To Infinity and Beyond

Time Warped Network Emulation

Diwaker Gupta Ken Yocum Marvin McNett
Alex C. Snoeren Amin Vahdat Geoffrey M. Voelker

Department of Computer Science
University of California, San Diego

May 8, 2006

Imagine if you could . . .

- **evaluate** TCP variants across 100-Gbps wide area links *in your lab*.
- **predict** the performance benefits of upgrading the networking hardware in your cluster.
- **explore** the performance bottlenecks of applications in resource-rich environments.

Time dilation promises all this, and more . . .

UCSDCSE Computer Science and Engineering

Protocol evaluation

The problem

TCP has known performance problems in high capacity networks. Several variants and enhancements have been proposed to address this. How do we evaluate them?

Traditional methodologies

- Test in the wild: e.g. PlanetLab
- Simulation: e.g. NS-2
- Emulation: e.g. ModelNet

Protocol evaluation

Comparing the methodologies

Real world testing

Target hardware is either unavailable or expensive.

Simulation

Might not reflect reality.

Emulation

Limited by the capacity of the underlying hardware.

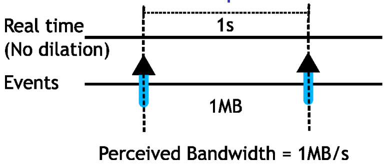
Experiments are limited by available *resources*.

UCSDCSE Computer Science and Engineering

Time Dilation

Key idea: Time is *also* a resource of the system.

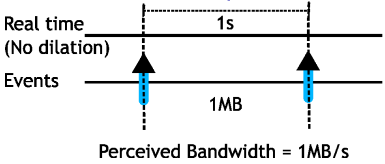
The concept



Time Dilation

Key idea: Time is *also* a resource of the system.

The concept



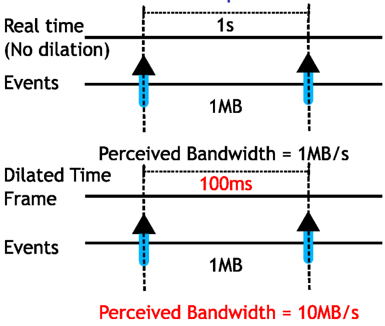
- Slow down passage of time within the OS.
- Perceived capacity increases.

UCSDCSE Computer Science and Engineering

Time Dilation

Key idea: Time is *also* a resource of the system.

The concept

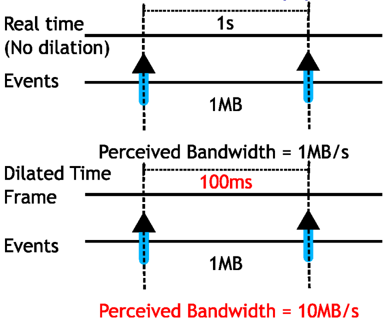


- Slow down passage of time within the OS.
- Perceived capacity increases.

Time Dilation

Key idea: Time is *also* a resource of the system.

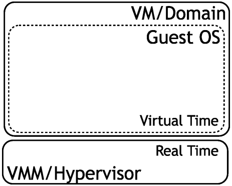
All resources *appear* faster under dilation



- Network: time taken for network I/O.
- CPU: time taken for computation.
- Disk: time taken for disk I/O.

Prototype Implementation

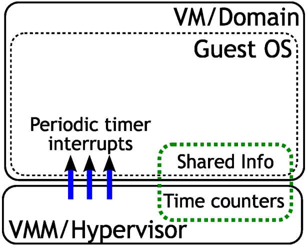
- Built on Xen (2.0.7) and XenoLinux (2.6.11).
- Virtual machines enable a clean architecture for isolating the OS from real time.
- Only fundamental requirement from VMM is the ability to manipulate guest OS's perception of time.
- TDF is the **T**ime **D**ilation **F**actor.



$$\text{TDF} = \frac{\text{Real time}}{\text{Virtual Time}}$$

Time flow in Xen

- Xen exposes time counters to guests: time since boot and wall clock time.
- Time values communicated via a per-VM data structure shared between the VM and the VMM.
- Guest clock is periodically synchronized with the host clock.
- Xen also delivers periodic timer interrupts to VMs.

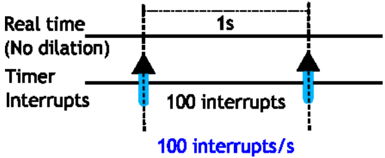


Modifications to the Xen Hypervisor

Variable	Original	Dilated
Time since boot	`TB`	`TB/tdf`
Wall clock time	`WC`	`WC/tdf`

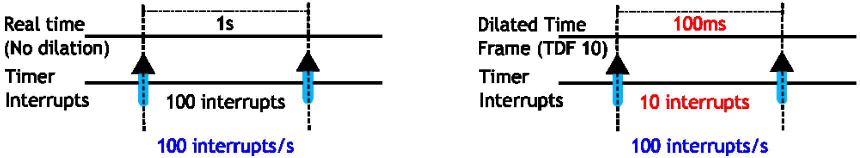
Modifications to the Xen Hypervisor

Variable	Original	Dilated
Time since boot	`TB`	`TB/tdf`
Wall clock time	`WC`	`WC/tdf`



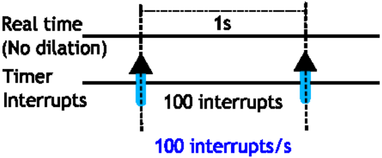
Modifications to the Xen Hypervisor

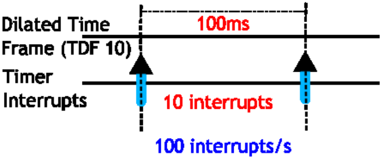
Variable	Original	Dilated
Time since boot	`TB`	`TB/tdf`
Wall clock time	`WC`	`WC/tdf`



Modifications to the Xen Hypervisor

Variable	Original	Dilated
Time since boot	`TB`	`TB/tdf`
Wall clock time	`WC`	`WC/tdf`
Timer interrupts	`INT/sec`	`(INT/tdf)/sec`





Modifications to XenoLinux

- Programmable alarm timers scaled back to real time.
- Scaled value of the Time Stamp Counter (TSC) is read within the kernel.

Prototype properties

- Each VM can run with a different TDF.
- Our modifications are compact: 500 lines of C and Python.

Using time dilation

- Experiments take TDF times longer.
- All end hosts should run with the same TDF.
- Dilation scales all system components (CPU/disk/network) uniformly.
- Dilation does not change the scheduling pattern of the VMs.

Evaluation

Validation

- Evaluate predictive accuracy of time dilation using old hardware.
- Establish accuracy using a single TCP flow.
- Validate under more complex scenarios.

Applications

- Protocol comparison: TCP NewReno vs. TCP BiC.
- Application evaluation: BitTorrent.

UCSDCSE Computer Science and Engineering

Validation methodology

How does one validate time dilation?

- Pick a **baseline** scenario that *is* currently attainable.
- Scale the baseline experiment by the TDF to get the **scaled** configuration.
- Run the scaled configuration under dilation to get the **perceived** configuration.
- Compare the **perceived** configuration with the **baseline** configuration.

Validation methodology

Invariant for validation of network dilation

Network characteristics (*perceived* bandwidths and latencies) must be preserved.

TDF	Real configuration	Perceived configuration
1	100 Mbps, 80 ms	100 Mbps, 80 ms

UCSDCSE Computer Science and Engineering

Validation methodology

Invariant for validation of network dilation

Network characteristics (*perceived* bandwidths and latencies) must be preserved.

TDF	Real configuration	Perceived configuration
1	100 Mbps, 80 ms	100 Mbps, 80 ms
10	100 Mbps, 80 ms	1000 Mbps, 8 ms

UCSDCSE Computer Science and Engineering

Validation methodology

Invariant for validation of network dilation

Network characteristics (*perceived* bandwidths and latencies) must be preserved.

TDF	Real configuration	Perceived configuration
1	100 Mbps, 80 ms	100 Mbps, 80 ms
10	100 Mbps, 80 ms	1000 Mbps, 8 ms
10	10 Mbps, 800 ms	100 Mbps, 80 ms

Validation methodology

Invariant for validation of network dilation

Network characteristics (*perceived* bandwidths and latencies) must be preserved.

TDF	Real configuration	Perceived configuration
1	100 Mbps, 80 ms	100 Mbps, 80 ms
10	100 Mbps, 80 ms	1000 Mbps, 8 ms
10	10 Mbps, 800 ms	100 Mbps, 80 ms
100	1 Mbps, 8000 ms	100 Mbps, 80 ms

- Link characteristics scaled according to TDF.
- Emulated using Dummynet and ModelNet.
- Dilation makes emulation *easier*.

UCSDCSE
Computer Science and Engineering

Experimental setup

Topology

- *N*: Total number of flows (netperf).
- *RTT*: Round trip time.
- *C*: Capacity of the bottleneck link.

Hardware validation

Can we use old hardware to predict performance of new hardware?

Configuration: N = 50 flows, RTT = 80 ms, C = 500 Mbps

Hardware Configuration	TDF	Mean (Mbps)	St.Dev. (Mbps)
2.6-GHz, 1-Gbps NIC restricted to 500 Mbps	1	9.39	1.91

UCSDCSE Computer Science and Engineering

Hardware validation

Can we use old hardware to predict performance of new hardware?

Configuration: N = 50 flows, RTT = 80 ms, C = 500 Mbps

Hardware Configuration	TDF	Mean (Mbps)	St.Dev. (Mbps)
2.6-GHz, 1-Gbps NIC restricted to 500 Mbps	1	9.39	1.91
1.13-GHz, 1-Gbps NIC restricted to 250 Mbps	2	9.57	1.76

UCSDCSE Computer Science and Engineering

Hardware validation

Can we use old hardware to predict performance of new hardware?

Configuration: N = 50 flows, RTT = 80 ms, C = 500 Mbps

Hardware Configuration	TDF	Mean (Mbps)	St.Dev. (Mbps)
2.6-GHz, 1-Gbps NIC restricted to 500 Mbps	1	9.39	1.91
1.13-GHz, 1-Gbps NIC restricted to 250 Mbps	2	9.57	1.76
500-MHz, 100-Mbps NIC	5	9.70	2.04

UCSDCSE
Computer Science and Engineering

Hardware validation

Can we use old hardware to predict performance of new hardware?

Proof of concept that time dilation has potential for predicting performance of newer hardware.

Confi ... lbps

Hard...		(Mbps)	St.Dev. (Mbps)
2.6-GHz, 1-Gbps NIC restricted to 500 Mbps	1	9.39	1.91
1.13-GHz, 1-Gbps NIC restricted to 250 Mbps	2	9.57	1.76
500-MHz, 100-Mbps NIC	5	9.70	2.04

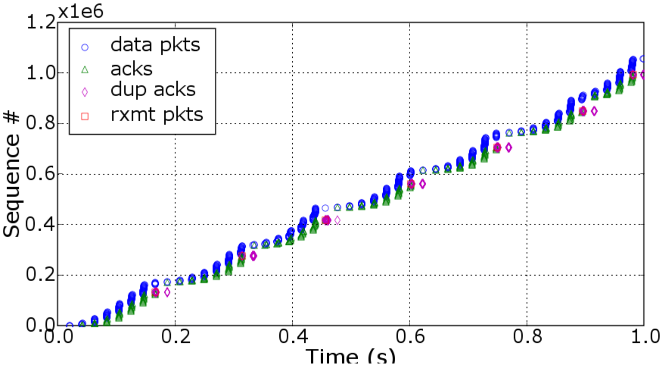
UCSDCSE Computer Science and Engineering

Single TCP flow

Packet level behavior

Baseline: N = 1 flow, C = 100 Mbps, RTT = 20 ms, 1 % deterministic losses.

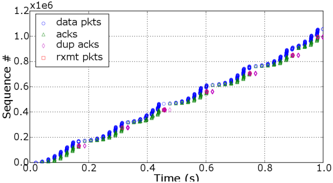
First second of trace.



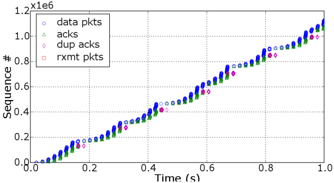
Single TCP flow

Packet level behavior

Baseline



TDF 10

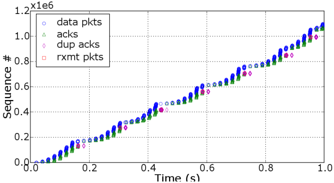


Perceived configuration

N = 1 flow, C = 100 Mbps, RTT = 20 ms.
1 % deterministic losses.

First second of trace.

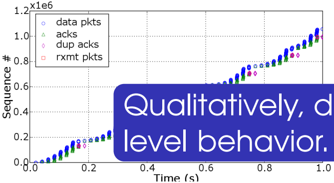
TDF 100



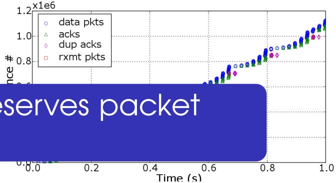
Single TCP flow

Packet level behavior

Baseline



TDF 10



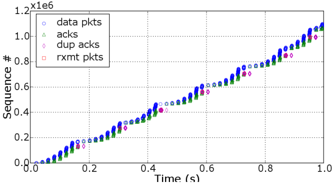
Qualitatively, dilation preserves packet level behavior.

Perceived configuration

N = 1 flow, C = 100 Mbps, RTT = 20 ms.
1 % deterministic losses.

First second of trace.

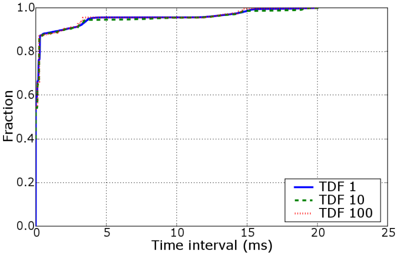
TDF 100



Single TCP flow

CDF of inter packet transmission times under 1% deterministic loss

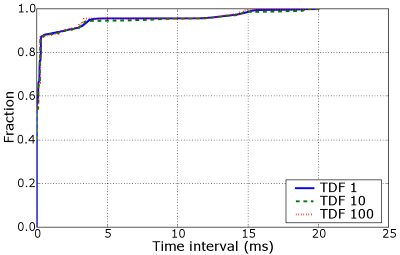
Distributions are similar.



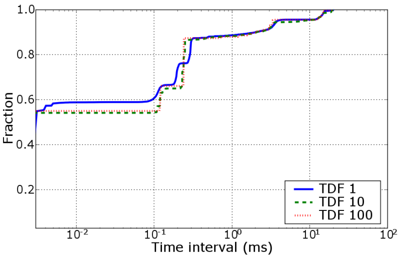
Single TCP flow

CDF of inter packet transmission times under 1% deterministic loss

Distributions are similar.



Closer look at the long tail

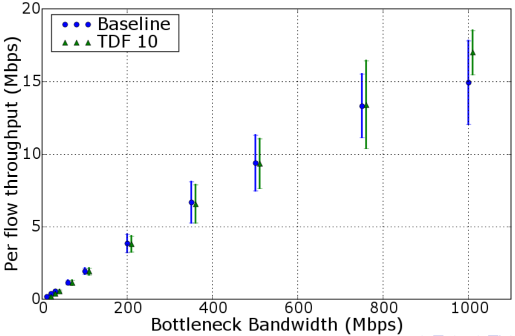


Dilation with multiple flows

Accuracy under varying bandwidth

Configuration: N = 50 flows.

TDF	Real Configuration	Perceived Configuration
1	B=b Mbps, RTT=80 ms	B=b Mbps, RTT=80 ms
10	B=b/10 Mbps, RTT=800 ms	B=b Mbps, RTT=80 ms



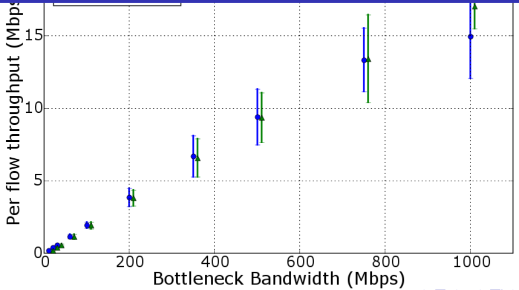
Dilation with multiple flows

Accuracy under varying bandwidth

Configuration: N = 50 flows.

TDF	Real Configuration	Perceived Configuration
1	B=b Mbps, RTT=80 ms	B=b Mbps, RTT=80 ms
10	B=b/10 Mbps, RTT=800 ms	B=b Mbps, RTT=80 ms

Dilation is accurate for multiple flows under varying bandwidth.



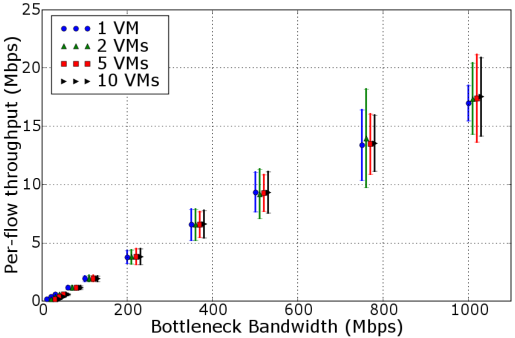
UCSDCSE Computer Science and Engineering

Dilation with multiple flows

Overhead of multiplexing multiple VMs

Configuration: N = 50 flows, TDF = 10

TDF	Real Configuration	Perceived Configuration
10	B=b/10 Mbps, RTT=800 ms	B=b Mbps, RTT=80 ms



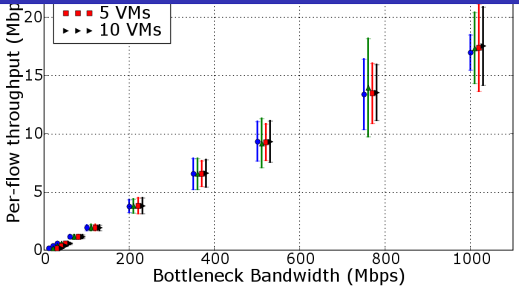
Dilation with multiple flows

Overhead of multiplexing multiple VMs

Configuration: N = 50 flows, TDF = 10

TDF	Real Configuration	Perceived Configuration
10	B=b/10 Mbps, RTT=800 ms	B=b Mbps, RTT=80 ms

Overhead of multiplexing VMs does not degrade performance.

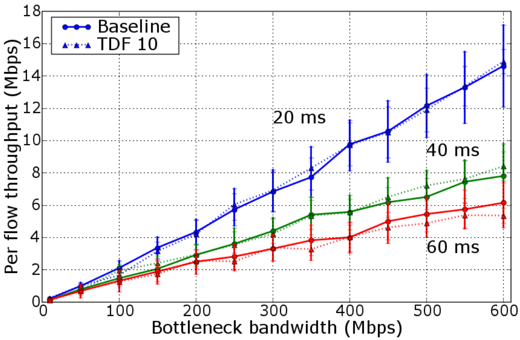


Dilation with multiple flows

Accuracy across multiple RTTs

Configuration: C = 150 Mbps. N = 60 flows in three groups of 20 each.
Each group experiences different RTT $\in \{20, 40, 60\}$ ms.

TDF	Real Configuration	Perceived Configuration
1	B=b Mbps, RTT=t ms	B=b Mbps, RTT=t ms
10	B=b/10 Mbps, RTT=10t ms	B=b Mbps, RTT=t ms



UCSDCSE Computer Science and Engineering

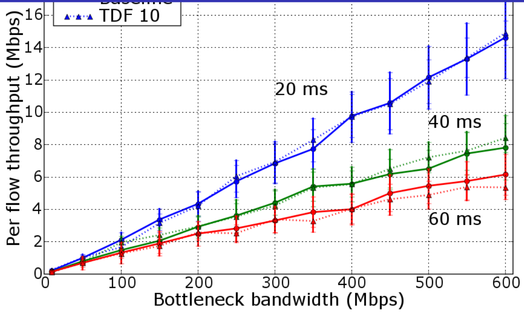
Dilation with multiple flows

Accuracy across multiple RTTs

Configuration: C = 150 Mbps. N = 60 flows in three groups of 20 each.
Each group experiences different RTT $\in \{20, 40, 60\}$ ms.

TDF	Real Configuration	Perceived Configuration
1	B=b Mbps, RTT=t ms	B=b Mbps, RTT=t ms
10	B=b/10 Mbps, RTT=10t ms	B=b Mbps, RTT=t ms

Dilation remains accurate for heterogeneous flows.



UCSDCSE Computer Science and Engineering

Protocol evaluation

TCP NewReno vs. TCP BiC

Default TCP implementation in Linux: NewReno + SACK.

BiC (Xu et. al, 2004)

- Enhancement to TCP's congestion control algorithm for better performance in high BDP networks.
- Implemented in the Linux 2.6.x kernels.

Goal

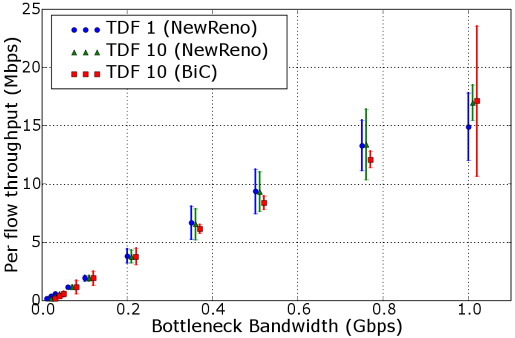
- Treat protocols as black boxes.
- Push beyond the hardware limit.
- Can time dilation uncover interesting behavior?

TCP NewReno vs. TCP BiC

0 – 1 Gbps

Configuration: N = 50 flows.

TDF	Real Configuration	Perceived Configuration
1	B=b Mbps, RTT=80 ms	B=b Mbps, RTT=80 ms
10	B=b/10 Mbps, RTT=800 ms	B=b Mbps, RTT=80 ms



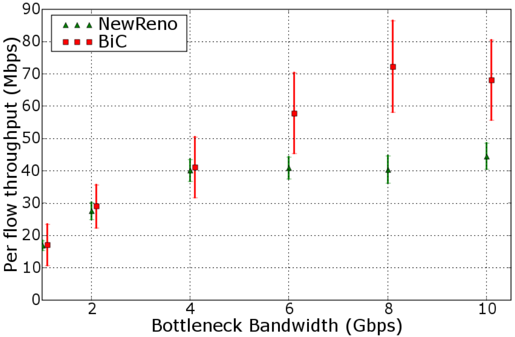
Time dilation can be validated in this range.

TCP NewReno vs. TCP BiC

1 – 10 Gbps

Configuration: N = 50 flows, TDF = 10.

TDF	Real Configuration	Perceived Configuration
10	B=b/10 Mbps, RTT=800 ms	B=b Mbps, RTT=80 ms



BiC begins to outperforms NewReno.

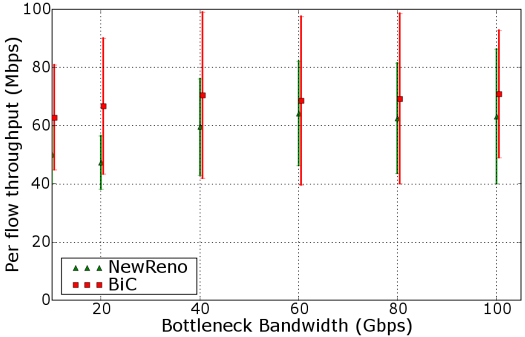
UCSDCSE Computer Science and Engineering

TCP NewReno vs. TCP BiC

10 – 100 Gbps

Configuration: N = 50 flows, TDF = 100.

TDF	Real Configuration	Perceived Configuration
100	B=b/100 Mbps, RTT=8000 ms	B=b Mbps, RTT=80 ms



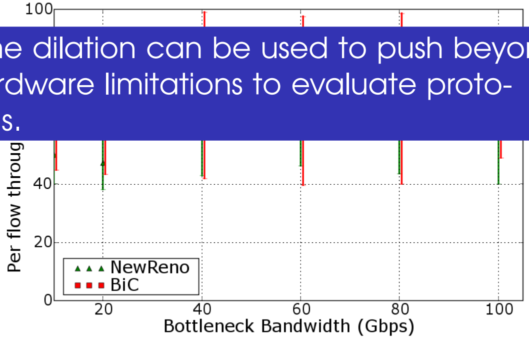
Both NewReno and BiC level off.

TCP NewReno vs. TCP BiC

10 – 100 Gbps

Configuration: N = 50 flows, TDF = 100.

TDF	Real Configuration	Perceived Configuration
100	B=b/100 Mbps, RTT=8000 ms	B=b Mbps, RTT=80 ms



Time dilation can be used to push beyond hardware limitations to evaluate protocols.

Both NewReno and BiC level off.

UCSDCSE Computer Science and Engineering

Evaluating BitTorrent

Goals

- Explore BitTorrent's performance in resource-rich environments.
- Use time dilation to work around limitations of physical resources in the test bed.
- As we increase resources, how does the performance evolve?

UCSDCSE Computer Science and Engineering

Evaluating BitTorrent

Experimental setup

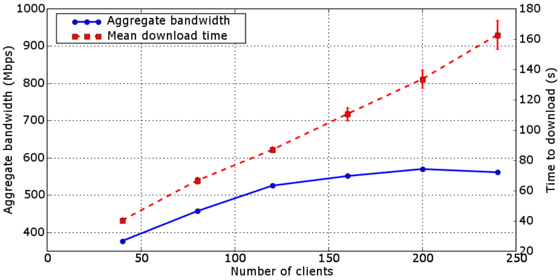
- Unconstrained topology emulated over Modelnet.
- Clients download a file from the seeder node.
- Clients distributed across 10 VMs on 10 physical machines (1 VM on each physical machine).
- Measure average download time across clients.
- Aggregate bandwidth = number of clients × average bandwidth.

UCSDCSE
Computer Science and Engineering

Evaluating BitTorrent

Baseline (TDF 1)

Configuration: TDF = 1. Perceived network capacity is 1 Gbps.

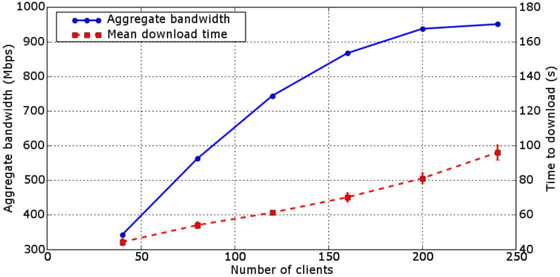


Performance degrades as clients contend for CPU resources.

Evaluating BitTorrent

Removing CPU contention with dilation

Configuration: TDF = 10. Real bandwidth set to 100 Mbps, so perceived network capacity is same as in the baseline case (1 Gbps).

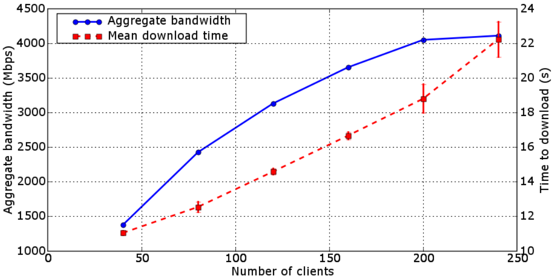


Dilation removes CPU contention, but now the network is saturated.

Evaluating BitTorrent

Scaling all resources using dilation

Configuration: TDF = 10. Real bandwidth set to 1 Gbps, so perceived network capacity is 10 Gbps.



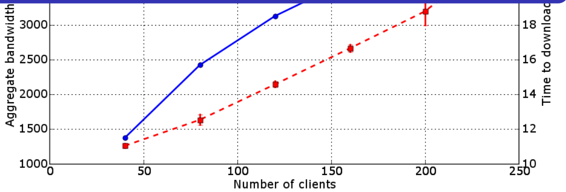
CPU again starts to become a bottleneck.

Evaluating BitTorrent

Scaling all resources using dilation

Configuration: TDF = 10. Real bandwidth set to 1 Gbps, so perceived network capacity is 10 Gbps.

Time dilation can be used to evaluate high-performance network applications in resource-rich environments.



CPU again starts to become a bottleneck.

Summary

- Time dilation: a powerful technique for evaluating distributed systems.
- Network dilation remains accurate for a wide range of interesting configurations.
- Time dilation can be used to evaluate protocols and high-bandwidth applications.

Moving forward

Time dilation with unmodified OSes

Utilize hardware support for virtualization (Intel VT, AMD Pacifica).

Reverse dilation

Using TDF < 1 for emulating extremely long, low-bandwidth traces.

Emulating distributed systems using fewer resources

Can we use dilation to emulate a 100 machine system using only 10 machines?

UCSDCSE Computer Science and Engineering

Thanks

Questions?

`http://sysnet.ucsd.edu/projects/time-dilation`

UCSDCSE
Computer Science and Engineering