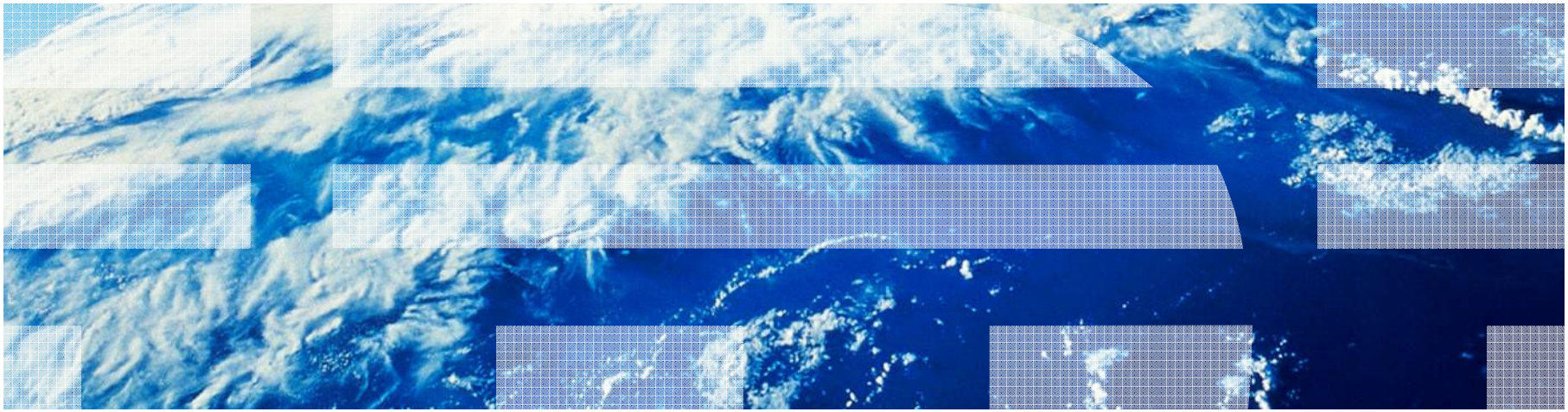
IBM

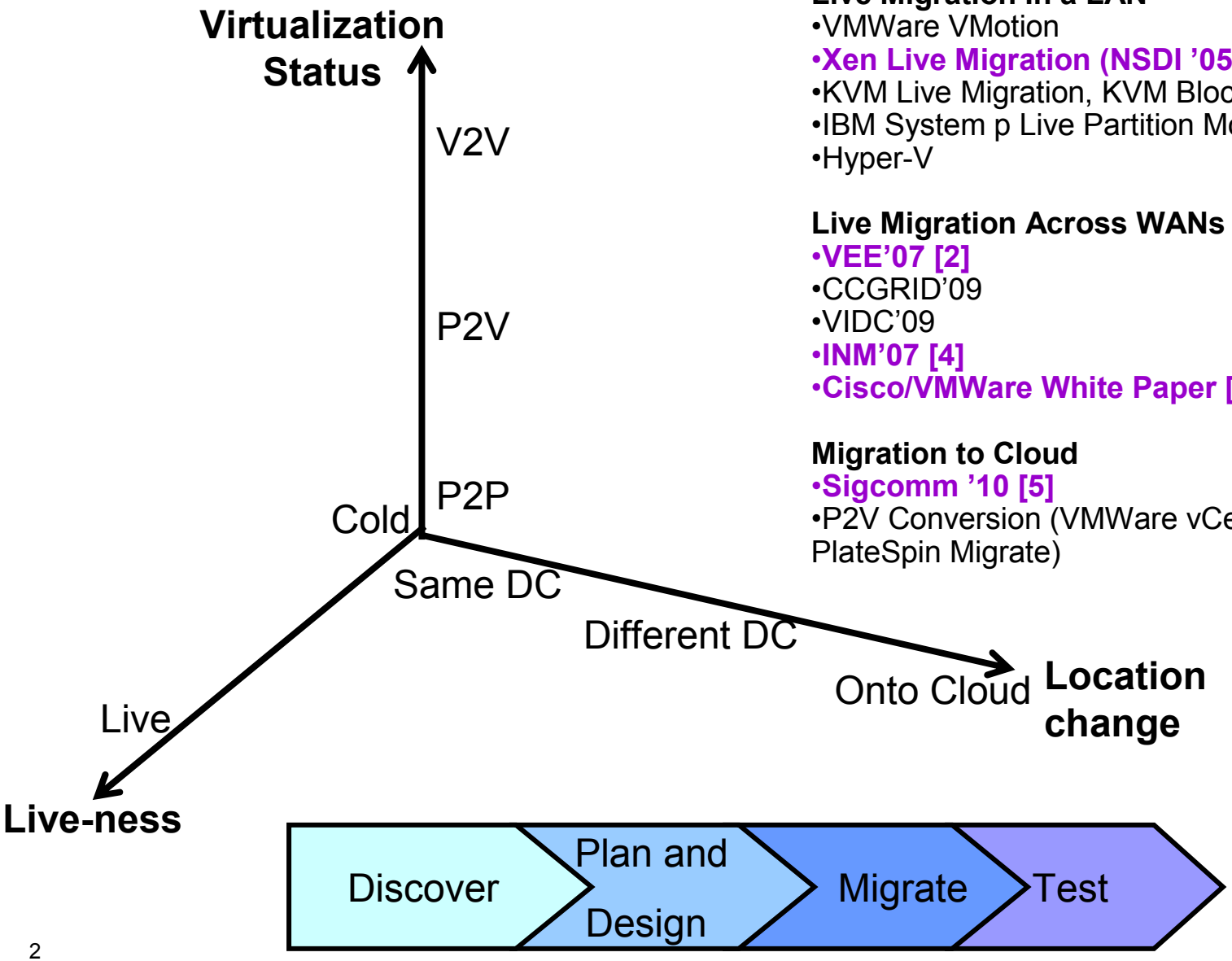
RESEARCH

Columbia University COMS W6998-6 Migration to Cloud

Kay Sripanidkulchai, IBM T.J. Watson Research Center
November 10, 2010



Migration Technologies and Process Steps



Live Migration In a LAN
- VMWare VMotion
- **Xen Live Migration (NSDI '05 [1])**
- KVM Live Migration, KVM Block Migration
- IBM System p Live Partition Mobility
- Hyper-V

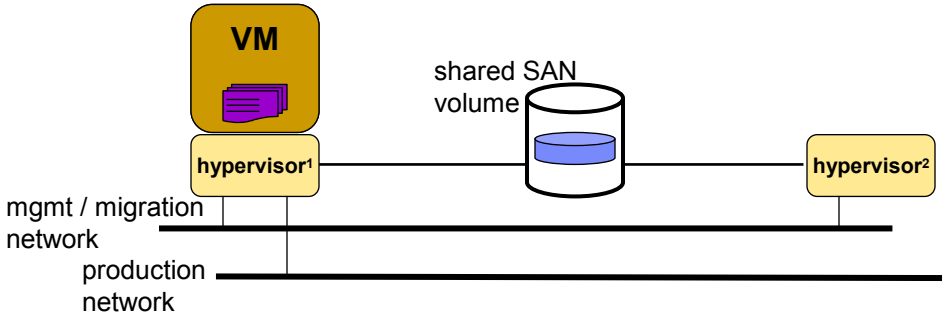
Live Migration Across WANs
- **VEE'07 [2]**
- CCGRID'09
- VIDC'09
- **INM'07 [4]**
- **Cisco/VMWare White Paper [3]**

Migration to Cloud
- **Sigcomm '10 [5]**
- P2V Conversion (VMWare vCenter Converter, PlateSpin Migrate)

Recap of Live Migration

- Demo

Migrate memory, register, and configuration files of a VM from one hypervisor to another hypervisor while the VM is running.

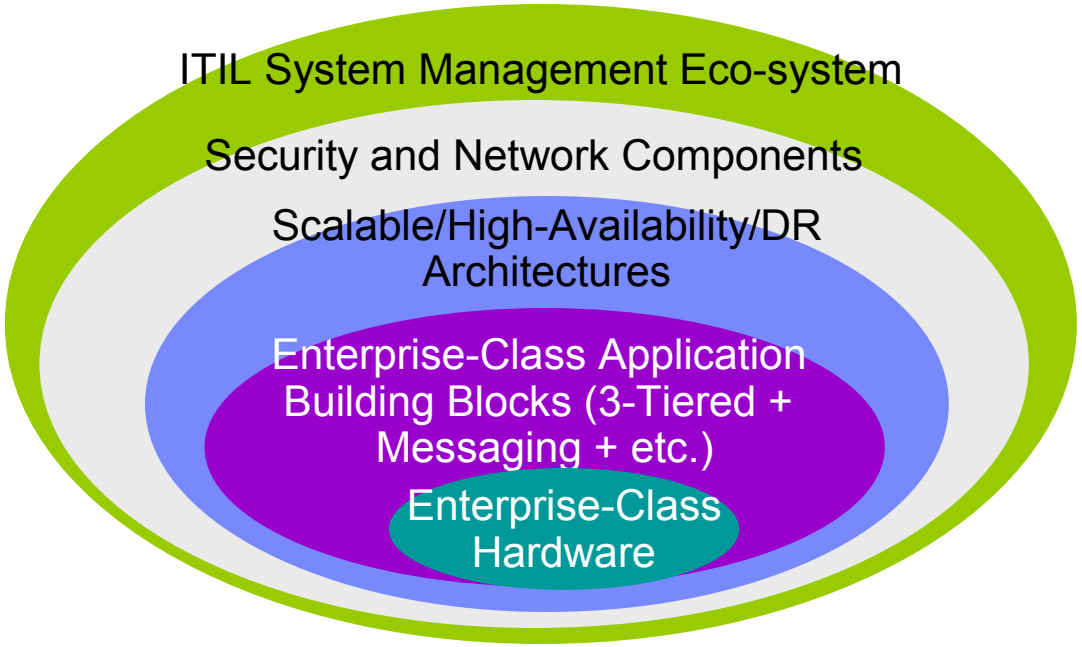


Outline “Migration to Cloud”

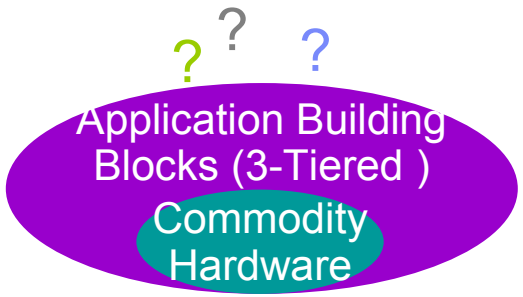
- Planning migrations, focusing on performance and SLA requirements
  - What to migrate?
  - Which cloud?
- Executing migrations
  - P2V conversions
  - Migrating to EC2

Enterprise vs. individual customers have different requirements [LADIS 09]

Typical Enterprise Application Architecture

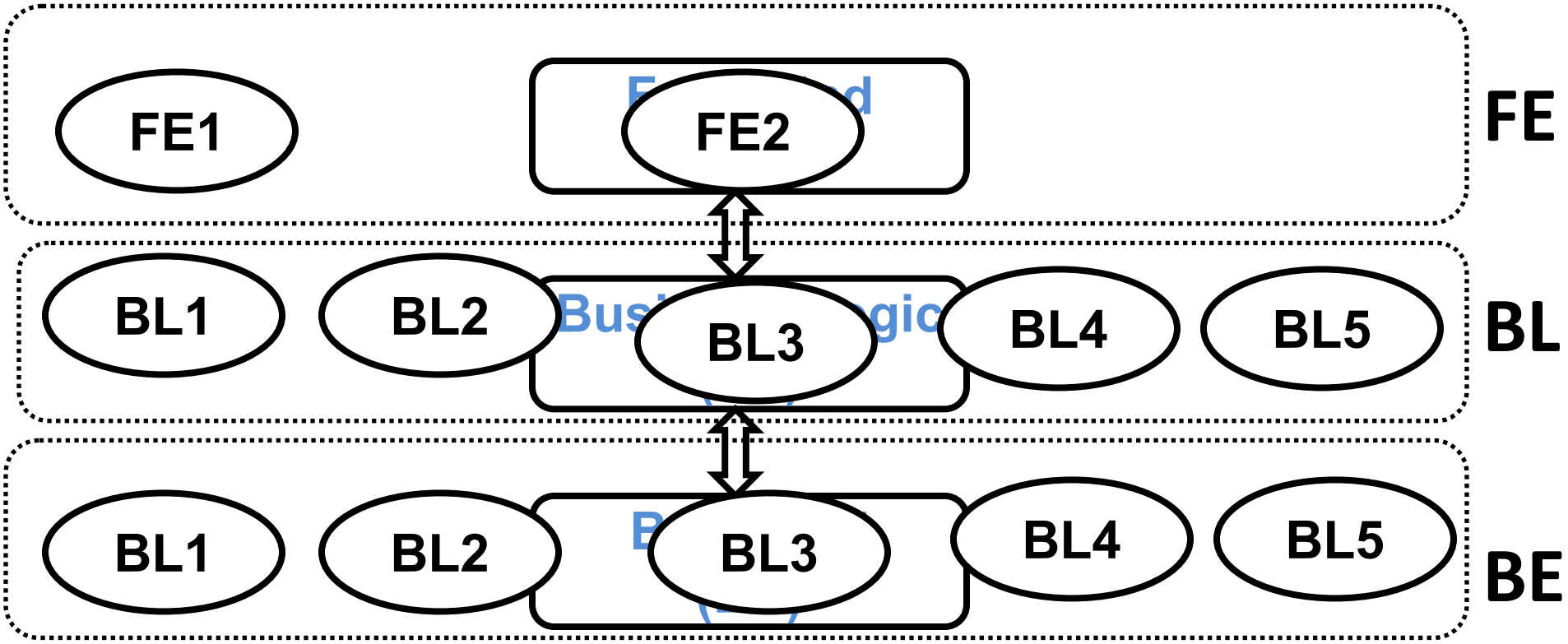


Typical Small/Individual Application Architecture



Enterprise Applications

E.g., Payroll, travel and expense reimbursement, customer relationship management etc.

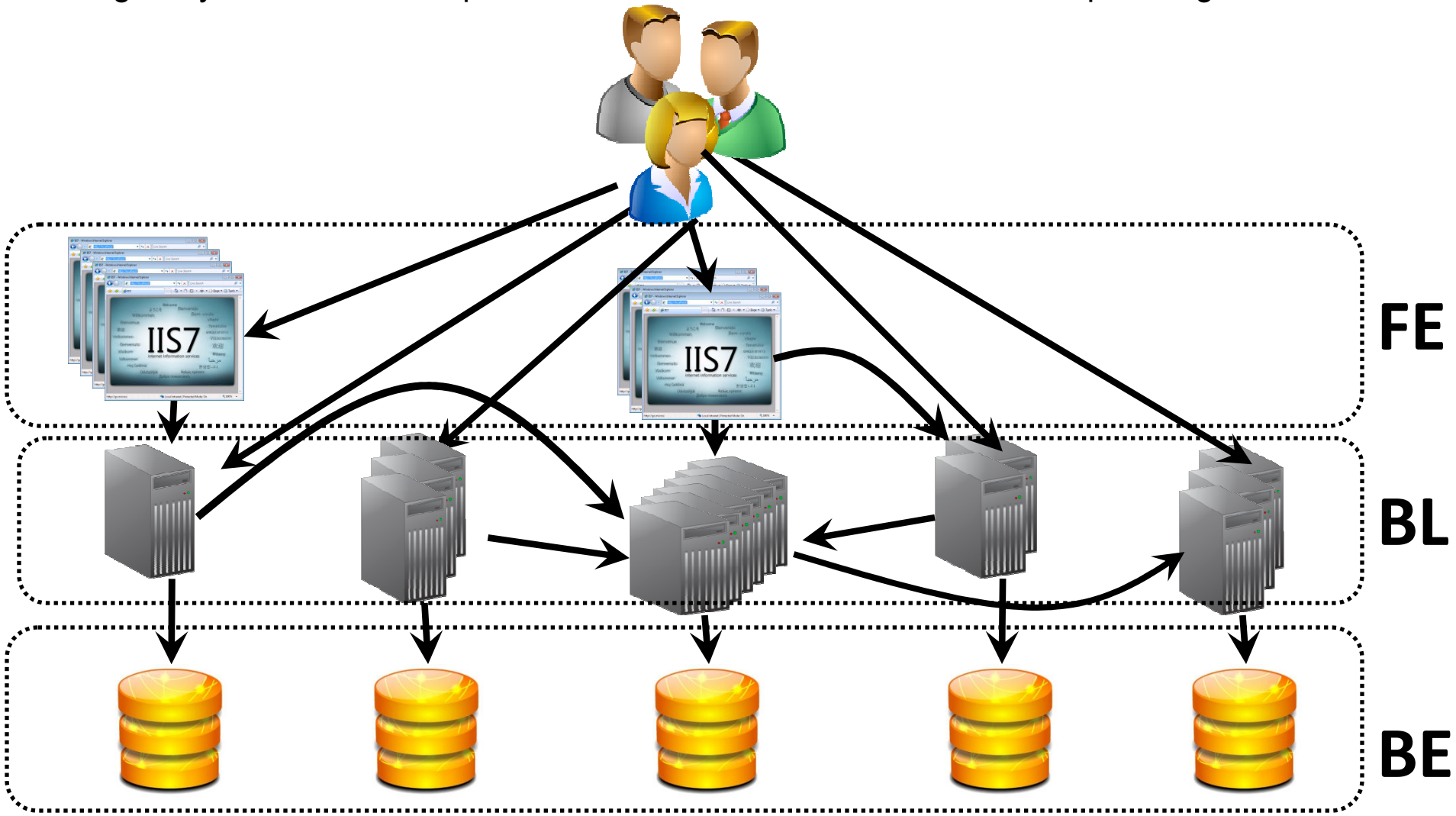


3-tier Application Structure

6

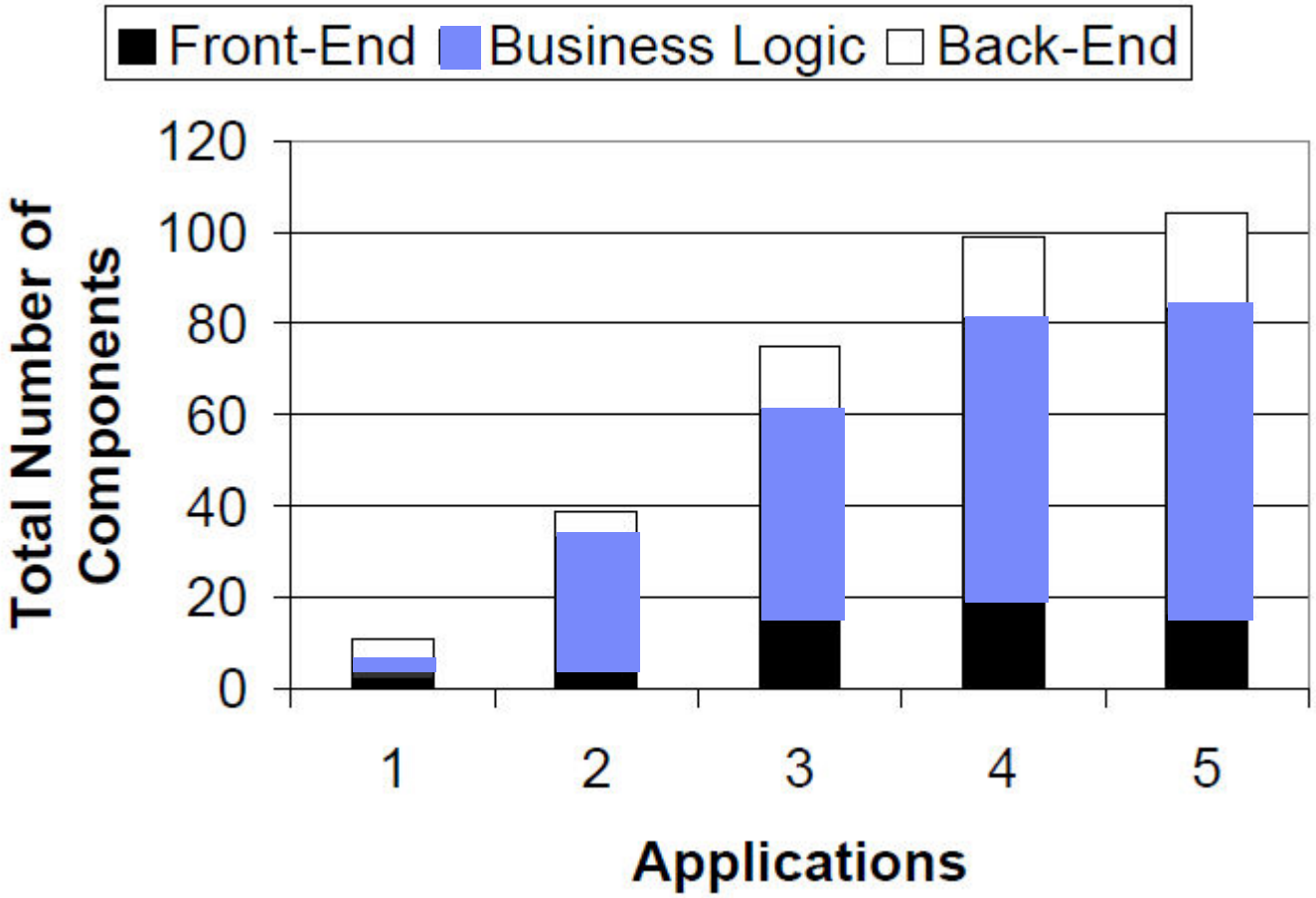
Enterprise Applications

E.g., Payroll, travel and expense reimbursement, customer relationship management etc.



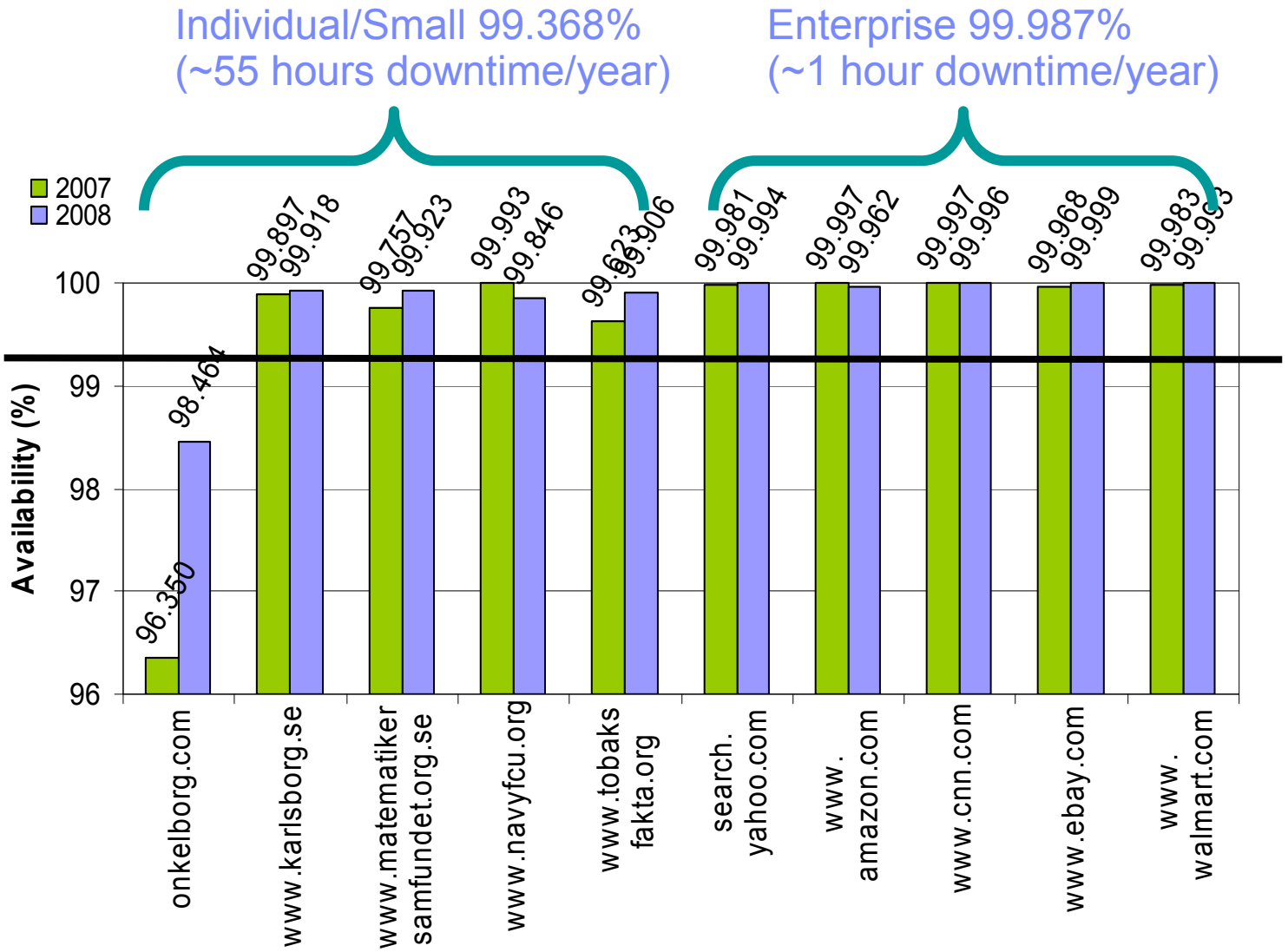
7

Planning migrations to the cloud [Sigcomm'10]



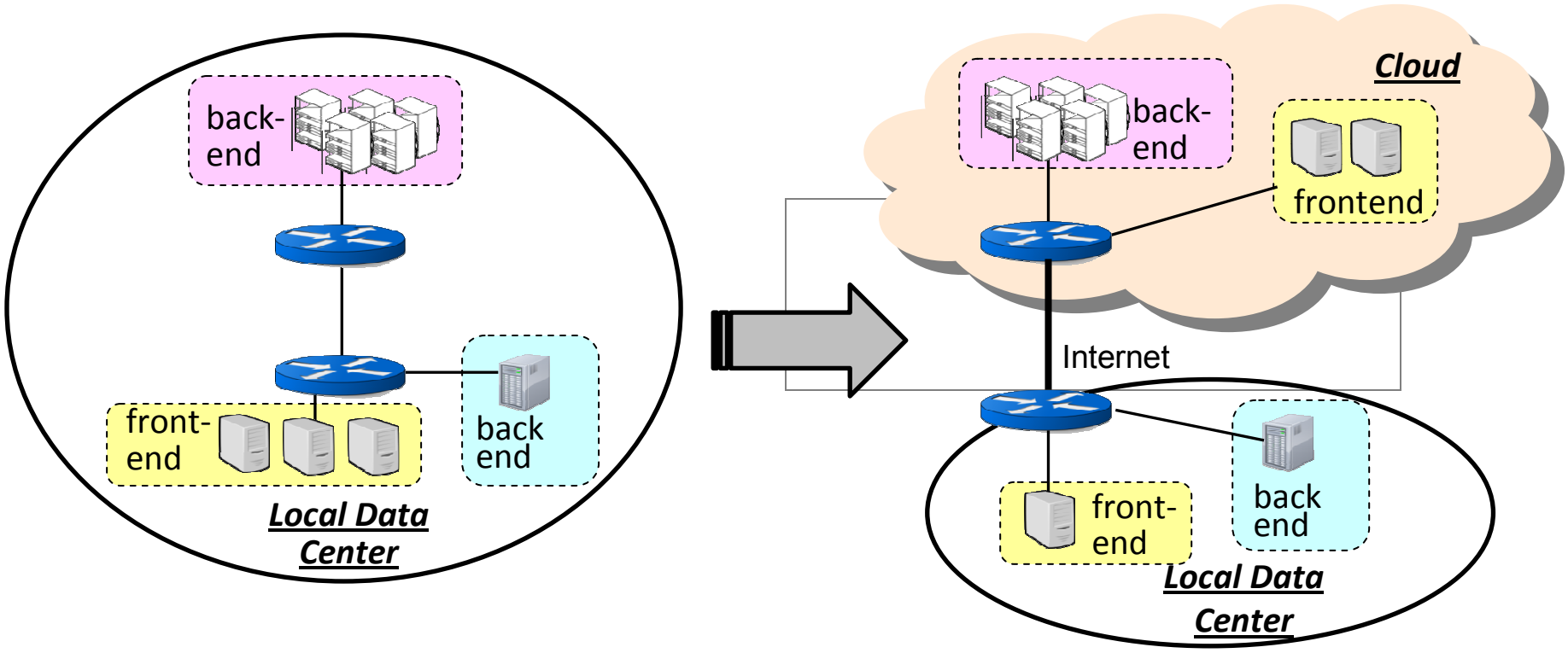
There are gaps in service availability requirements for enterprise users [LADIS 09]

State-of-the-art cloud SLA at 99.95% or ~4 hours downtime/ year.

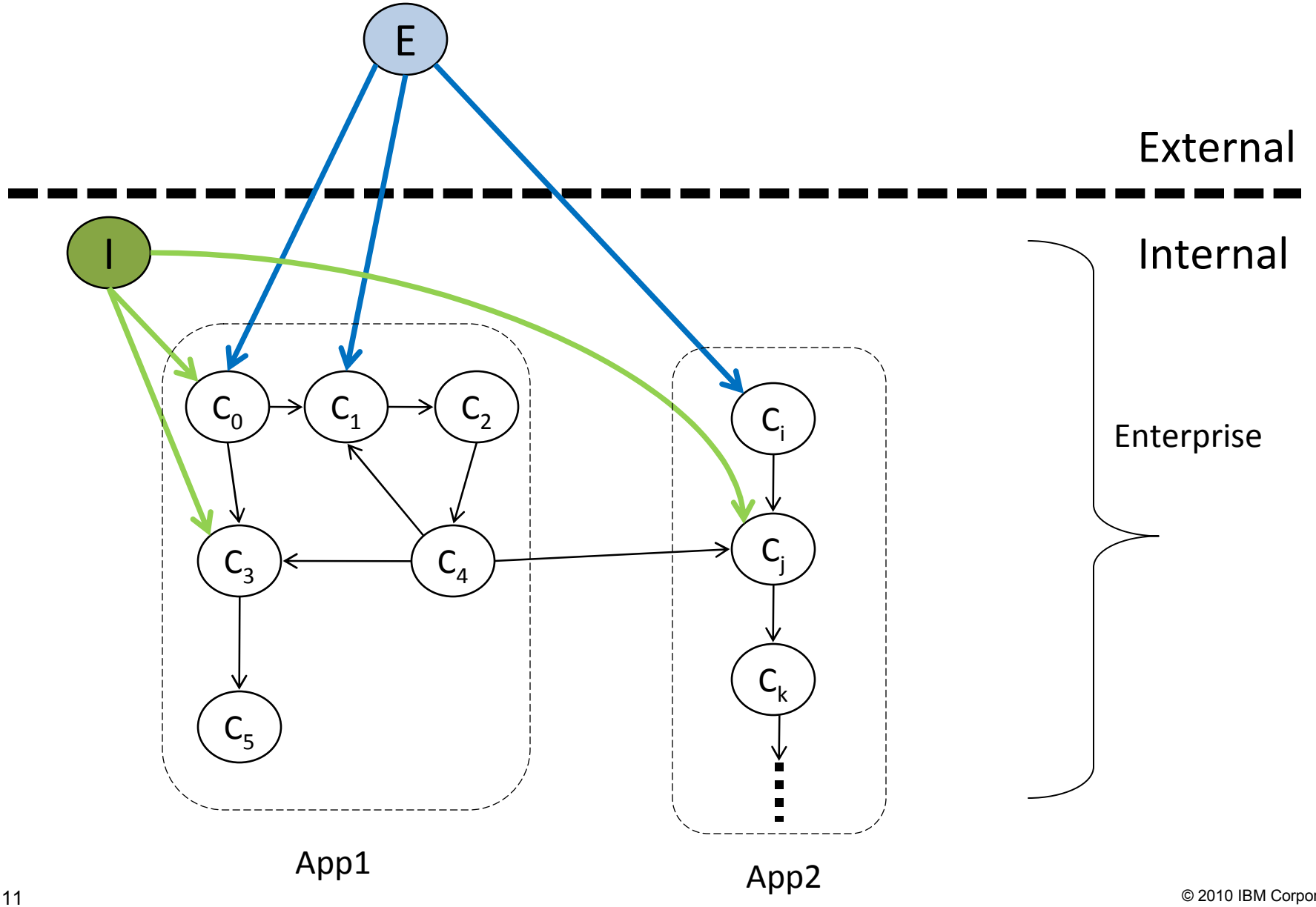


Our focus #1 : Planning hybrid cloud layouts

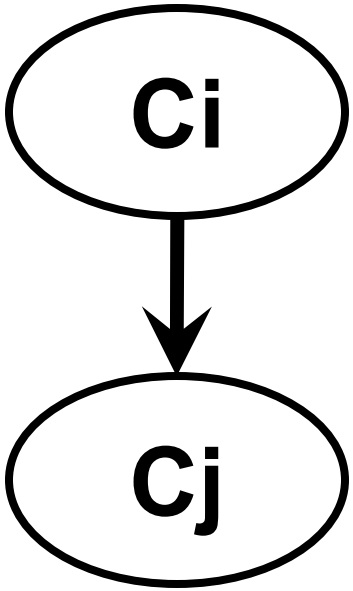
- Cost savings, Application response times, Bandwidth costs
- Scale and complexity of enterprises applications



Abstracting the planning problem



Abstracting the planning problem



$\mathbf{N_i}$ = number of servers in component C_i

$\mathbf{T_{ij}}$= number of transactions per second along (i,j)

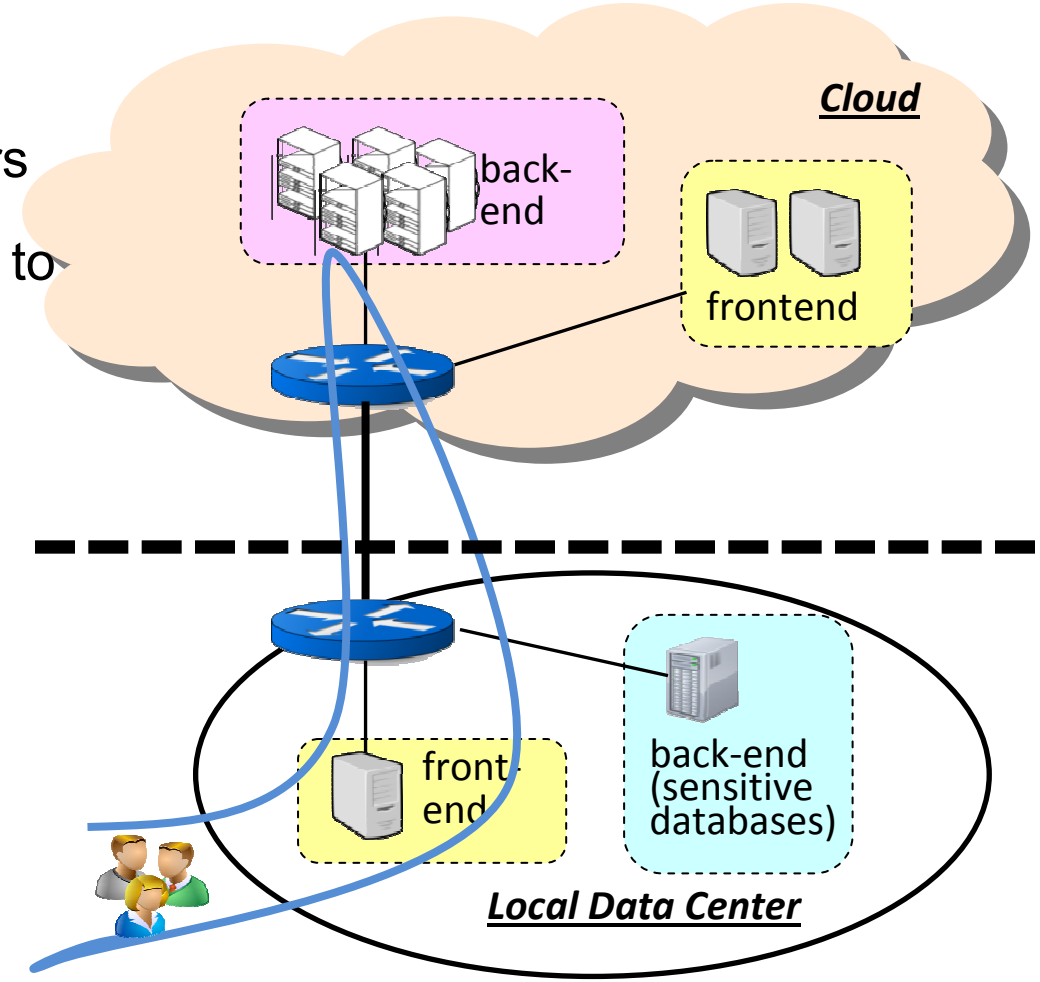
$\mathbf{S_{ij}}$= average size of transactions along (i,j)

To determine:

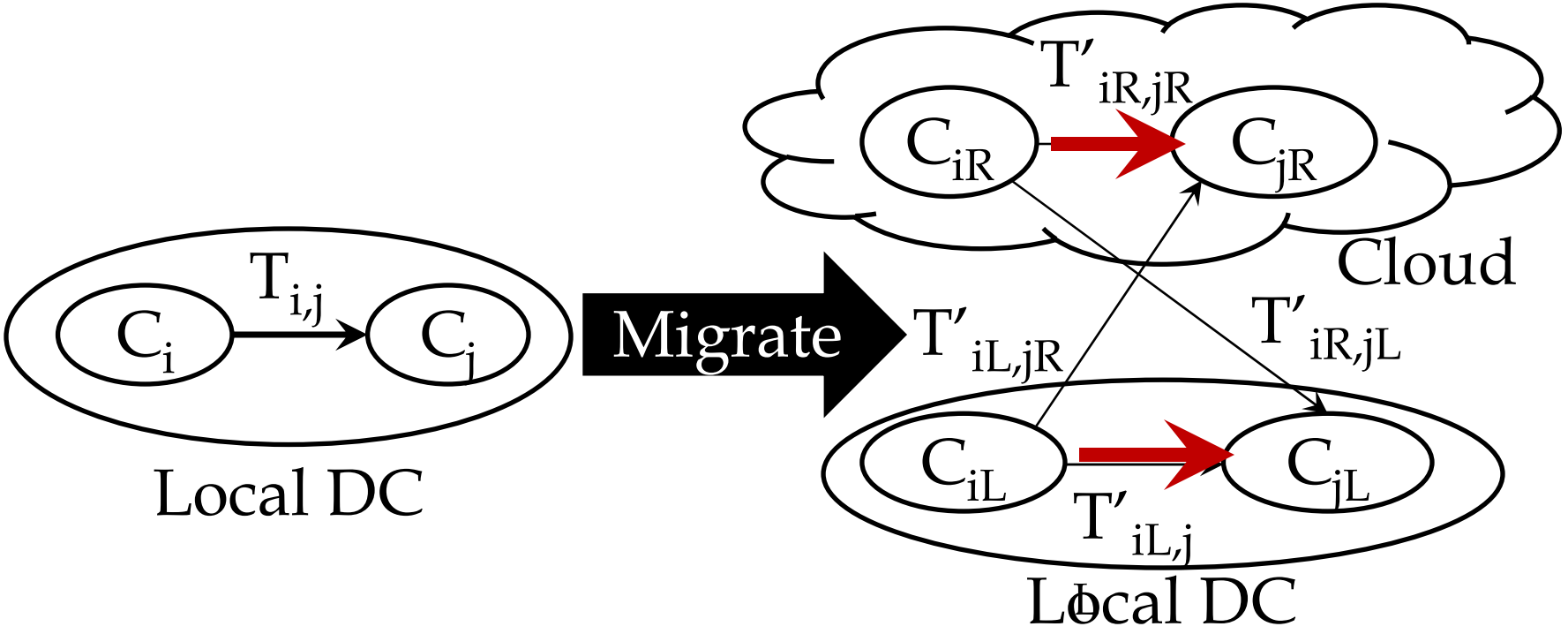
$\mathbf{m_i}$= number of servers of component C_i to migrate to the cloud ($m_i \leq N_i$)

Formulating the planning problem

- **Objective:** Maximize cost savings on migration
  - Benefits due to hosting servers in the cloud
  - Cost increase/savings related to wide area Internet communication
- **Constraints:**
  - Policy constraints
  - Bounds on increase in transaction delay
- **Future work:**
  - Application availability



Partitioning requests after migration



(1) Location sensitive routing

(2) Location Independent routing

- Split in proportion to the number of servers in C_{jL} and C_{jR}
- Introduces non-linearity in constraints.

Modeling Approach

Model complexity Vs. Practicality of data collection

Fine-grained models:

- Potentially more accurate
- Model parameters harder to collect

Our Approach:

- Use easily available information (e.g., computation times of components and communication times on links)
- Empirical experience to drive iterative model refinements

Modeling user response times

- Ideally, desirable to bound increase in:
  - Mean response time
  - Response time variations (e.g., 95%ile response times).
- Bounding changes to mean delay relatively easier
  - Linearity of expectations
- Bounding delay variations harder
  - E.g., need distribution of component service times
  - Feasible to bound changes to variance of response times
    - By conditioning on path taken by transactions
    - Assuming independence of individual component response times etc.
    - Can be extended to applications with non path-like transactions
  - Conservative bounds on changes to delay percentiles feasible

Benefits/costs on migration

- Benefits due to hosting servers in the cloud
  - Economies of scale, lowered operational expenses
  - Benefit estimates from Armbrust et al (Berkeley TR, 2009)
  - Benefits dependent on compute or storage servers
- Costs related to Internet communication
  - Linear cost model
  - Matches charging model of EC2, Azure etc.
- Future Extensions:
  - One-time costs of executing migrations
  - Savings due to not provisioning enterprises for peaks

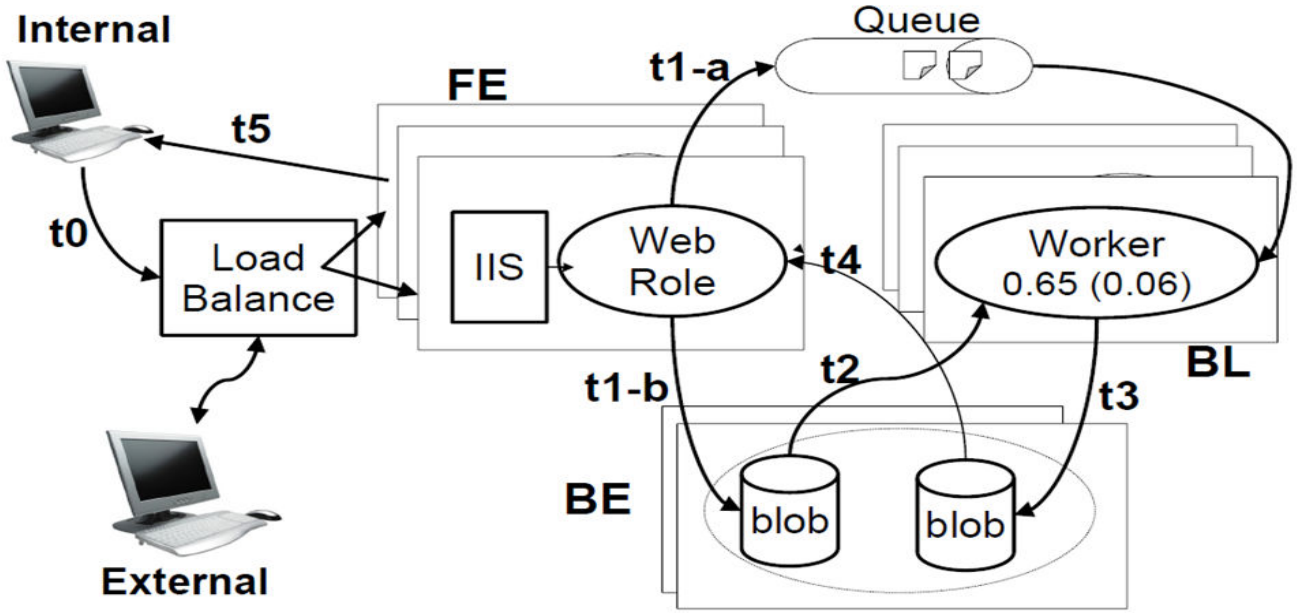
Evaluation Goals and Case Studies

- Evaluation Goals:
  - Are there scenarios where a hybrid approach makes sense?
  - What are the cost savings associated with going to the cloud?
  - How effective are coarse-grained planning models?

- Case Studies:
  - Windows Azure SDK application
  - Campus Enterprise Resource Planning (ERP) application

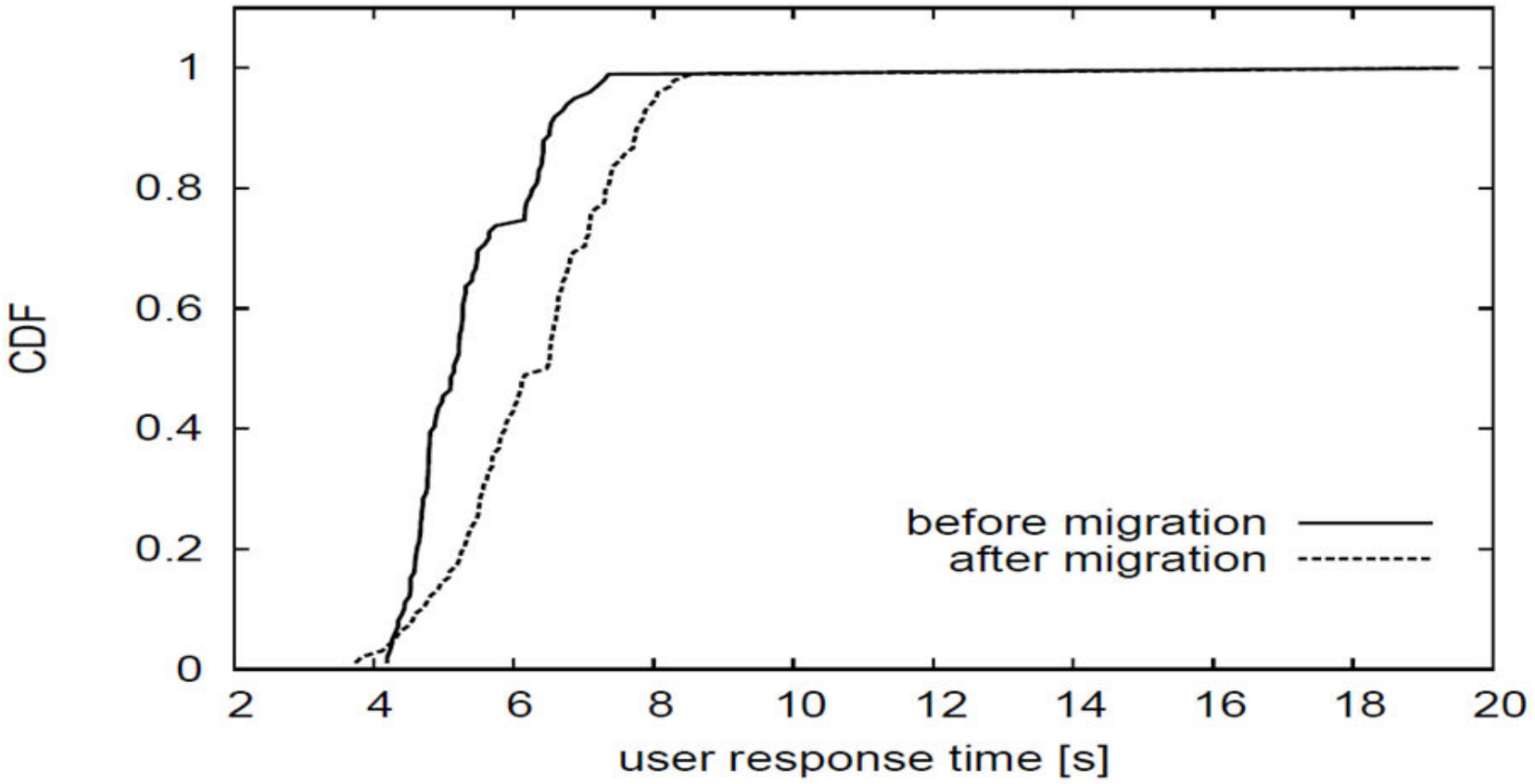
Experiments on cloud test-bed

- Thumbnail example application
- Two Azure data centers (DCs), represent local/remote
- Internal users: hosts in campus close to internal DC
- External users: Planetlab
- Reengineer application for hybrid cloud deployment



Results

- Plan requirements: increase in mean delay less than 10%, increase in variance less than 50%
- Algorithm Recommendation: Migrate 1 FE , 3 BL servers
- Observed: 17% increase in mean, 12% increase in variance



Conclusions [SIGCOMM 10]

- Hybrid cloud models often make sense
  - Enable cost savings, while meeting enterprise policies and application response time requirements
- Planned approach to migration important and feasible
  - Algorithms for hybrid cloud layouts
  - Algorithms for correct reconfiguration of security policies
- Future Work
  - Exploring model complexity and performance inaccuracy
  - Wider range of application case studies
  - Take workload and network dynamics into account

Outline “Migration to Cloud”

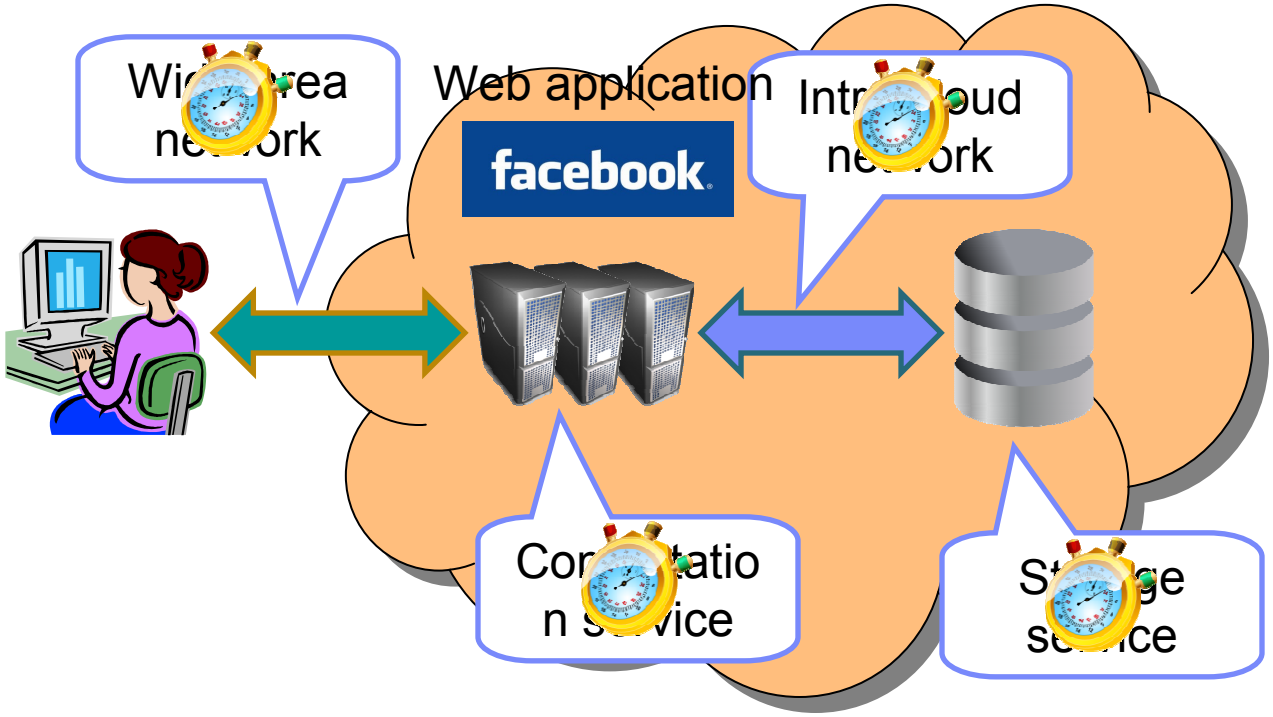
- Planning migrations, focusing on performance and SLA requirements
  - What to migrate?
  - Which cloud?
- Executing migrations
  - P2V conversions
  - Migrating to EC2

Which cloud provider is best suited for my application? [HotCloud 10]

- Reason #1: clouds have different **service models**
  - Infrastructure-as-a-Service
  - Platform-as-a-Service
  - A mixture of both
- Reason #2: clouds offer different **charging schemes**
  - Pay per instance-hour
  - Pay per CPU cycle
- Reason #3: applications have different characteristics
  - Storage intensive
  - Computation intensive
  - Network latency sensitive
- Reason #4: high overhead to port application to clouds
  - Different and incompatible APIs
  - Configuration and data migration

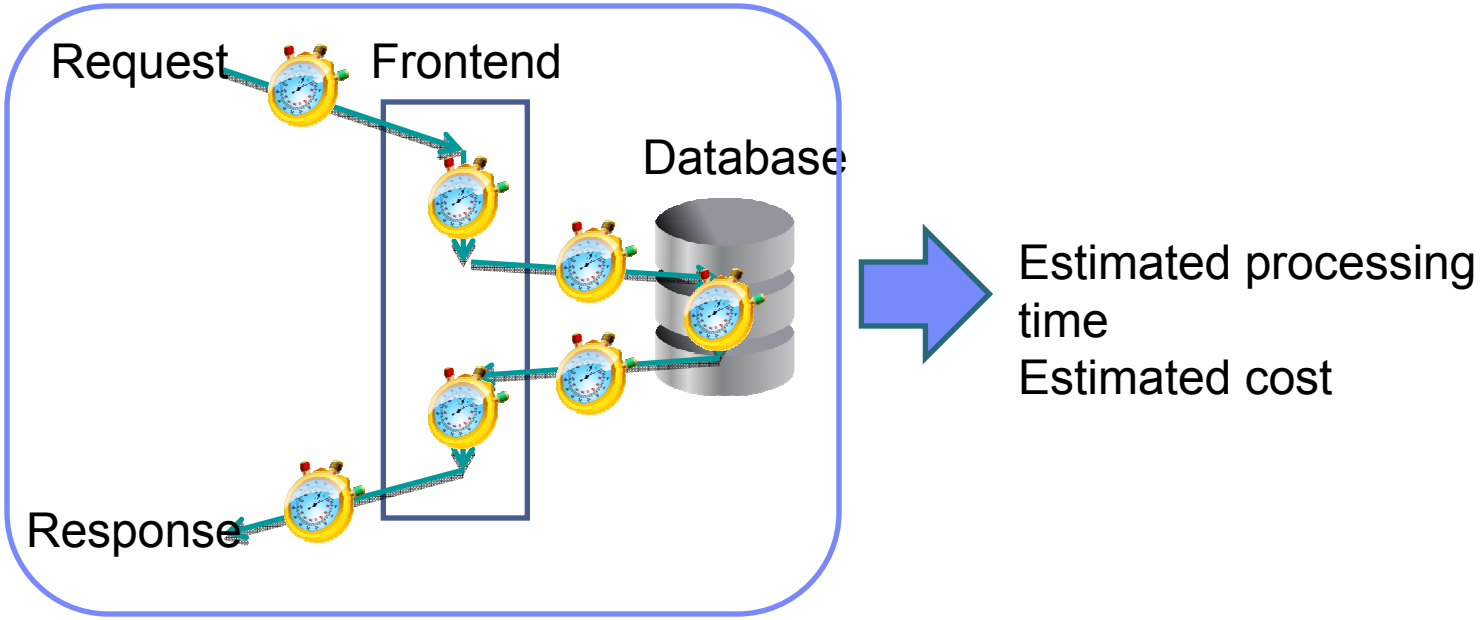
How does CloudCmp work?

- Step 1: identify the common cloud services
- Step 2: benchmark the services



How does CloudCmp work?

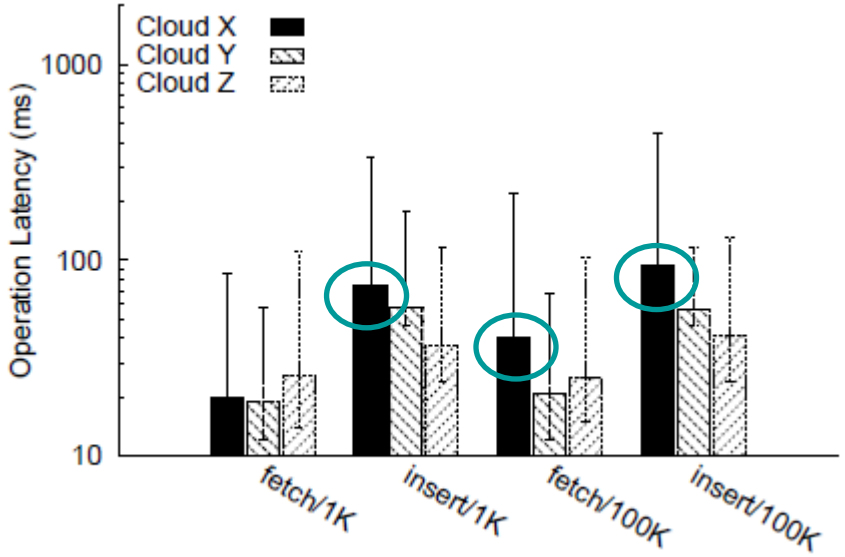
- Step 3: capture realistic application workload
  - Extract the execution path of each request
- Step 4: estimate the performance and costs
  - Combine benchmarking results and workload information



Challenges

- How to design the benchmarking tasks?
  - Fair and representative
- How to accurately capture the execution path of a request?
  - An execution path can be complex, across multiple machines
- How to estimate the overall processing time of an application
  - Applications can be multi-threaded

Results: storage

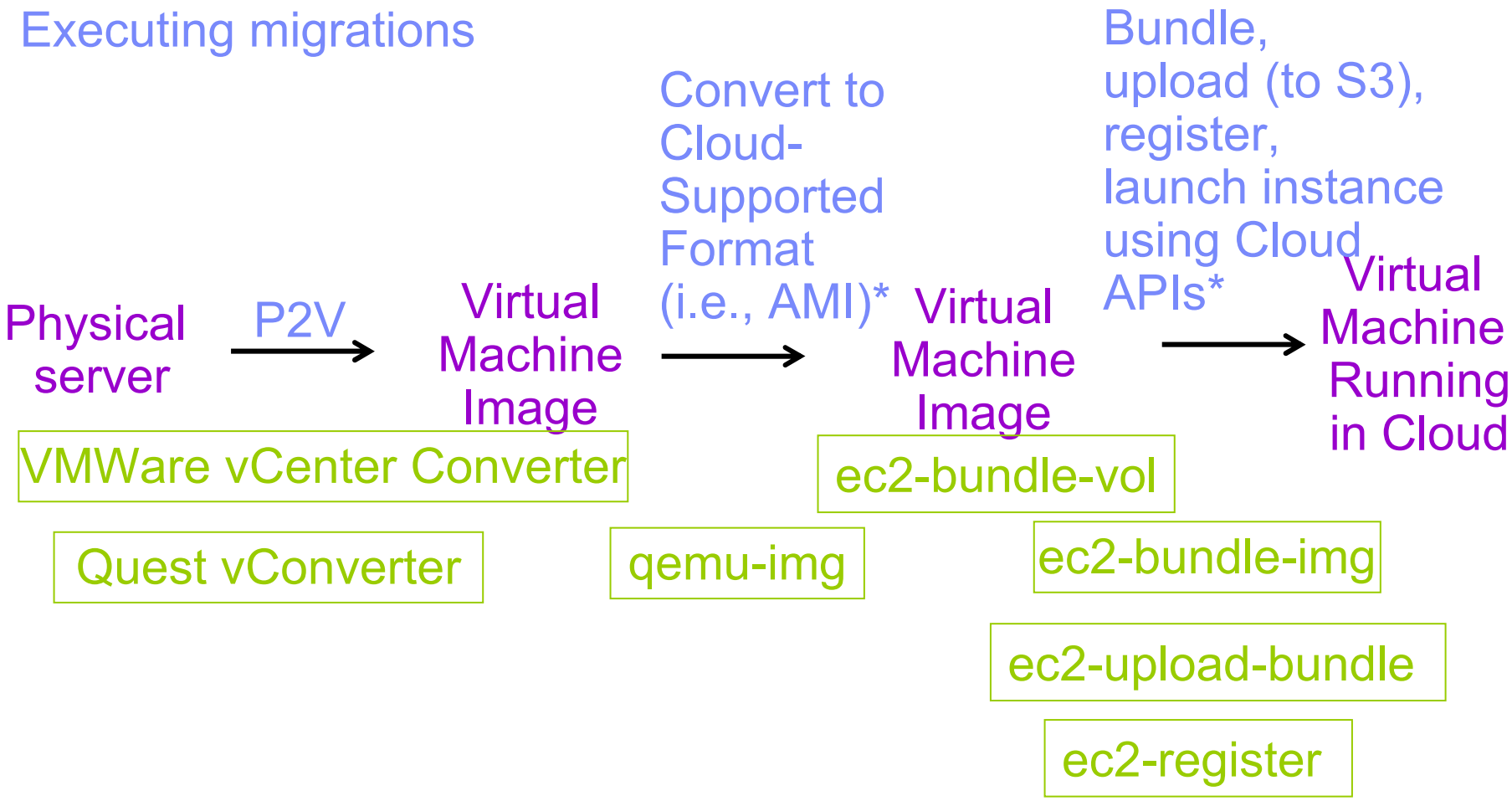


- Despite X's good performance in computation, its storage service can be slower than the others
- A cloud may not ace all services

Outline “Migration to Cloud”

- Planning migrations, focusing on performance and SLA requirements
  - What to migrate?
  - Which cloud?
- Executing migrations
  - P2V conversions
  - Migrating to EC2

Executing migrations



* http://thewebfellas.com/blog/2008/9/1/creating-an-new-ec2-ami-from-within-vmware-or-from-vmdk-files

Reference Material

1. Mohammad Hajjat, Xin Sun, Yu-Wei Sung, Dave Maltz, Sanjay Rao, Kunwadee Sripanidkulchai and Mohit Tawarmalani. Cloudward Bound: Planning for Benefical Migration of Enterprise Applications to the Cloud, Sigcomm 2010.
2. Ang Li, Xiaowei Yang, Srikanth Kandula and Ming Zhang. CloudCmp: Shopping for a Cloud Made Easy. HotCloud 2010.
3. Timothy Wood, Prashant Shenoy, Arun Venkataramani, and Mazin Yousif. Black-box and Gray-box Strategies for Virtual Machine Migration. NSDI 2007.
4. Kunwadee Sripanidkulchai, Sambit Sahu, Yaoping Ruan, Anees Shaikh, and Chitra Dorai, Are Clouds Ready for Large Distributed Applications?, LADIS 2009.

Migration Project Ideas

- Planning live migration within the LAN
  - Algorithms for when to migrate, what to migrate, where to migrate
    - VMWare: Build 2 ESXi hypervisors, run vSphere Enterprise (or above), understand how DRS works, design algorithm to automate live migration, emulate resource contention to trigger migration, and evaluate algorithm
    - KVM or Xen: Improve KVM or Xen's management capabilities to automate live migration by implementing capabilities similar to VMWare's DRS in libvirt
    - Look at reference [3] for examples of algorithms for inspiration

- Migration to cloud
  - Fast migration of instances from local data center to EC2
    - Build new migration capabilities to migrate virtual machines from your local data center (in whichever image format you like – VMWare, Xen, etc.) to EC2. Look at how to use S3 and image conversion technologies for ami. See if you can optimize migration performance using caching, deduplication, etc.