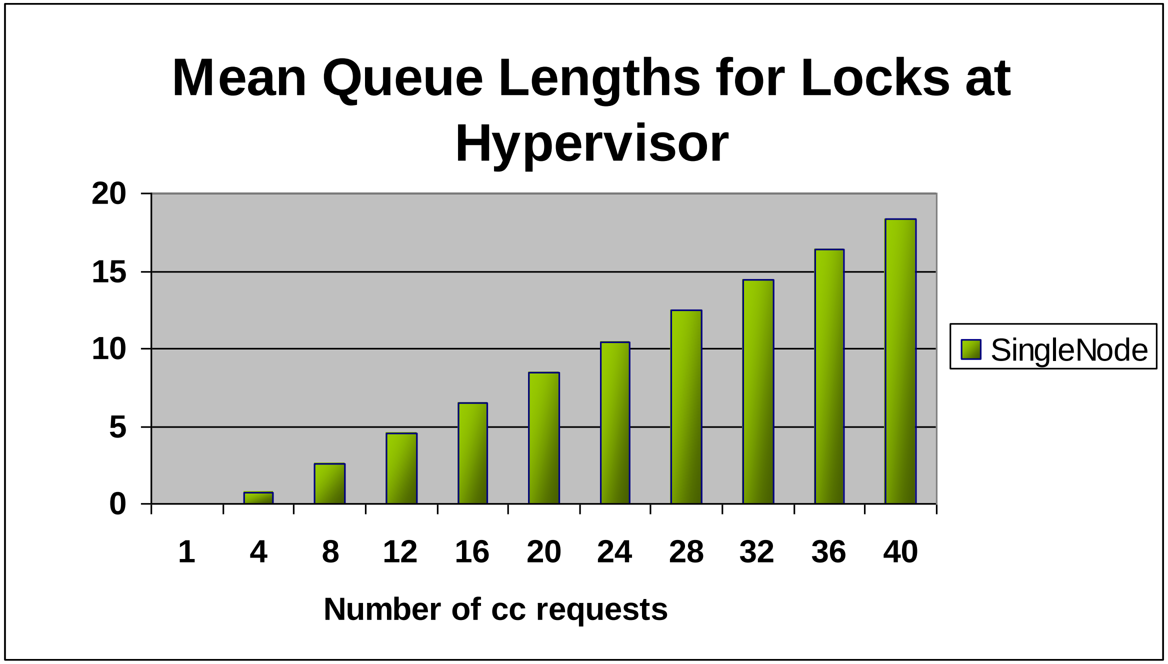
Simulation Results (2/7):





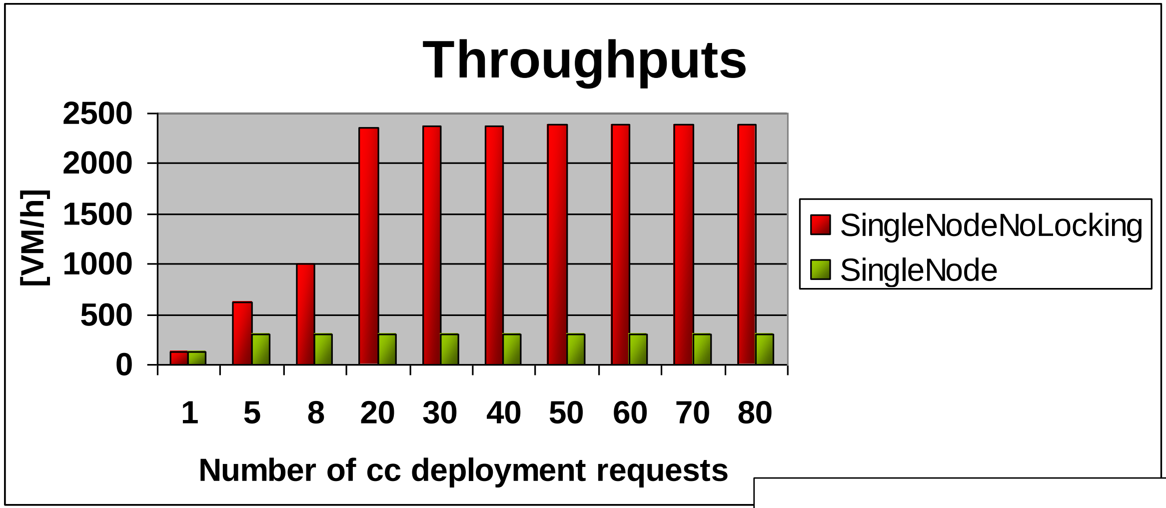
...response times pointing to the same problems

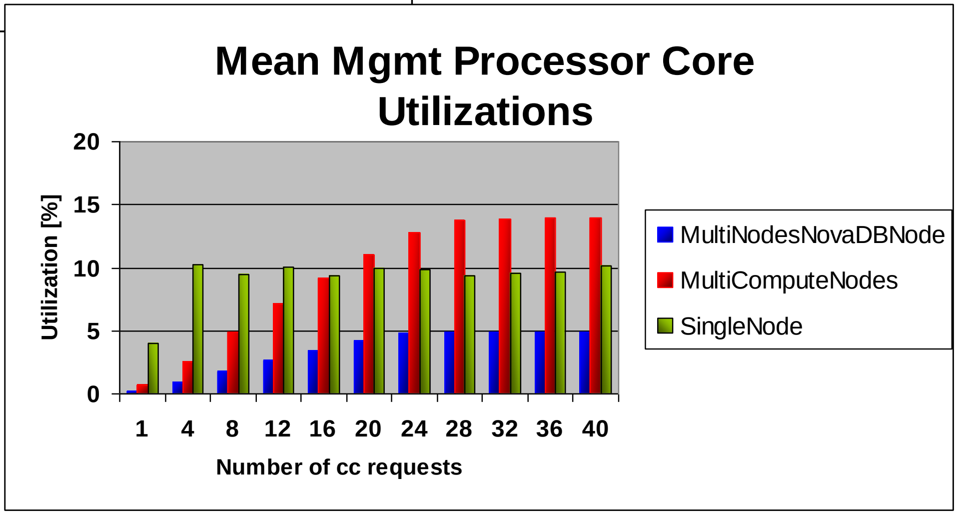
Simulation Results (3/7):



SingleNode scenario:
Contention for locks at nova-compute and hypervisor limits deployment scalability

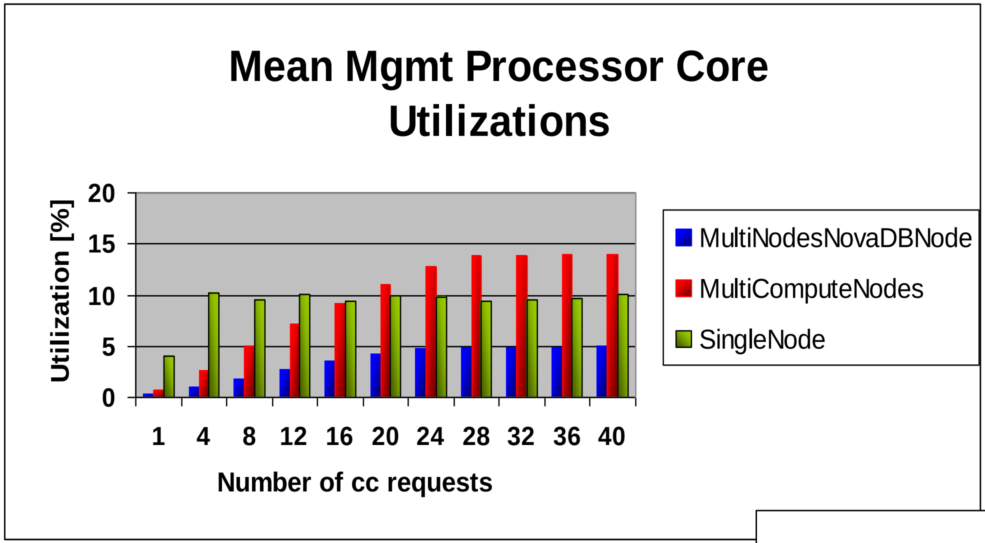
Simulation Results (4/7):



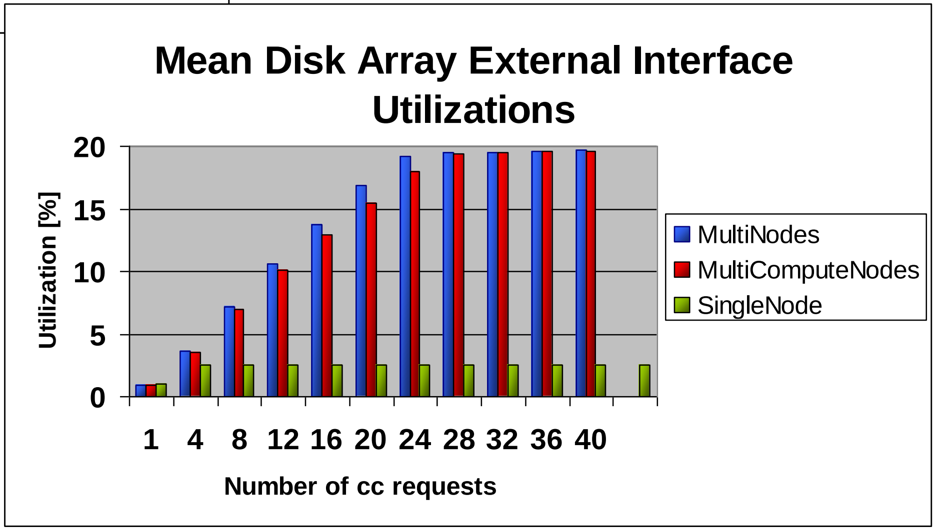


SingleNode scenario with locking turned off:
contention for CPU cycles limits deployment scalability

Simulation Results (5/7):



no issues with processor utilization



no issues with mean storage utilization, but high CoV...

Simulation Results (7/7):

- **Summary**
  - current OpenStack scalability limited by contention for
    - SW resources on single node
    - storage and network resources for multiple nodes

Contents:

- Cloud Performance & Scalabilty
- Cloud Performance Simulations
- The IBM Performance Simulation Framework for Cloud
- Performance Simulation of OpenStack Image Deployment
- **Summary**
- Outlook

Summary:

- **Pros:**
  - Cloud performance modeling & simulation has significant potential to evaluate impact of new SW heuristics or HW infrastructure, *especially at scale*
  - OMNeT based discrete-event simulation technology provides excellent basis here
- **Issues:**
  - Cloud SW is a very fast moving target
    → how can I adapt the simiulation model quick enough?
  - How to identify the artefacts that need to be included in a simulation model at scale?
  - How to obtain required workflow details and measurement results?
  - How to make cloud performance modeling & simulation accessable for non-experts?
    → Cloud Performance Simulation / Cloud Capacity Planning as a Service (CPSaS / CCPaS)?

Contents:

- Cloud Performance & Scalabilty
- Cloud Performance Simulations
- The IBM Performance Simulation Framework for Cloud
- Performance Simulation of OpenStack Image Deployment
- Summary
- **Outlook**

Outlook:

- **More R&D required to address the issues mentioned above**

- **Seamlessly integrate with**

  – cloud monitoring / logging tools
    → collect workflow infos and parameters

  – cloud resource orchestration / composition tools
    → ensure elasticity

Thank you!