UCSDCSE
Computer Science and Engineering

DieCast: Testing Distributed Systems with an Accurate Scale Model

Diwaker Gupta

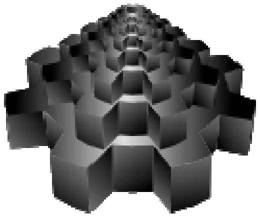
Kashi V. Vishwanath

Amin Vahdat

University of California, San Diego

UCSD

Alice



High performance filesystem

Diverse deployment environments

Limited testing infrastructure

Use smaller infrastructure to test a much larger system

Goals

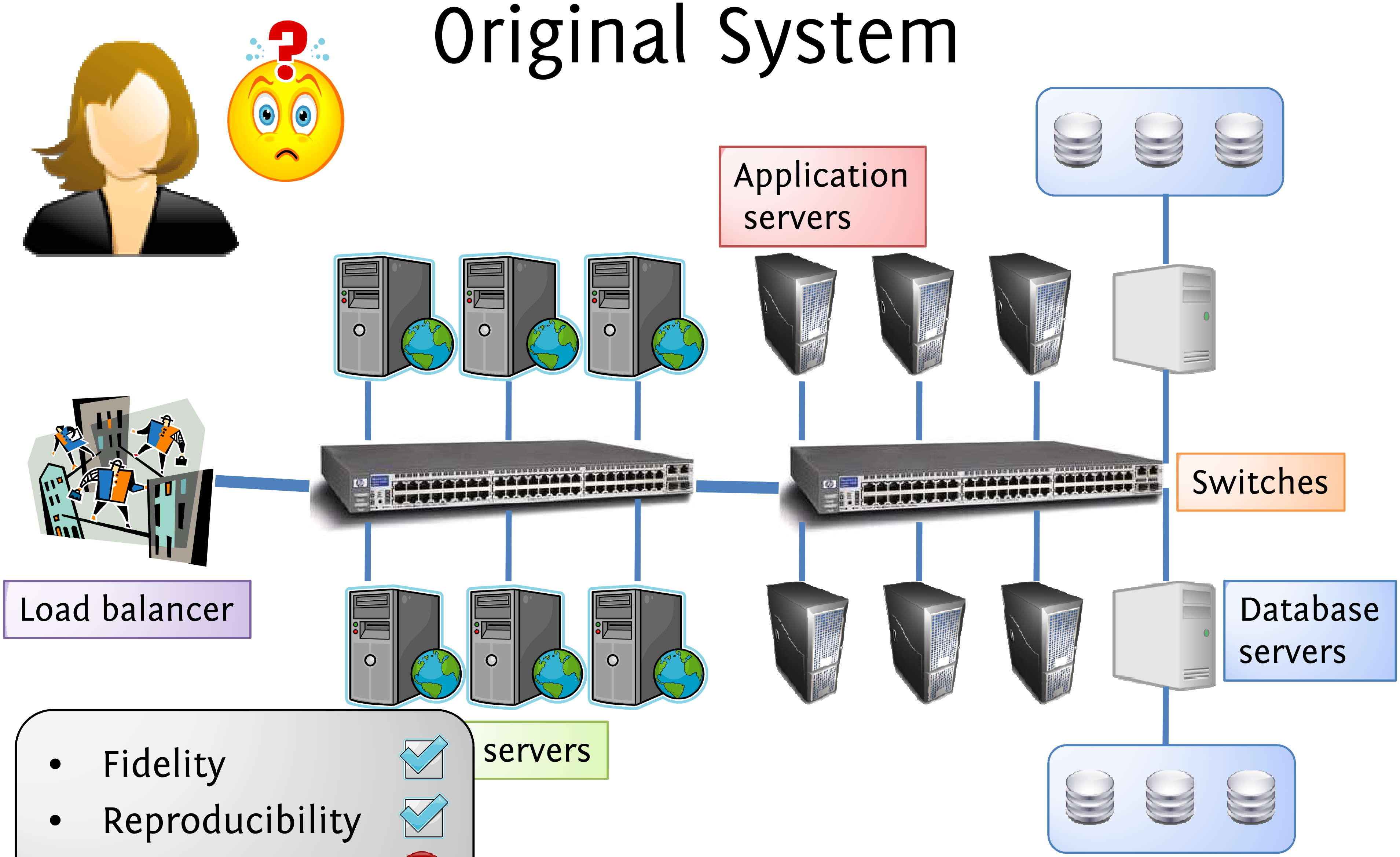
- Fidelity
  - How closely can we replicate the target system?
- Reproducibility
  - Can we do controlled experiments?
- Efficiency
  - Use fewer resources

DieCast can scale up a test infrastructure by an order of magnitude

DieCast Overview

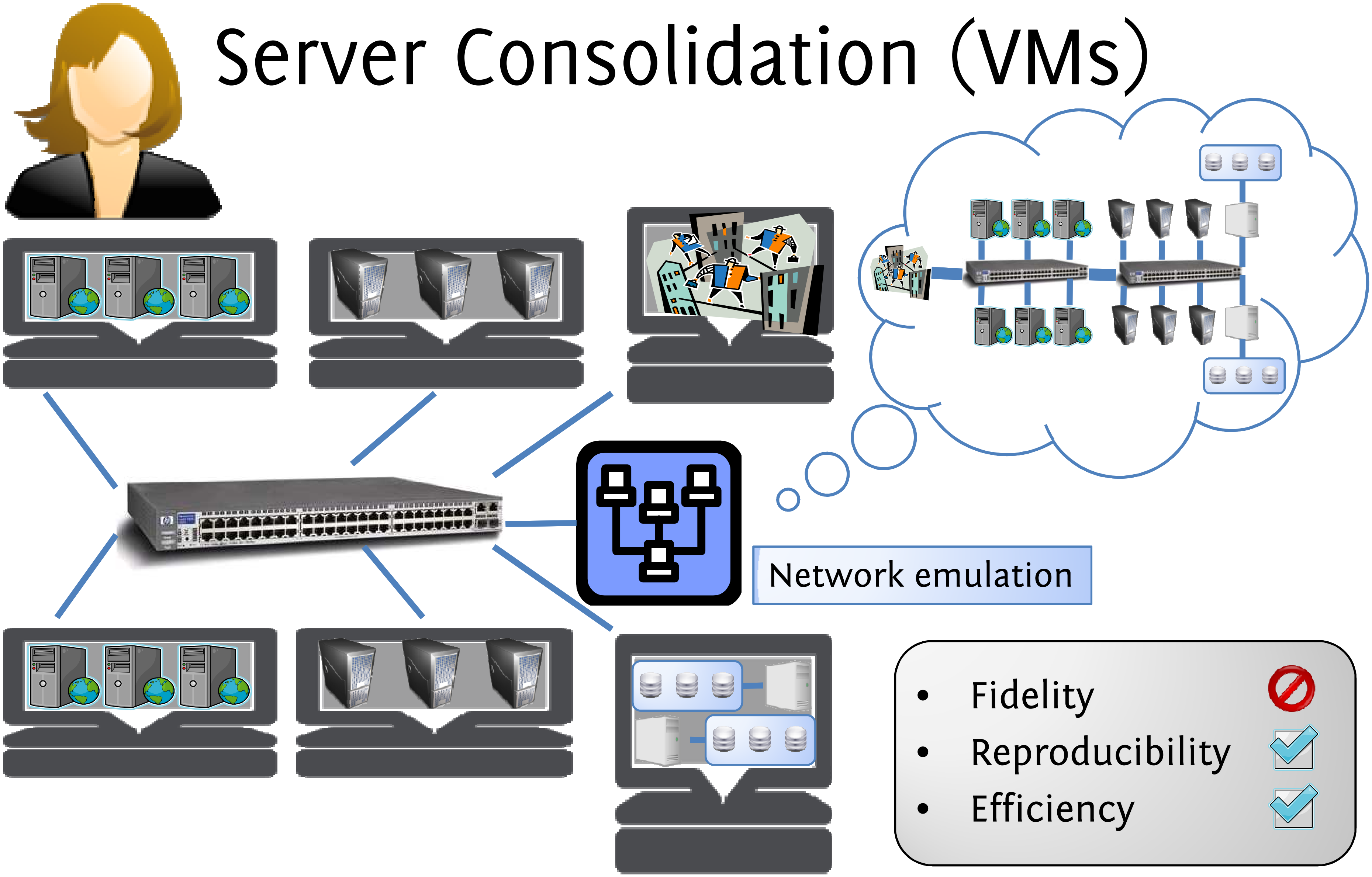
- ✓ Replicate target system using fewer machines
- ✓ Resource equivalence: *perceived* CPU capacity, disk and network characteristics
- ✓ Preserve application performance
- ✗ Not scaled
  - ✗ Physical memory: mitigating solutions
  - ✗ Secondary storage: cheap

Original System



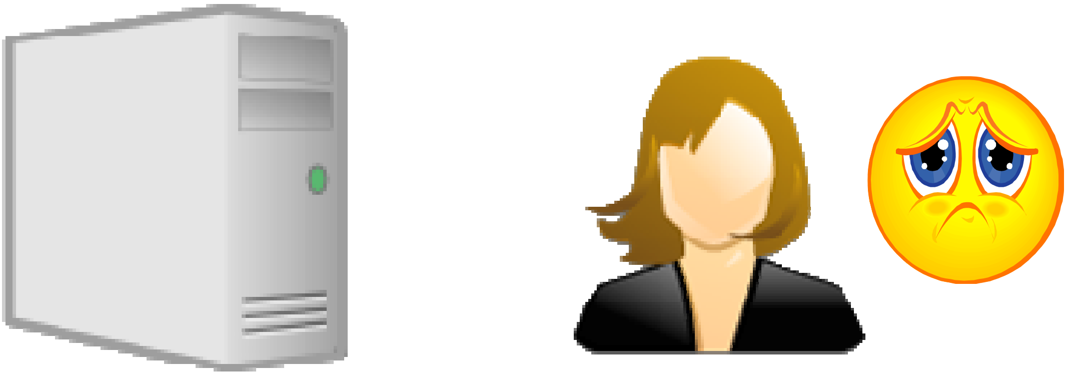
- Fidelity ✓
- Reproducibility ✓
- Efficiency ⊘

Server Consolidation (VMs)



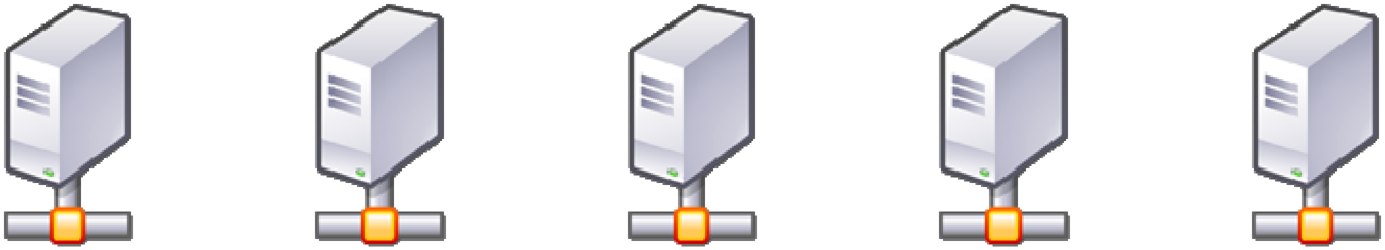
- Fidelity 🚫
- Reproducibility ✓
- Efficiency ✓

Multiplexing Leads to Resource Partitioning



3 GHz CPU, 1 Gbps N/W, 15 Mbps disk I/O, 2 GB RAM

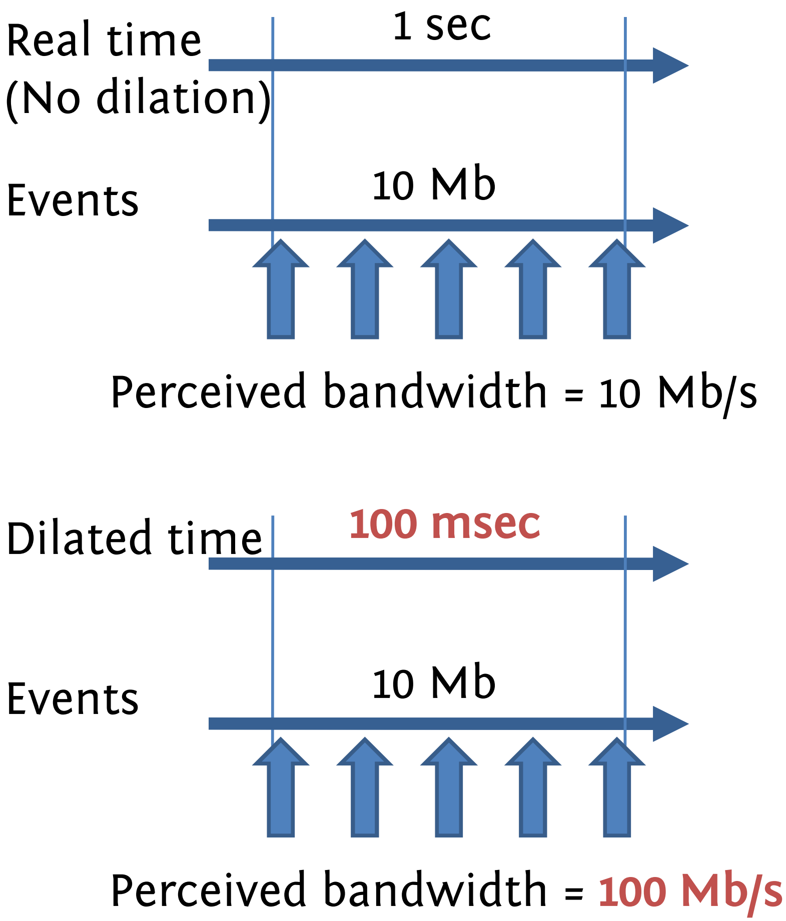
Split equally among 5 VMs



~ 600 MHz CPU, 200 Mbps N/W, 3 Mbps disk I/O, 400 MB RAM **each**

Time Dilation [NSDI 2006]

Key idea: time is also a resource!



Real time (No dilation)

1 sec

Events

10 Mb

Perceived bandwidth = 10 Mb/s

Dilated time

100 msec

Events

10 Mb

Perceived bandwidth = **100 Mb/s**

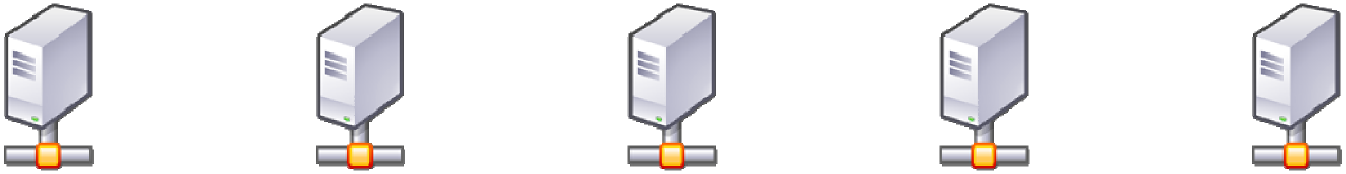
- Slow down passage of time within the OS
- CPU, network, disk – all *appear* faster
- Experiments take longer

Time Dilation Factor (TDF) = Real time/Virtual time

In this example,
TDF = 1sec/100ms = 10

Multiplexing Under Time Dilation

3 GHz CPU, 1 Gbps N/W, 15 Mbps disk I/O, 2 GB RAM

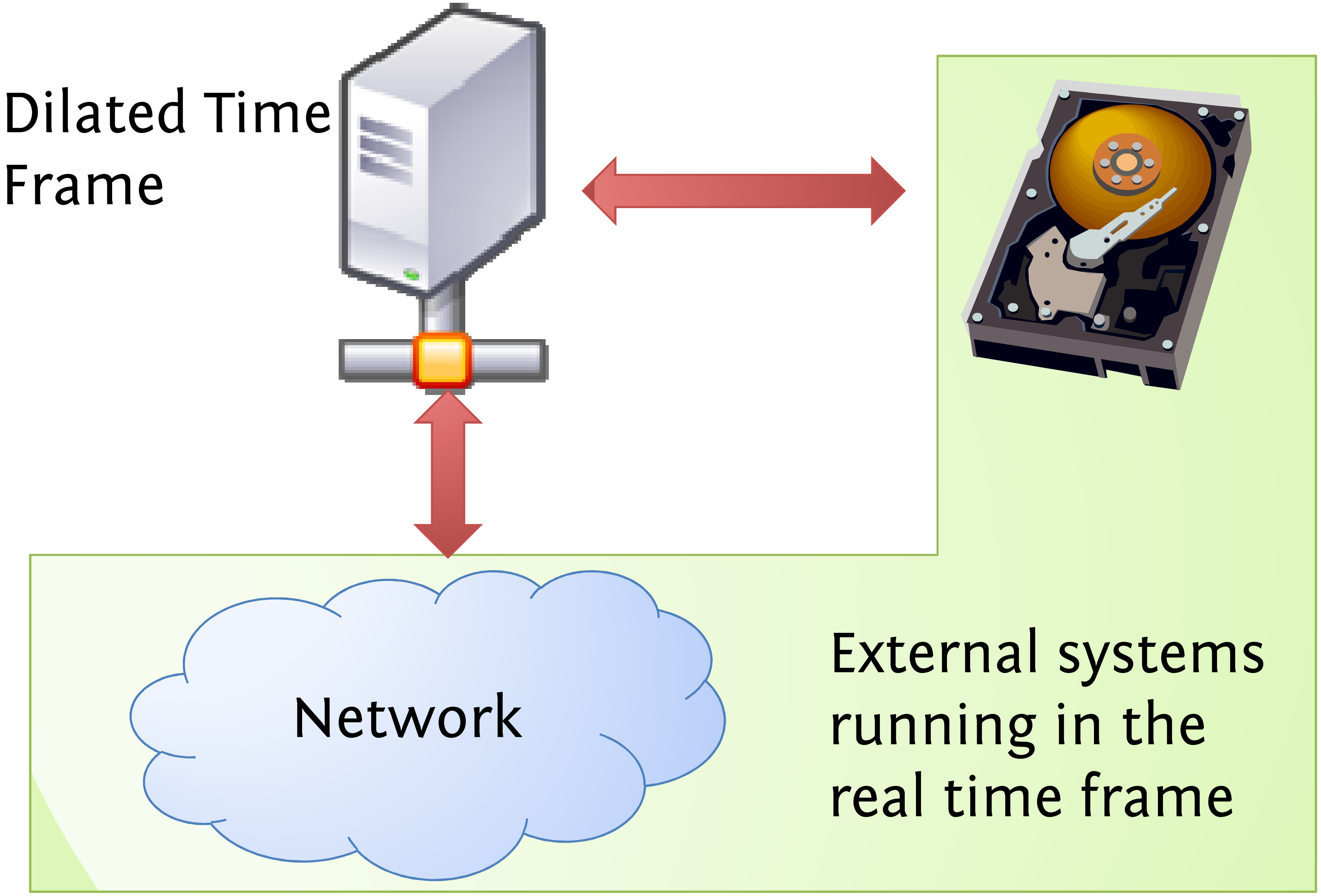


~ 600 MHz CPU, 200 Mbps N/W, 3 Mbps disk I/O, 400-MB RAM, **each**

TDF 5

~ 3 GHz CPU, 1 Gbps N/W, 15 Mbps disk I/O?, 400 MB RAM each

Time Dilation: External Interactions



Dilated Time Frame

Network

External systems running in the real time frame

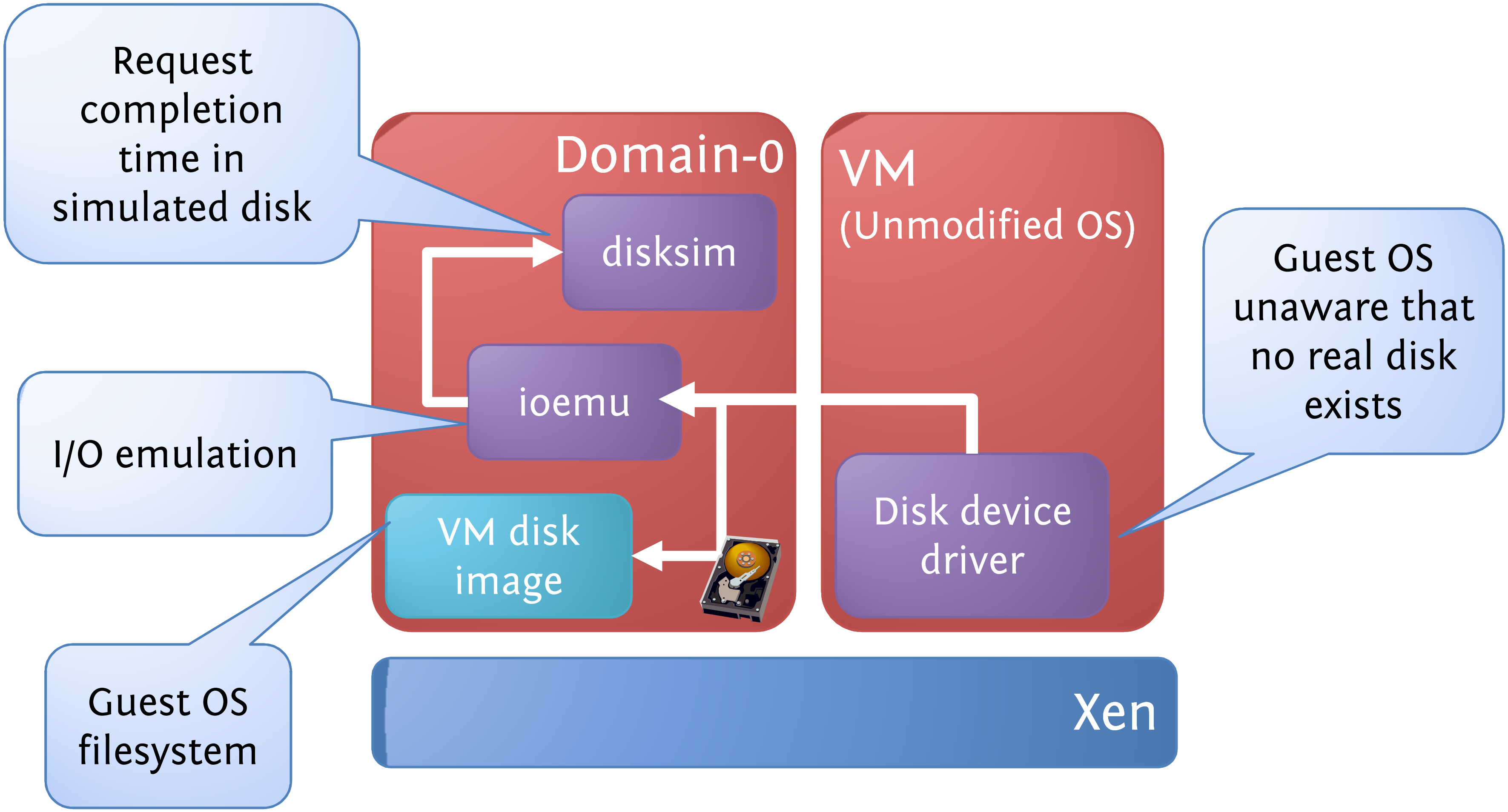
Disk I/O Scaling

- **Invariant**: perceived disk characteristics are preserved
  - Seek time
  - Read/write throughput
- Issues
  - Low level functionality in firmware
  - Different I/O models
  - Per request scaling is difficult

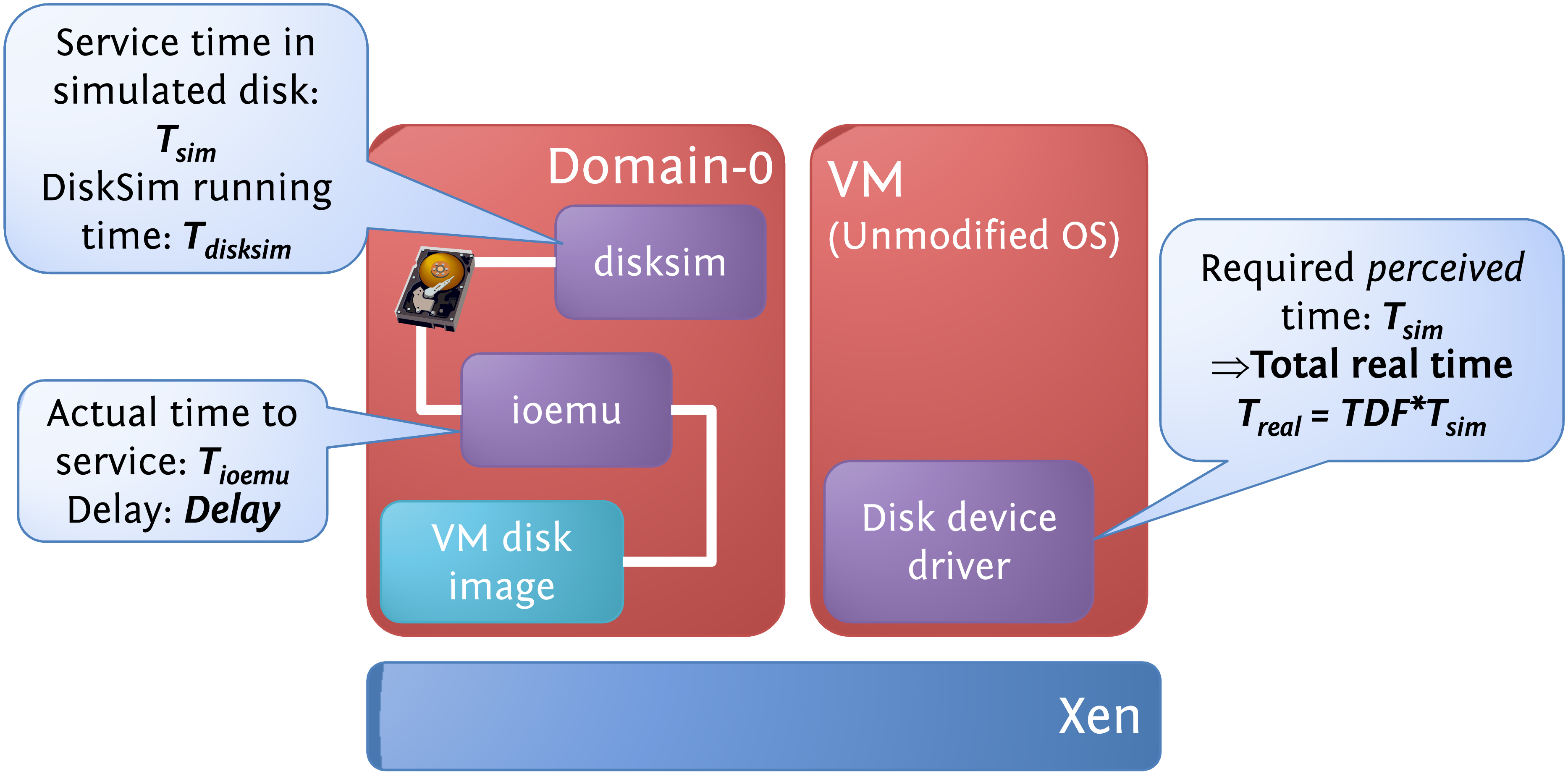
Implementation Details

- Supported platforms
  - Xen 2.0.7, 3.0.4, 3.1
  - Can be ported to non-virtualized systems
- Support for unmodified guest OSes
- Disk I/O scaling for different I/O models
  - Fully virtualized: **integration with DiskSim**
  - Paravirtualized: scaling in device driver

Disk I/O Scaling: Fully Virtualized VMs



Disk I/O Scaling: Fully Virtualized VMs



Service time in simulated disk: T_{sim}
DiskSim running time: $T_{disksim}$

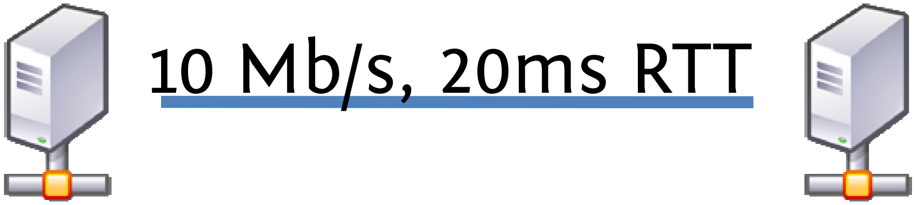
Actual time to service: T_{ioemu}
Delay: ***Delay***

Required *perceived* time: T_{sim}
⇒**Total real time $T_{real} = TDF^{*}T_{sim}$**

$$T_{real} = T_{ioemu} + \mathit{Delay} + T_{disksim}$$
$$\Rightarrow \mathit{Delay} = (TDF^{*}T_{sim}) - T_{disksim} - T_{ioemu}$$

Network I/O Scaling

Invariant: *Perceived* network characteristics (bandwidths and latencies) must be preserved



10 Mb/s, 20ms RTT

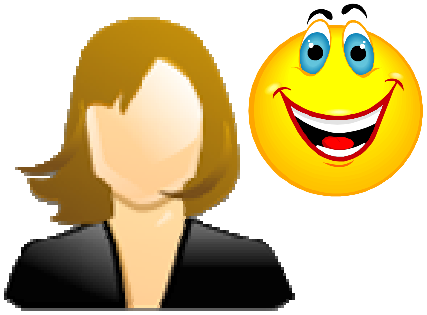
	Real Configuration	Perceived Configuration
Original system (TDF 1)	10 Mb/s, 20 ms	10 Mb/s, 20 ms
Time Dilation (TDF 5)	10 Mb/s, 20 ms	**50 Mb/s, 4 ms**
DieCast (TDF 5)	2 Mb/s, 100 ms	**10 Mb/s, 20 ms**

Network emulation: ModelNet, Dummynet

Recap

- Multiplex VMs for efficiency
- Time dilation to scale resources
- Disk I/O scaling
- Network I/O scaling

- Fidelity ✓
- Reproducibility ✓
- Efficiency ✓

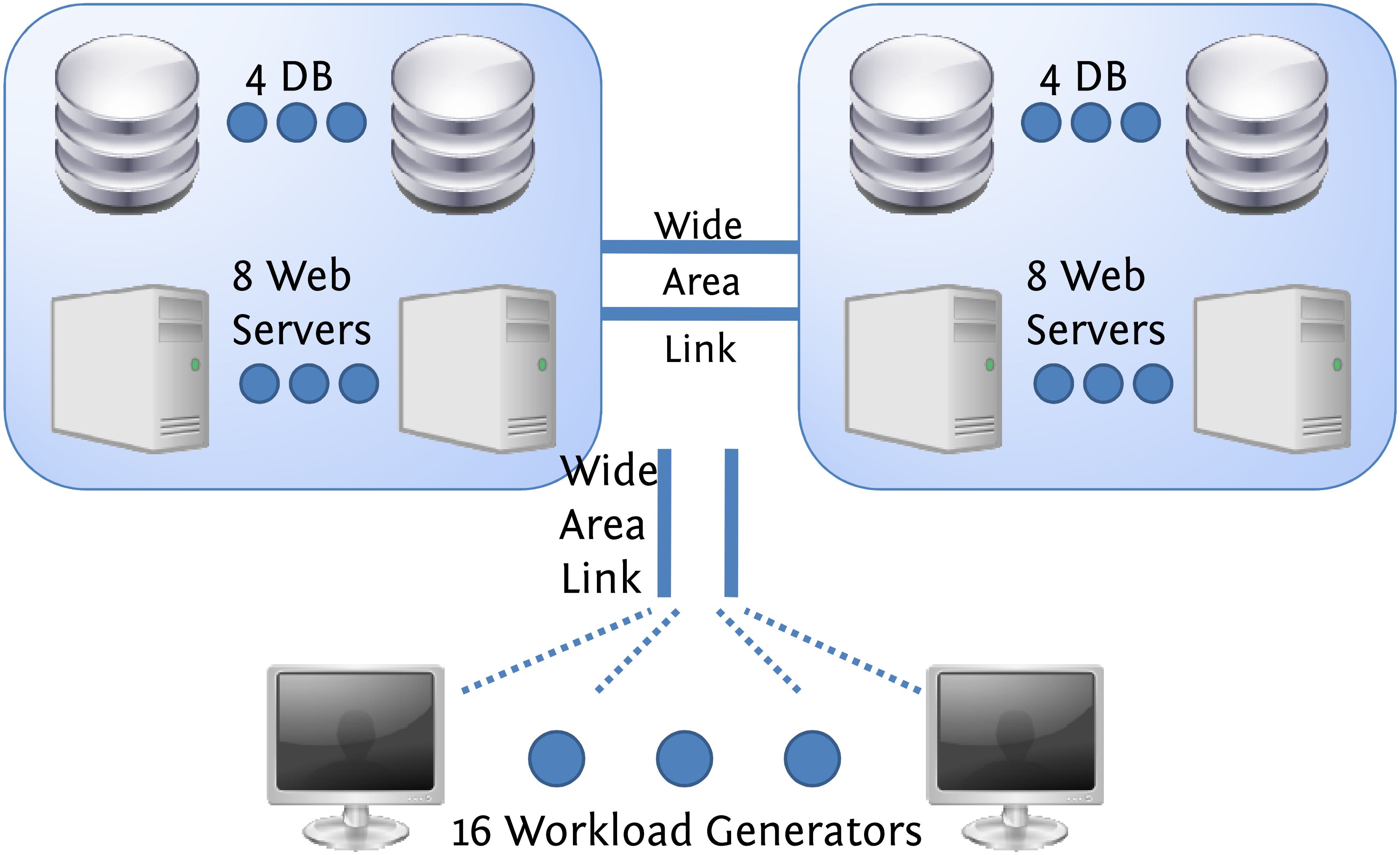


At this point, the scaled system *almost* looks like original system!

Validation

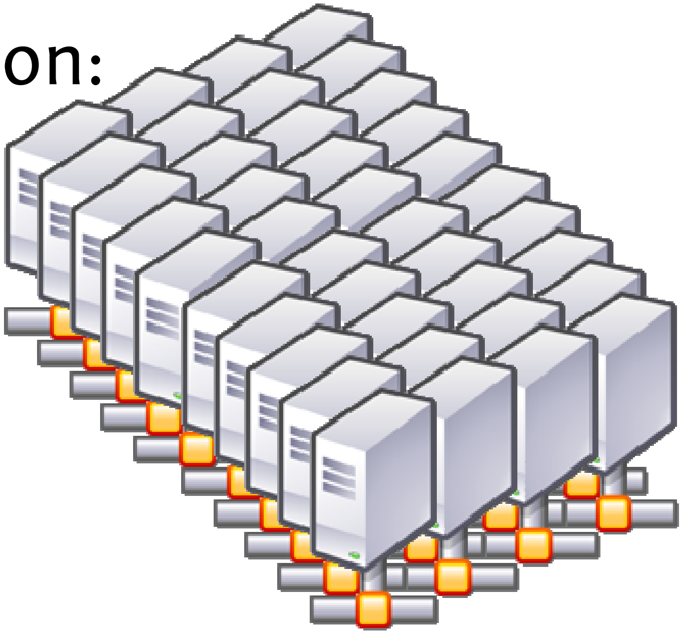
- How well does DieCast scaled performance match the original system?
  - Application specific metrics
- Can a smaller system be configured to match the resources of a larger system?
  - Resource utilization profiles
- Applications: **RUBiS**, BitTorrent, Isaac
- RUBiS
  - eBay like e-Commerce service
  - Ships with workload generator

RUBiS: Topology

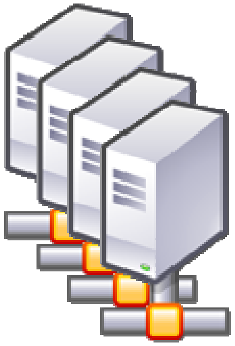


Experimental Setup

Baseline configuration: 40 physical machines

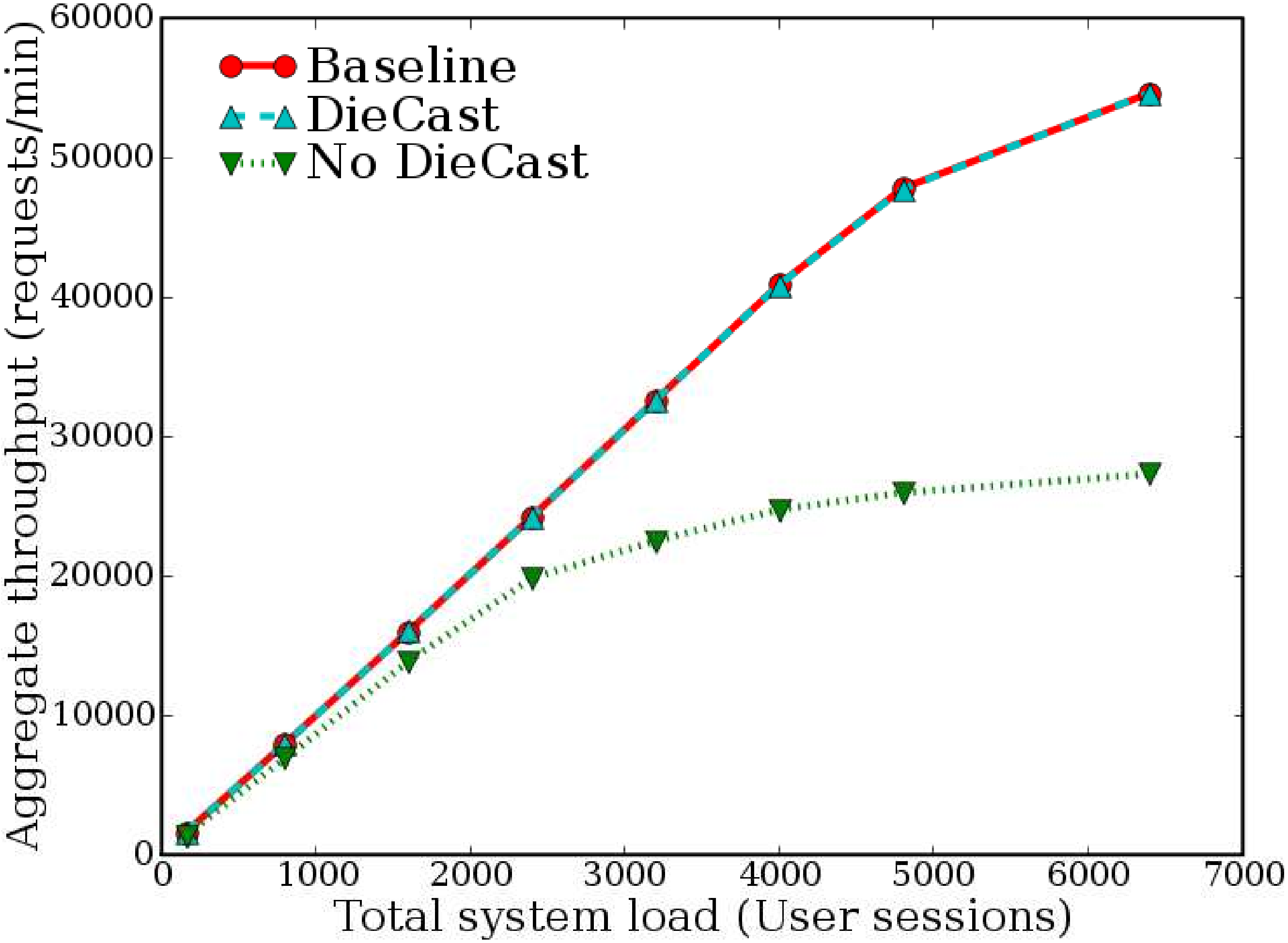


DieCast scaled Configuration: 4 physical machines, 10 VMs each

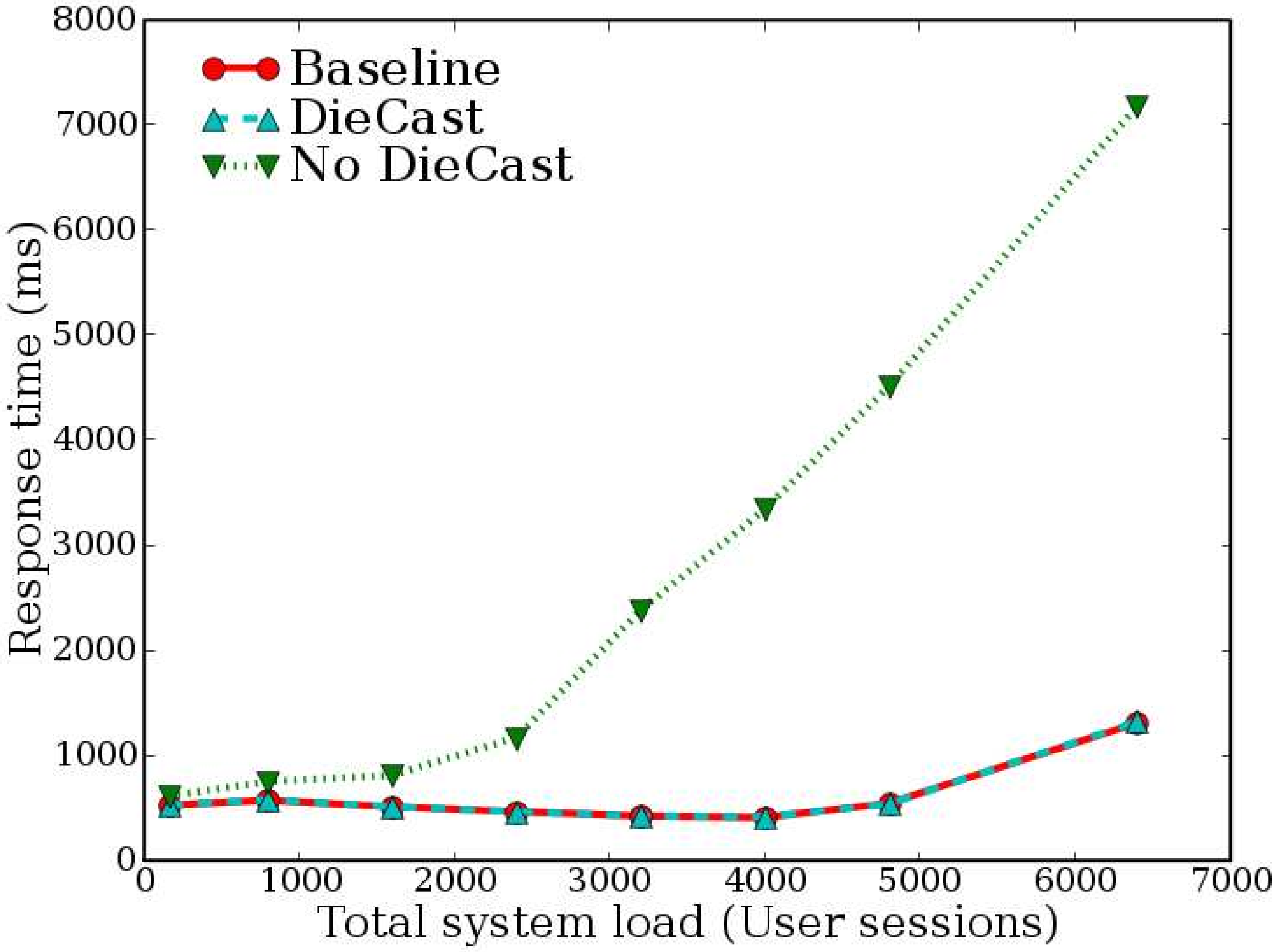


- Xen 3.1, fully virtualized VMs
- Debian Etch, Linux 2.6.17, 256 MB RAM
- DiskSim emulating Seagate ST3217
- Network emulation using ModelNet

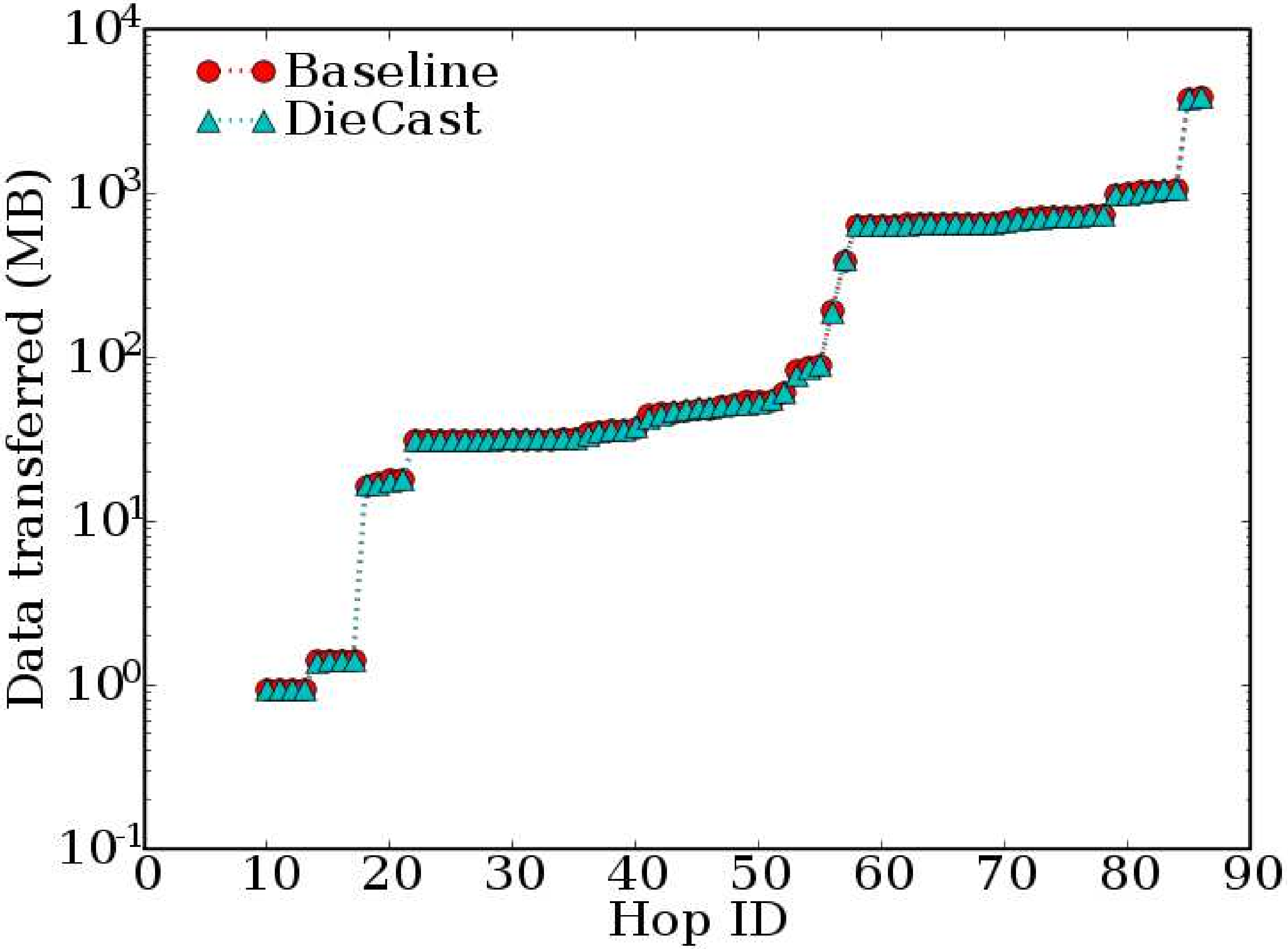
RUBiS: Throughput



RUBiS: Response Time



RUBiS: Resource Usage



Validation Recap

- Evaluated
  - RUBiS
  - BitTorrent
  - Isaac
- Demonstrated
  - Match application specific metrics
  - Preserve resource utilization profile

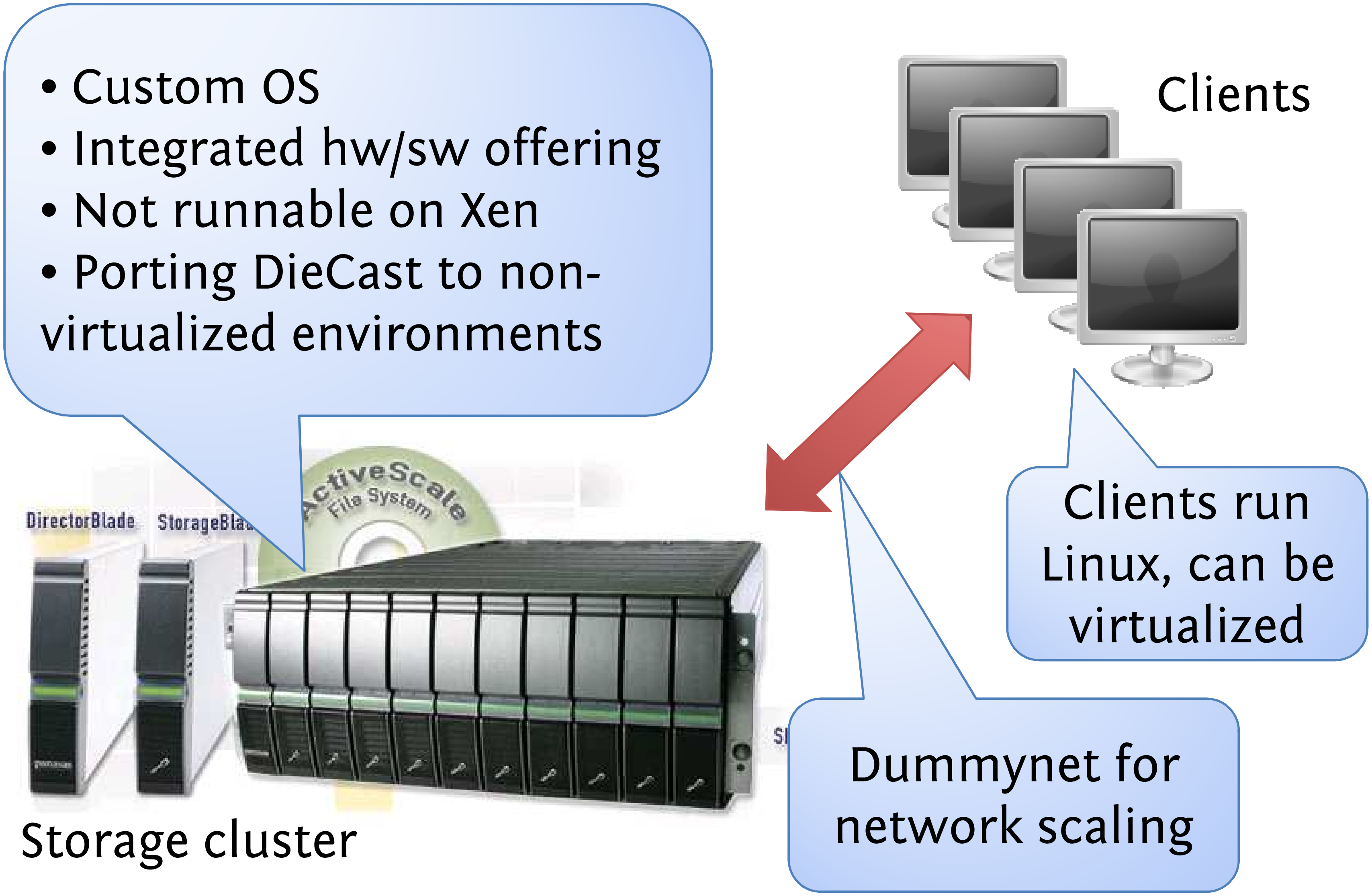
Many more details in the paper

Case study: Panasas

- Panasas builds scalable storage systems for high performance computing
  - http://www.panasas.com
- Caters to variety of clients
- Difficult or even impossible to replicate deployment environment of all clients
- Limited resources for testing

DieCast in Panasas

- Custom OS
- Integrated hw/sw offering
- Not runnable on Xen
- Porting DieCast to non-virtualized environments



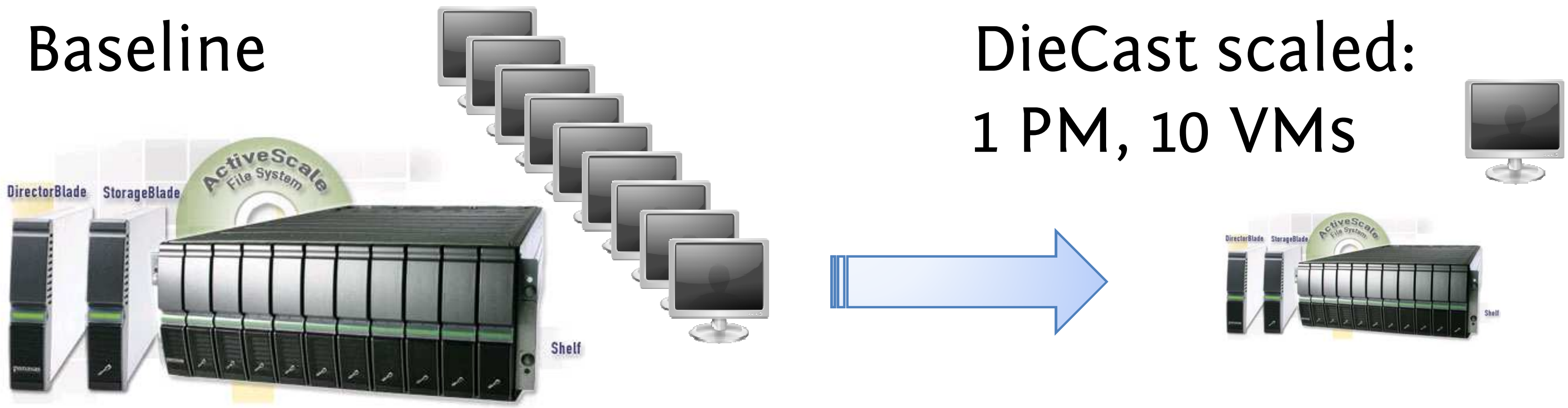
Clients

Clients run Linux, can be virtualized

Dummynet for network scaling

Storage cluster

Panasas: Evaluation Summary



Baseline

DieCast scaled:
1 PM, 10 VMs

- Validation
  - Two benchmarks from standard test suite: IOZone, MPI-IO; varying block sizes
  - Match performance metrics

Scaling: Used 100 machines to scale to **1000 clients**

Limitations

- Memory scaling
- Long running workloads
- Specialized hardware appliances
- Fine grained timing

Summary

- DieCast: scalable testing
  - Fidelity, Reproducibility, Efficiency
- Contributions
  - Support for unmodified operating systems
  - Implement disk I/O scaling (DiskSim integration)
  - CPU scheduler enhancements for time dilation
  - Comprehensive evaluation, including a commercial storage system

Thanks!

Questions?

dgupta@cs.ucsd.edu