Akamai

Akarouting:

A Better Way to Go

Akamai

Akarouting Team:

Claudson Bornstein

Tim Canfield

Gary Miller

DELIVERING A BETTER INTERNET℠

Akarouting Contributors:

- Guy Blelloch
- Timo Burkhard
- Hilla Dishon
- Michelle Henley
- Satish Rao
- Margaret Reid-Miller
- Marc Ringel
- Jennifer Sun
- ShangHua Teng
- Hoeteck Wee
- Joel Wein

Talk Outline

- Introduction
  (Making the Internet Faster and More Reliable)
- The Triangle Inequality
- Applications of Akarouting
- Experimental Foundations
- Design Principals determined from Experiments
- Major Components built for the Akarouting Project
- EdgeSuite download times using Akarouting
- Akarouting’s Effect on Bandwidth usage
- Future Applications

Network Reliability

- Transient Internet glitches:

  E.g..: I can't get from my home to Yahoo but Akamai can get to both sites.

  Usually don't last very long.

Network Reliability

- Major outages
  - L3 melts down (Dec 2000)
  - PSI and C&W stop peering (Jun 2001)
  - 9/11

Can a company with the presences of Akamai use only 32-bit stamps to move packets?

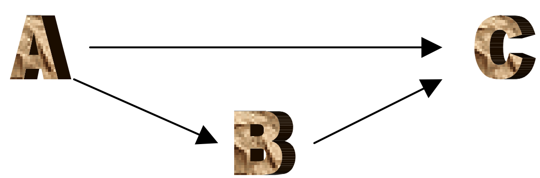
- The Internet gives you only one way to communicate a 32 bit IP address.
- One bit at an Akamai server may translate into 100 end user bits.
- Akamai needs alternate routes. We must have higher reliability!
- Our goal was to have our cake and eat it too!
  - Higher Reliability.
  - Faster Download Times
  - Small Increase in bandwidth Usage.
  - Low Budget (off-the-shelf components).

Have Our Cake and Eat it Too!

Download speed

Perfect World:

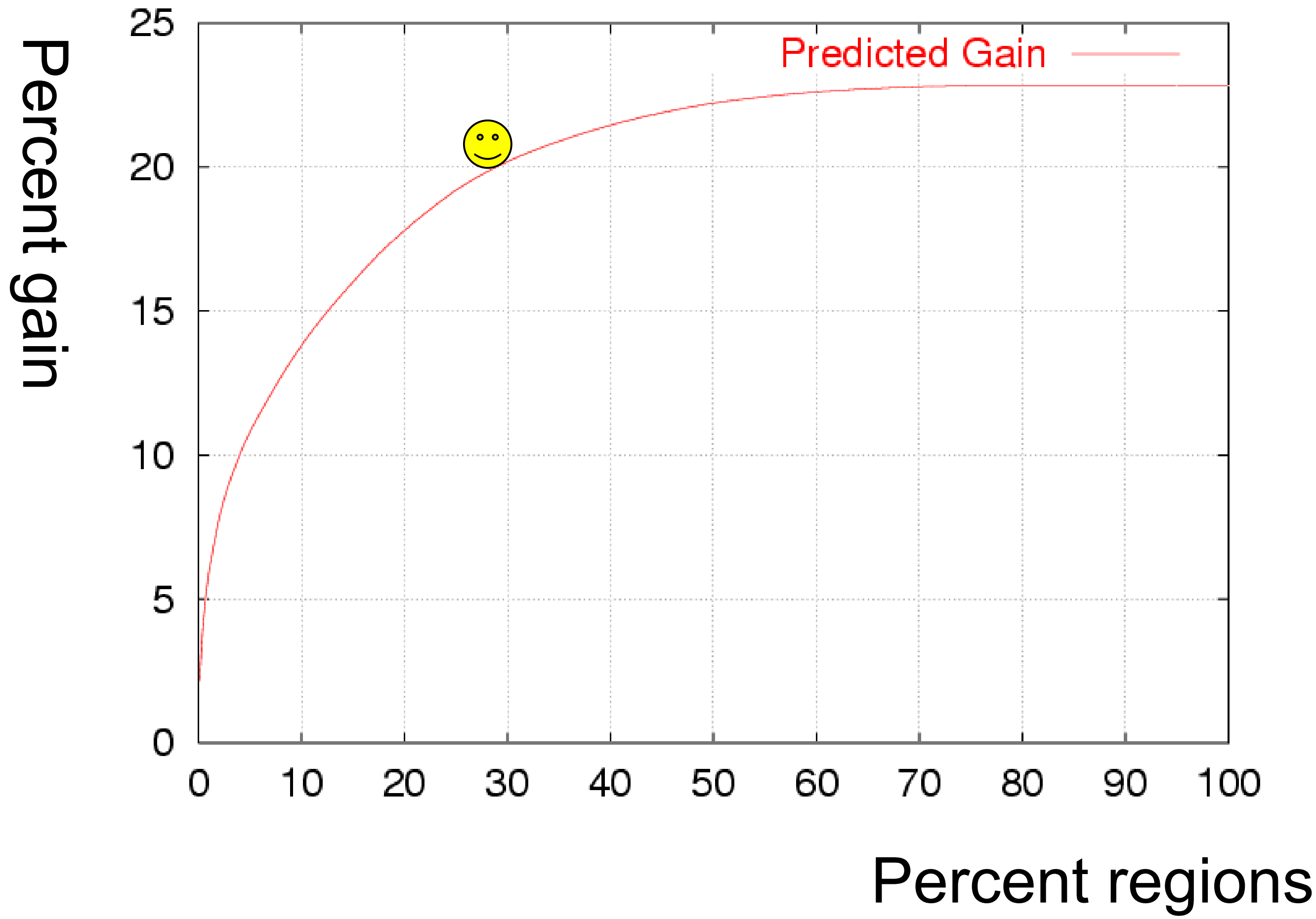
Ping-Time(A,C) $\leq$ Ping-Time(A,B) + Ping-Time(B,C)



Real World:

Average gain from Akamai regions to Yahoo via another Akamai region varies between 15 to 30%

Two-hop x direct ping times



The Akarouting Vision

- We will move traffic from a region A to a region B by sending it through an intermediate region C.
- An Instance of Tunneling

Possible Application for Akarouting

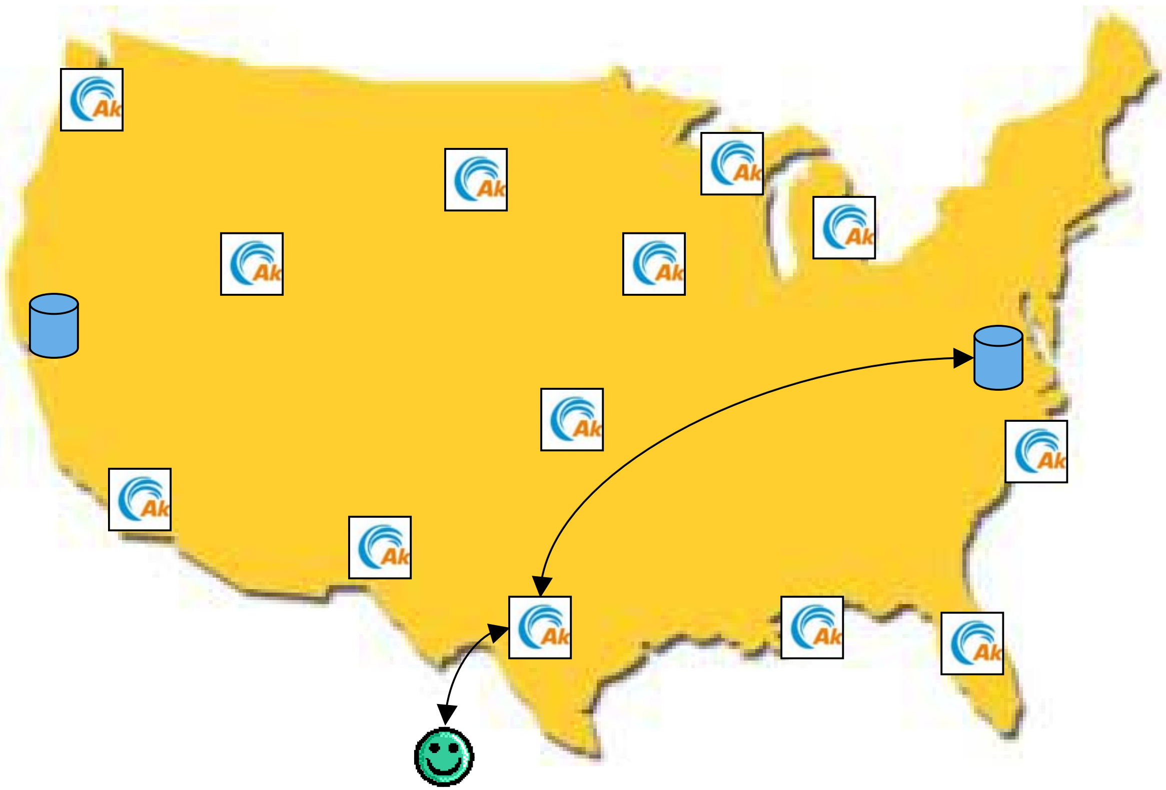
- SSH (Original)
- Streaming Network
- Akamai Powered Web browser
- Voice over IP
- VPN
- EdgeSuite

Application: EdgeSuite/ESI

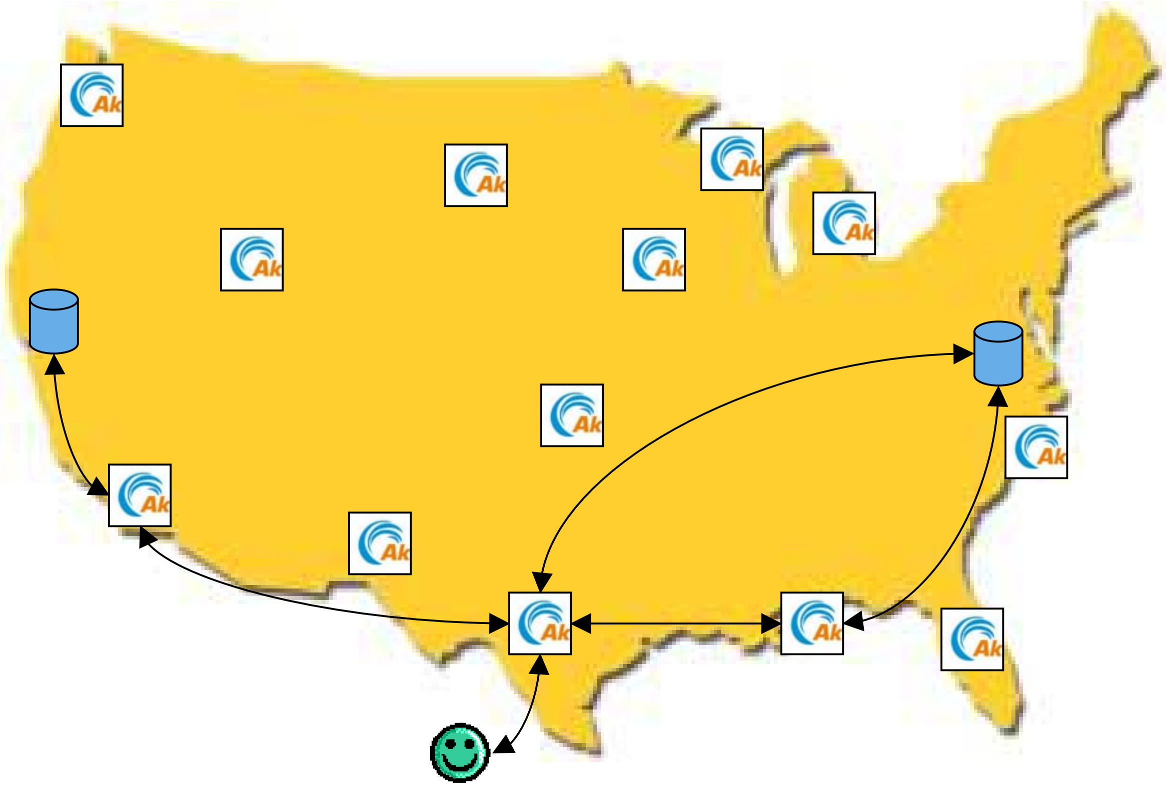
- (ESI) Assembly on the edge
  and why Akarouting:

  - Small amount of time-critical dynamic content to/from Akamai and CP.
  - Akarouting can make EdgeSuite faster and more reliable.

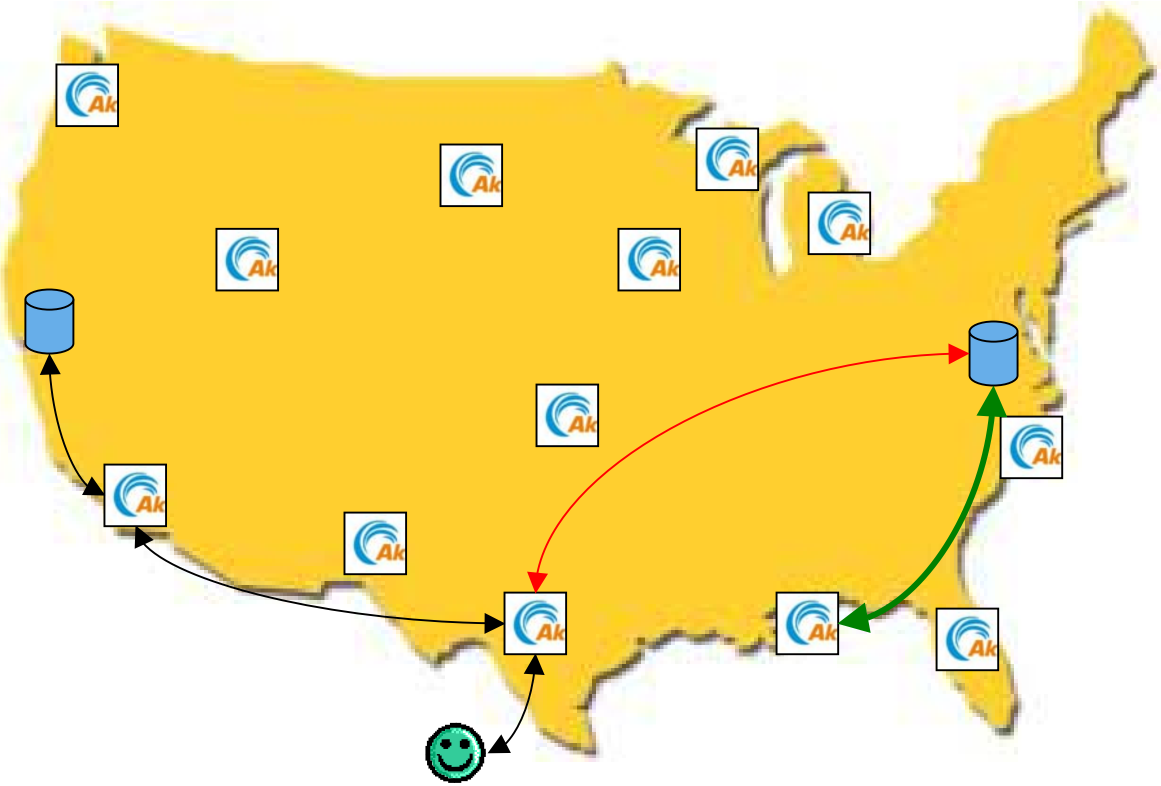
EdgeSuite (no Akarouting)



Akarouting Example



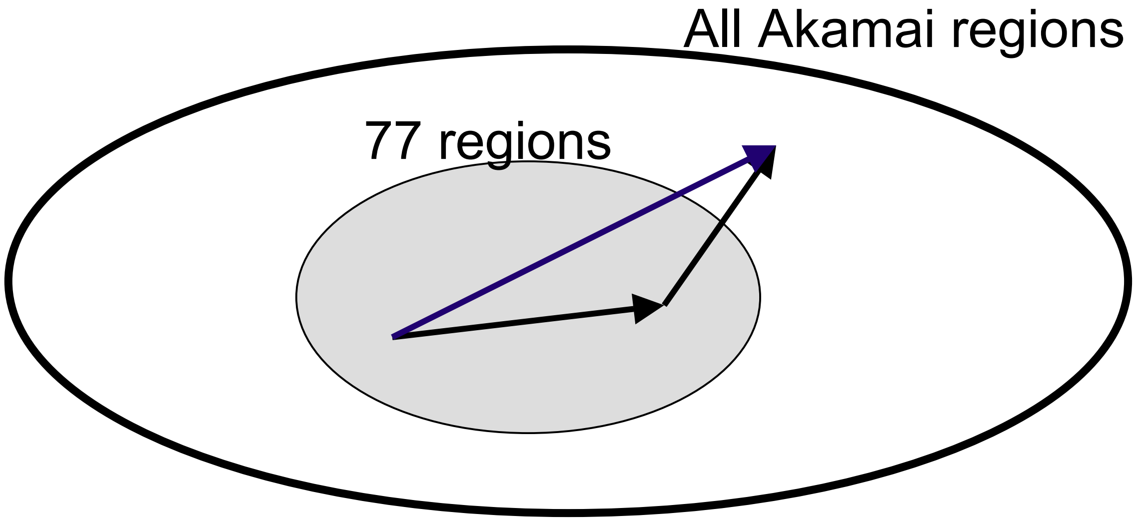
Akarouting Example



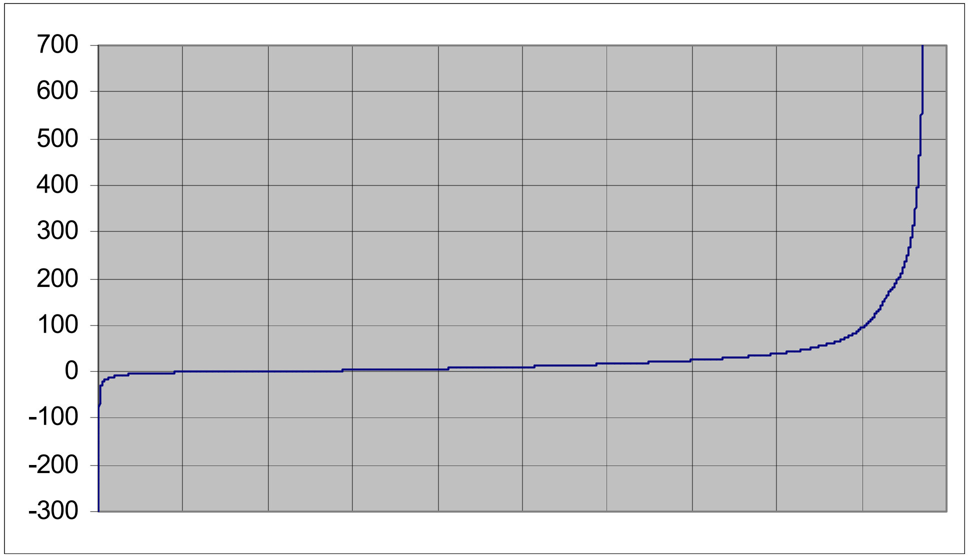
Back To the Laboratory

- In the design phase of this project, We ran experiments on about 40 machines scattered around the world.
- We will also show numbers recorded by our Akaroute MapMaker.

Measuring Ping Time Gains

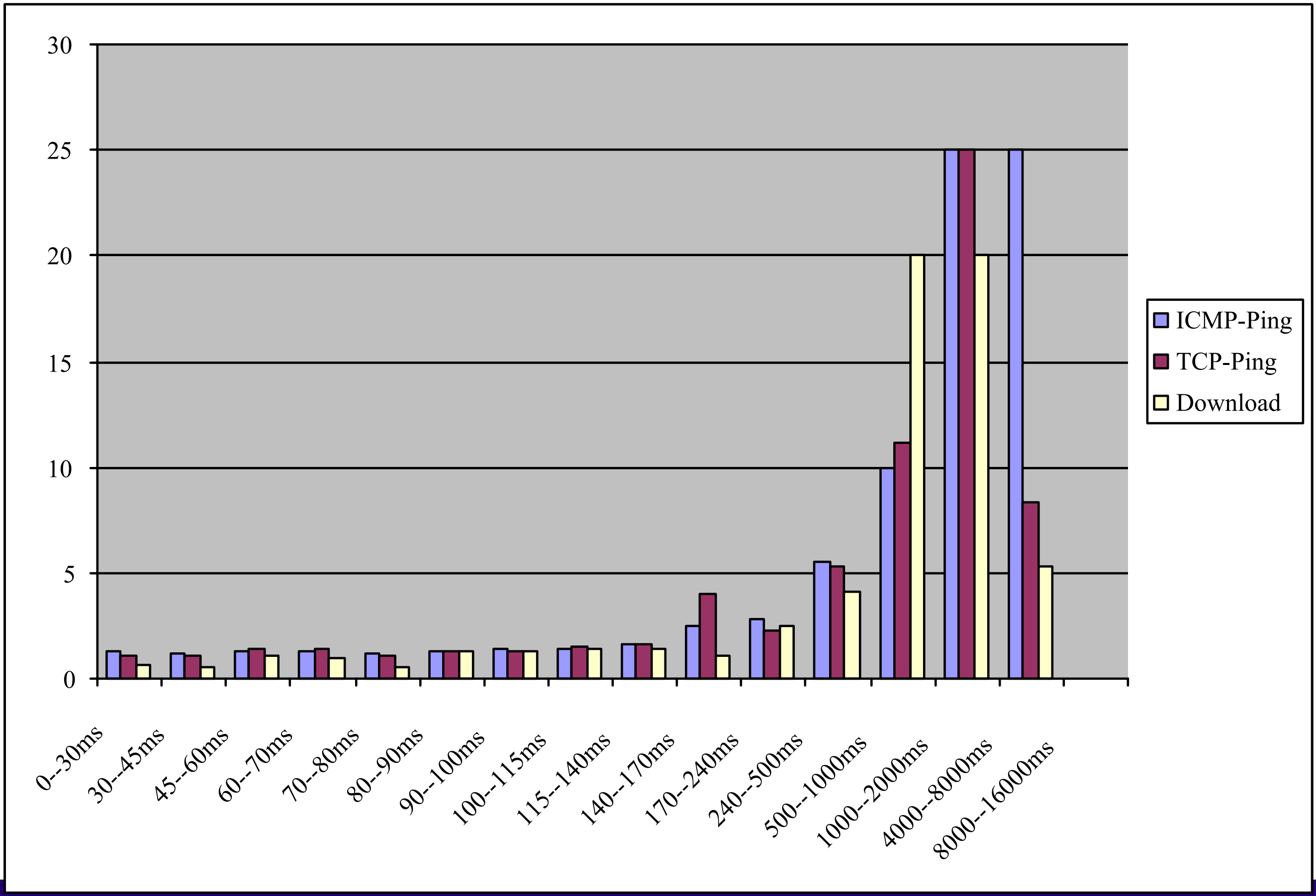


Absolute differences



Sorted ping differences for 30,000 pairs of centers.

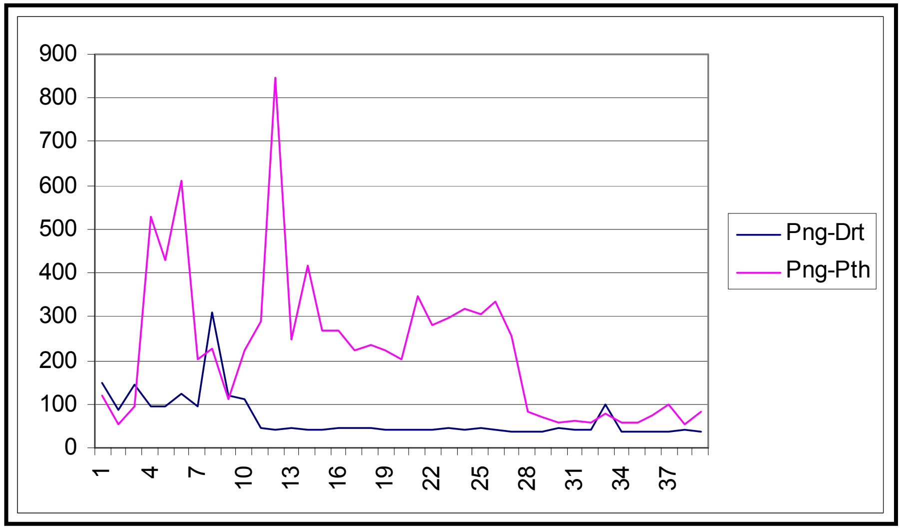
Using Ping Times to Predict Download Times

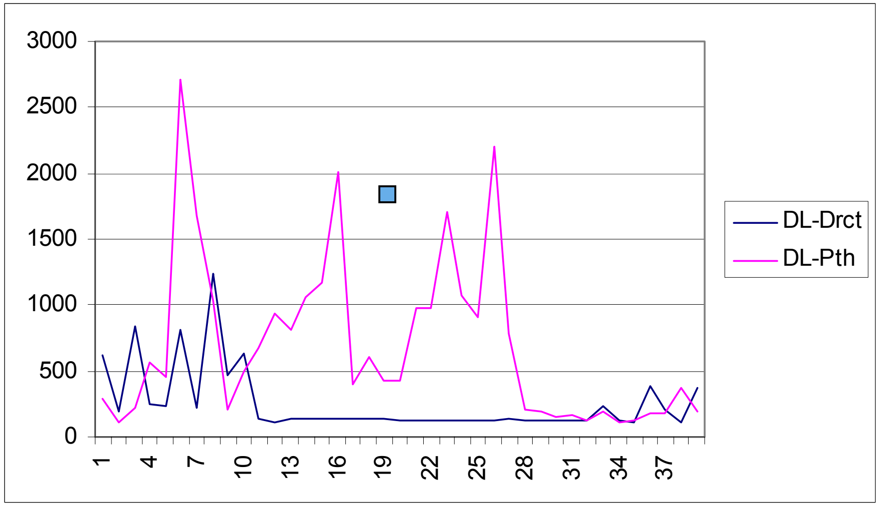


Why We Change Horses Midstream!

- What may look like a good path can go bad!
- In the following slide we will see an example of a path that goes bad.

A Bad path





TCP-Ping and download times every 5 minutes between two centers

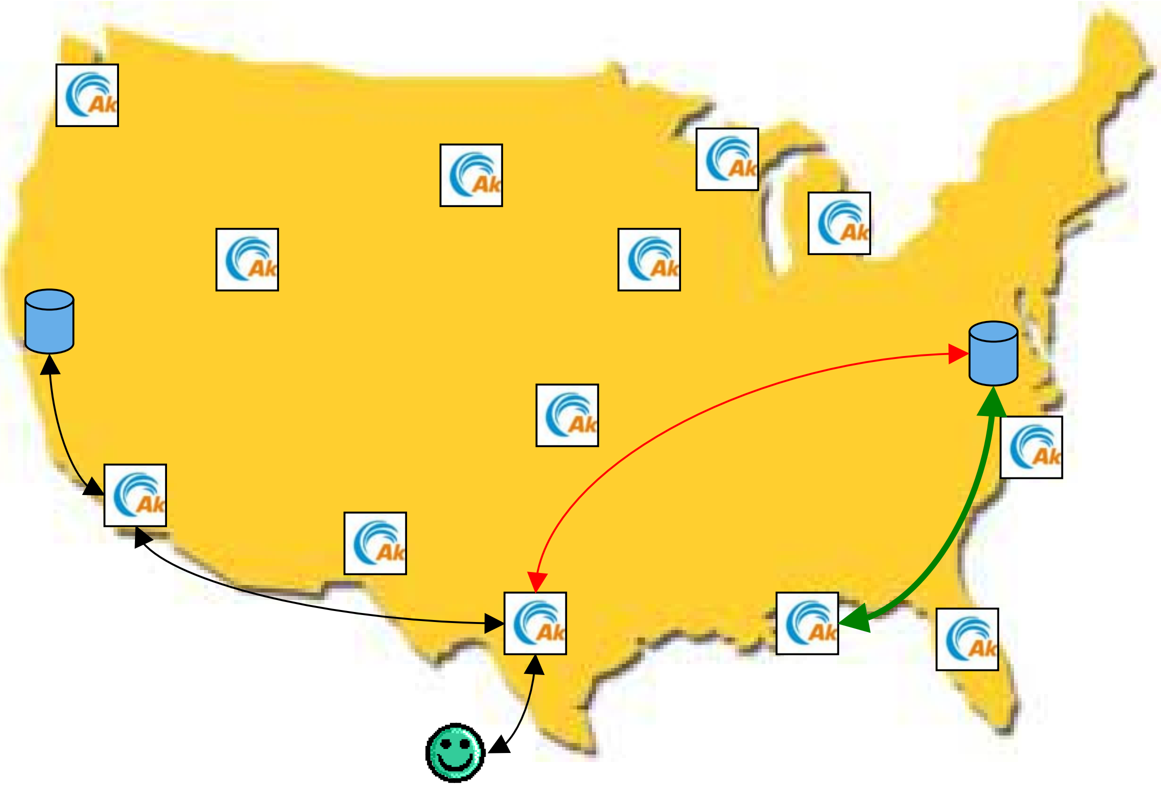
Predicting a Good Path

- Experiment: Downloads every 5 minutes over 3 different paths for 25 pairs of centers.
- Goal: Determine good algorithms that predict the best path.

A good predictor: Races

- Every so often, a **race** takes place:
  - Do three simultaneous downloads
  - Record the winner
  - The direct path gets a handicap
  - Record winner
  - Use that path for the near future.

Races Results



Race Results

Akarouting Requirements

1. Improve download times.
2. Route around network problems.
   - Respond quickly to changes in the network.
3. Fairness:
   - No client should have a worse experience using Akarouting.

Product: Akarouting Components

- The Global view: **MapMaker**

- The View from the edge: **Guide**

MapMaker

- Pings each mirror site for each content provider (every 15 min)
- Makes map tailored to each content provider
- **Strategy:** e.g.,
  - Yahoo-images
  - A CP with VA and CA Mirrors

Ping Data for the MapMaker

- Sources
  - Akanote:
    35 Akamai Data Centers to all Akamai DC’s
  - TPS (Trace Ping Server) 90 Akamai Data Centers to
    - All Akamai DC’s
    - 20 CP Data Centers (meta-data configurable)

MapMaker

- Determines distance between Centers and CP, based on ping data (age, loss and latency)
- Computes best one and two-hop paths to CP, from every Akamai center.
- Publishes best paths via DNS

Processing Ping Time and Loss

Goal is to compute effective-distance between DC's

- Magic Formula to compute effective-distance from ping latency and ping loss.

Processing Ping Time and Loss

Goal is to compute effective distance between DC's

- What to do with 100% ping loss?
  - The target machine is down but otherwise the DC is OK!
  - The Internet connection between the machines is down!
- If the machine is down we will discard ping data

  otherwise 100% loss will be charged,

  dramatically increasing the distance.
- We use rule:

  A machine is up at some time T if someone has received a packet after time T from the machine.

Selecting Middle Data Centers

The usable middle DC's are set on a per strategy basis.

- Our standard lists of middle DC's
- An explicit list

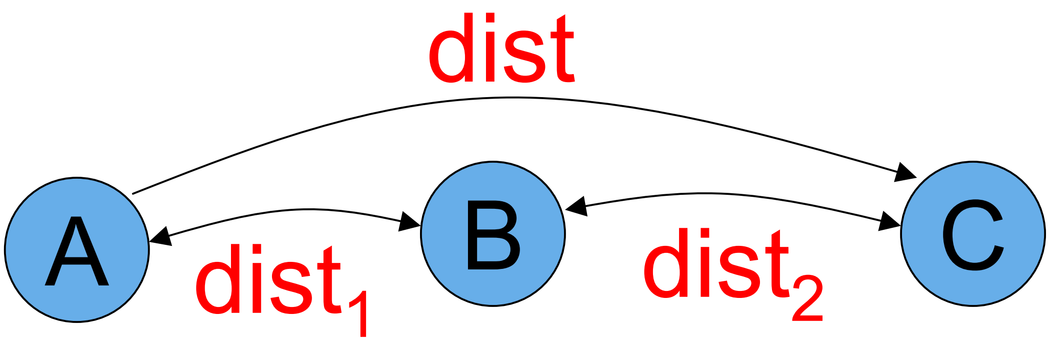
Middle DC's are removed:

- Using ghost info data, suspended DC's are eliminated.
- A cut off based on DC load (not used)

Diversity:

- Select paths with different Server-Providers

Computing short paths



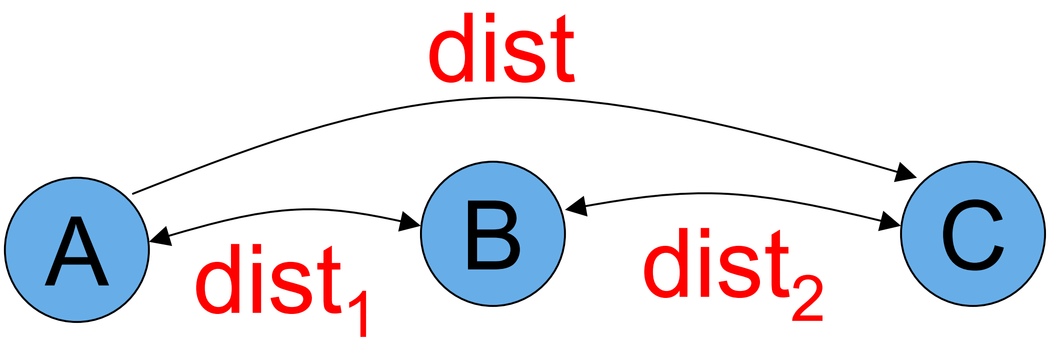
What should the distance dist equal?

$$L_1 = dist_1 + dist_2?$$

$$L_\infty = \max\{dist_1 + dist_2\}?$$

$$L_2 = (dist_1^2 + dist_2^2)^{1/2}$$

Computing short paths



What should the distance dist equal?

$$L_\infty \leq L_2 \leq \mathbf{L_{1.4}} \leq L_1$$

Which Map Should be used at a DNS?

Two important properties of a map:

- Amount of ping data in map.
- Freshest of data in map

Measure used:

Suppose a map is based on N samples with ages $a_1, \ldots, a_N$.

We define/use: **quality =** $\sum 1 / a_i$

Computing the Quality of a Map

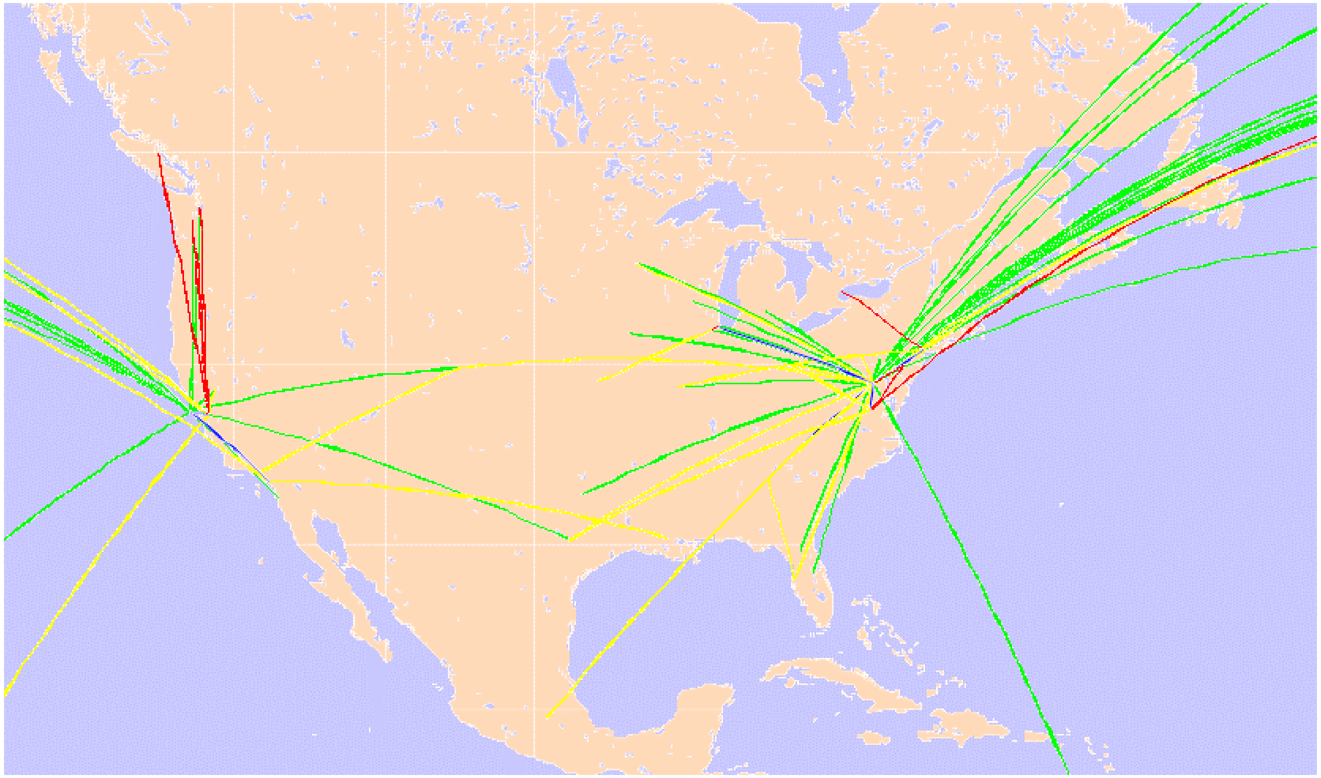
If the ages of the data samples are $a_1, \dots, a_N$ at the time the map was made and the map is now t units old.

The new quality is $\sum 1/(a_i + t)$.

We compute these qualities using a exponential bucketing scheme.

Each map is shipped with a vector $(t, b_1, \dots b_{50})$.

Yahoo: Jun 27th,1pm (ping times)



Green: Direct

Yellow/Blue >25% better

Red/Blue > 50% better

Guide - Route ranking

- MapMaker suggests 3 routes
  - CP (Best route to Yahoo)
  - P0 (Best middle region for tunneling)
  - P1 (Second best middle region)

- Routes are ordered by actual download times (races)

Guide – Are races allowed ?

- Not all content is raceable.
  - If not allowed, then we will need to perform a test download.
  - First client will use direct route if no data is available.

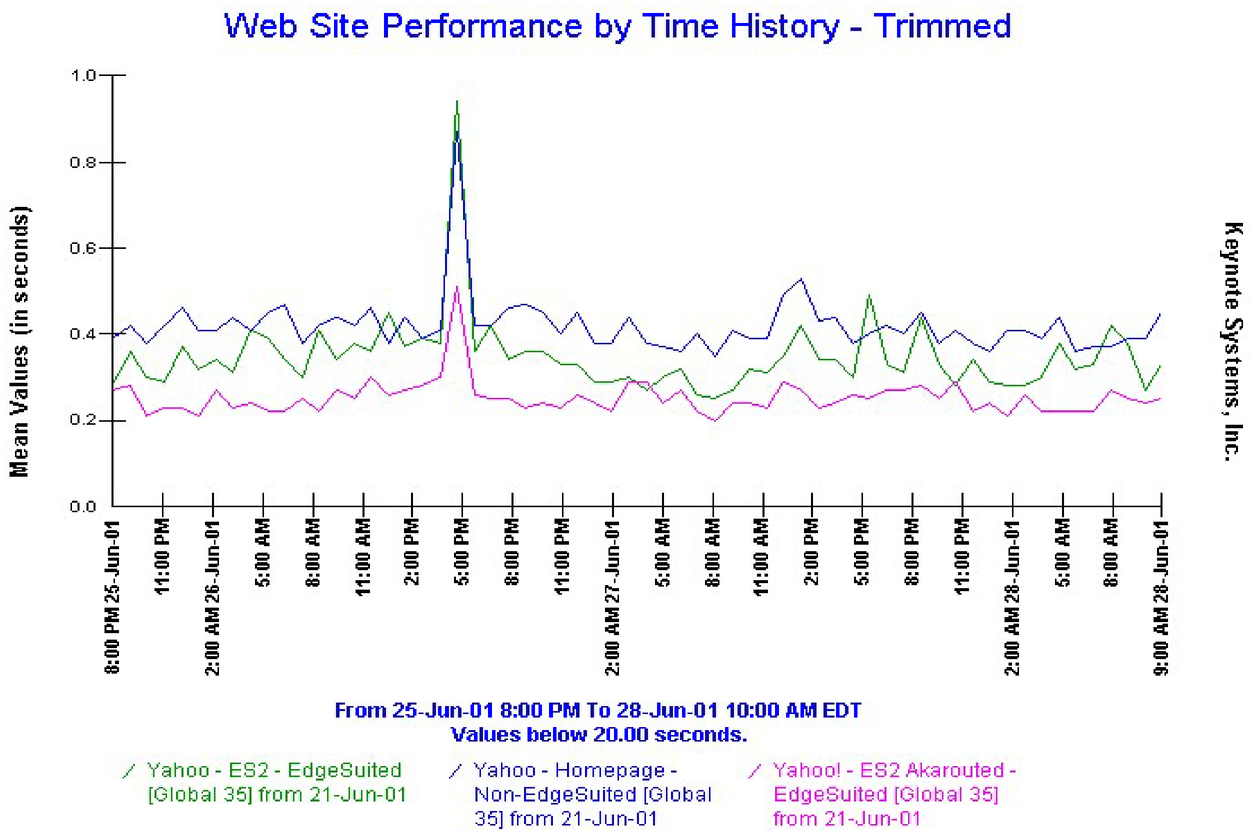
Guide – Are races allowed ?

- Races are better !
  - Race actually translates into the first request also achieving better performance.

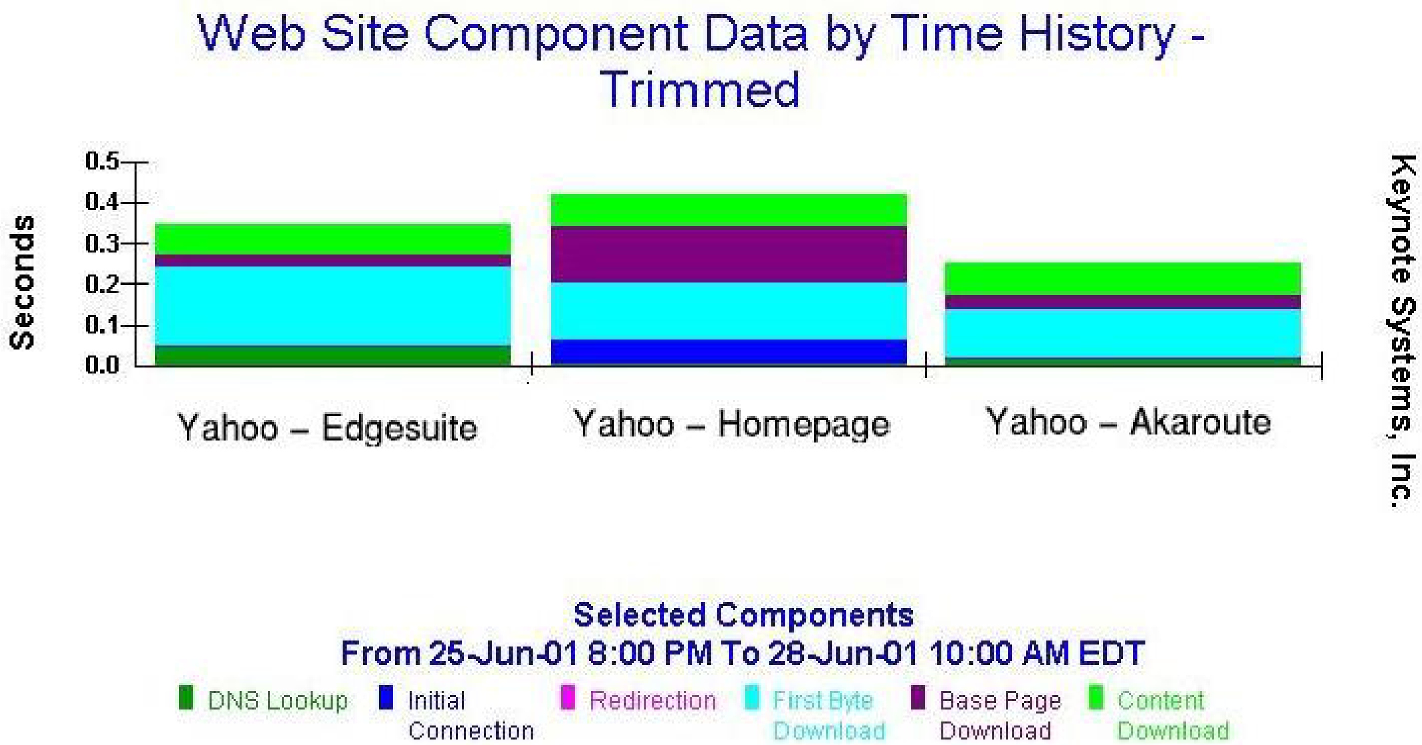
Keynote measurements

- 3 downloads: images are cached using FF.
  - Yahoo-Homepage: basic FreeFlow
  - Edgesuite – uses ESI
  - Edgesuite/Akaroute – also uses ESI

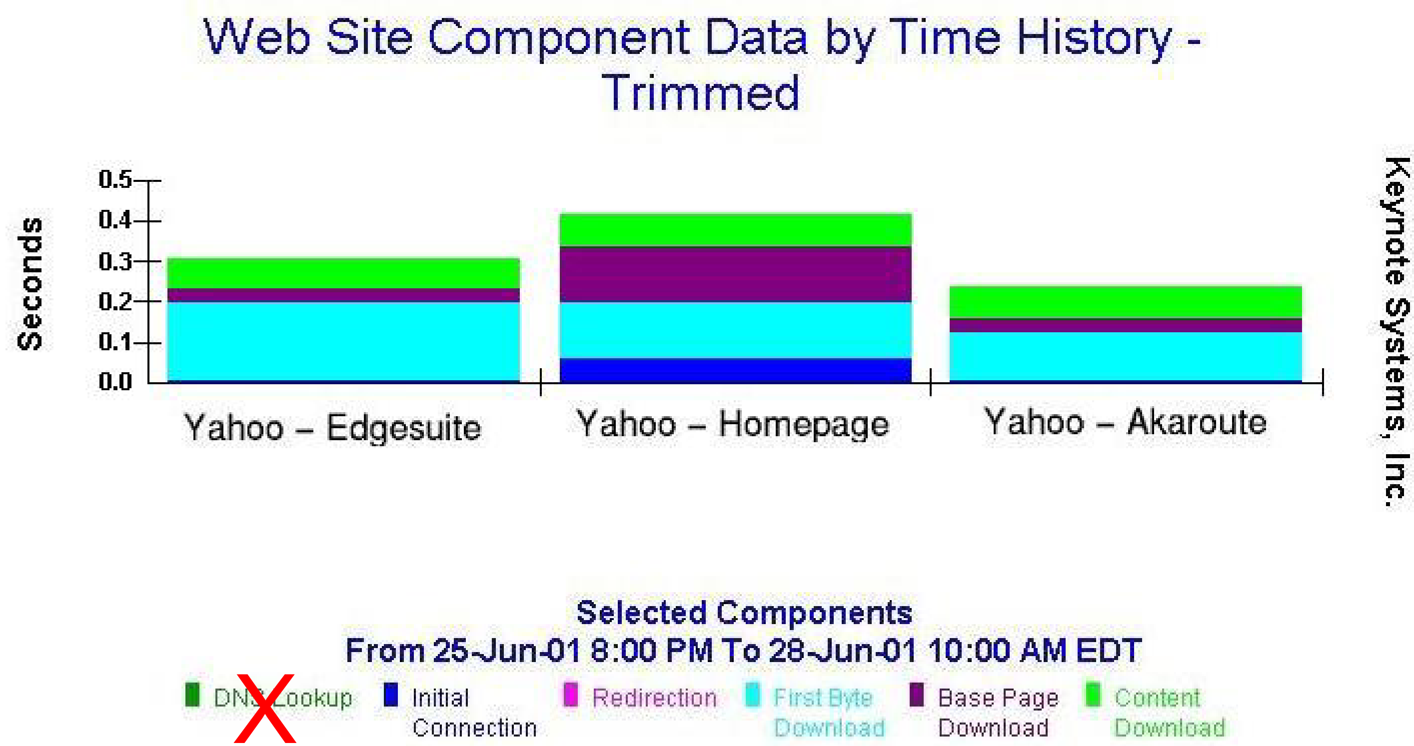
Keynote time series



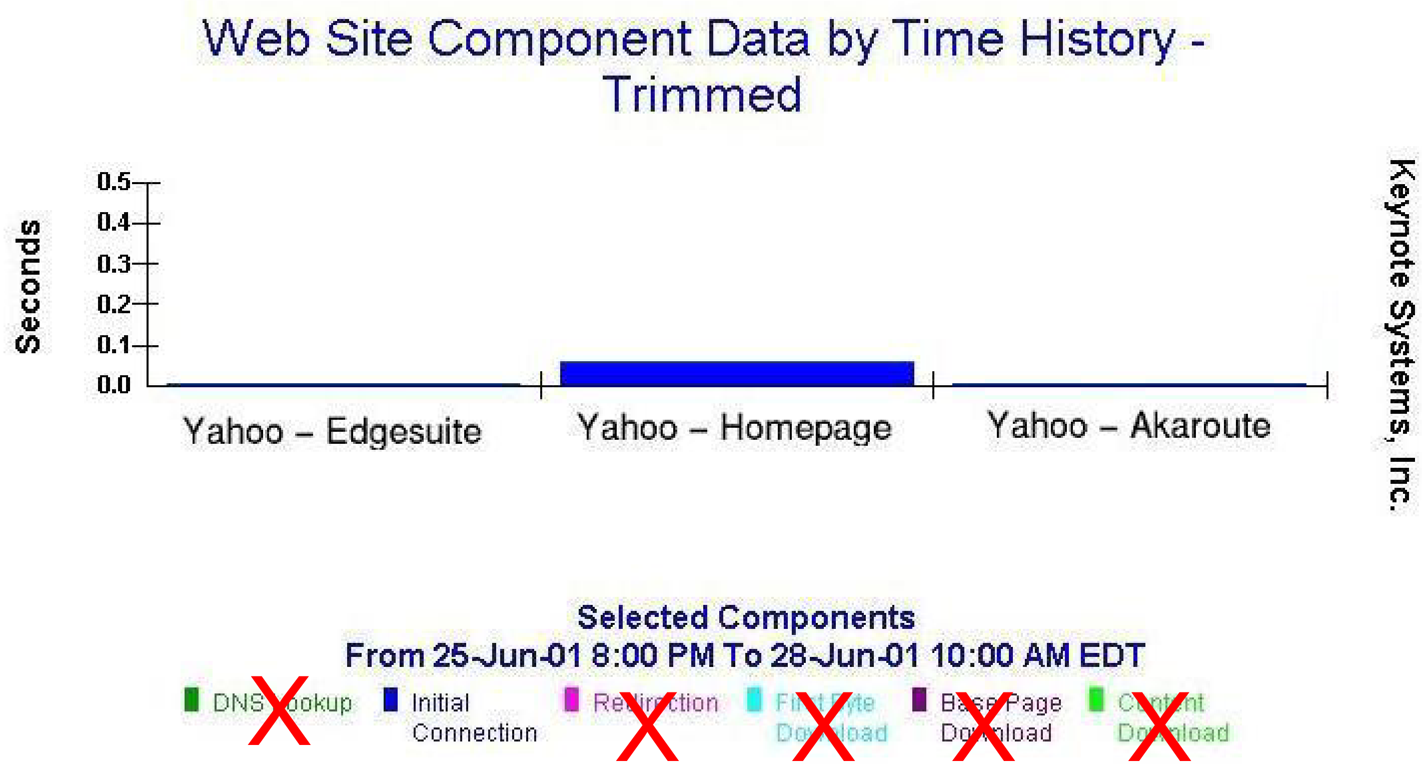
Keynote: component times



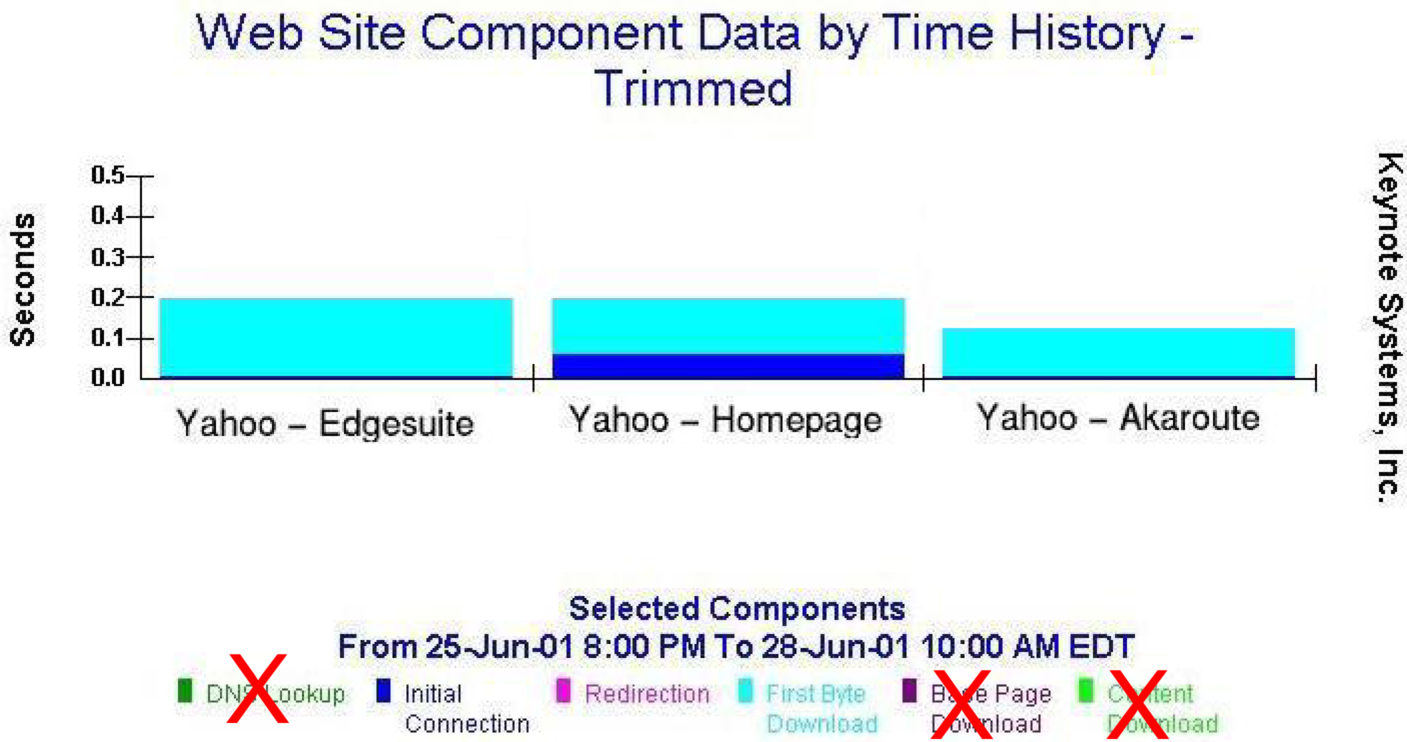
Keynote: component times



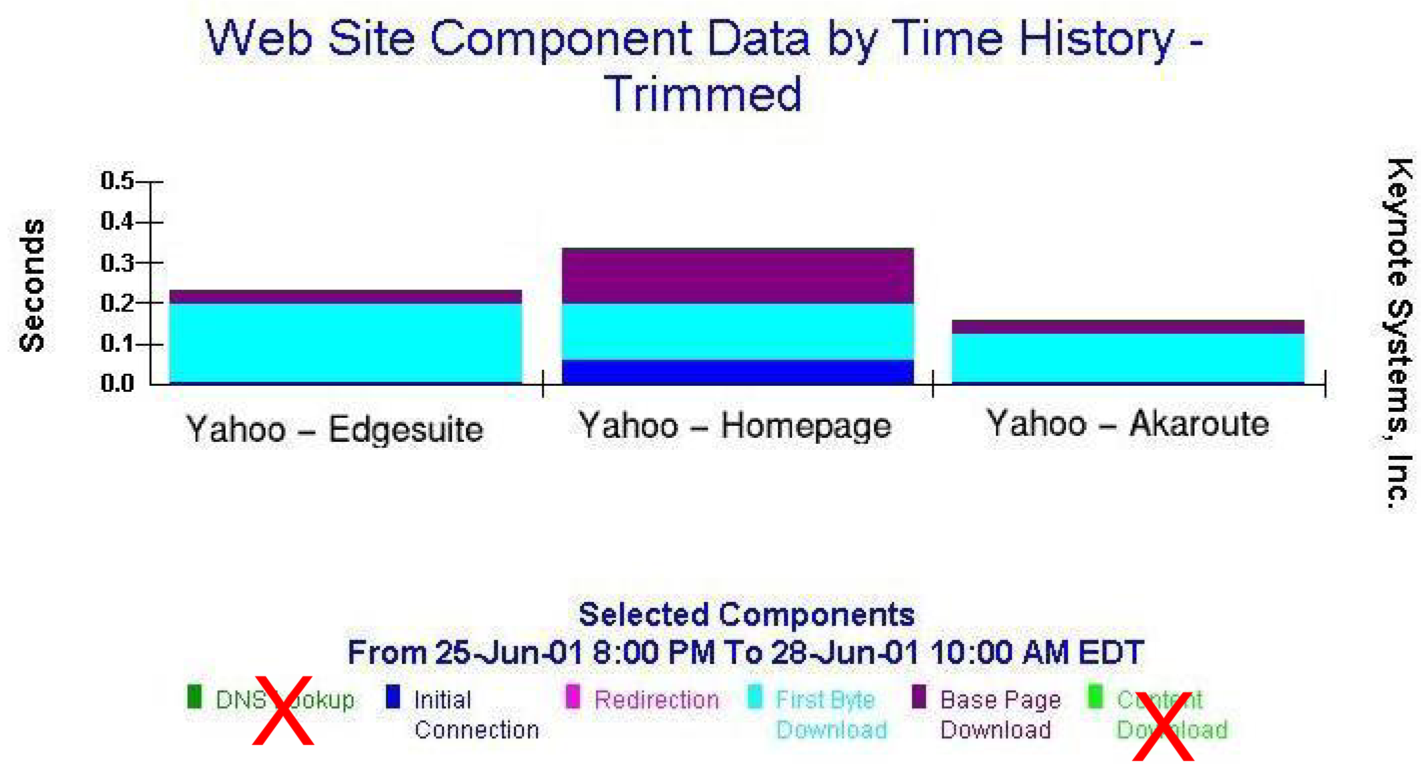
Keynote: component times



Keynote: component times



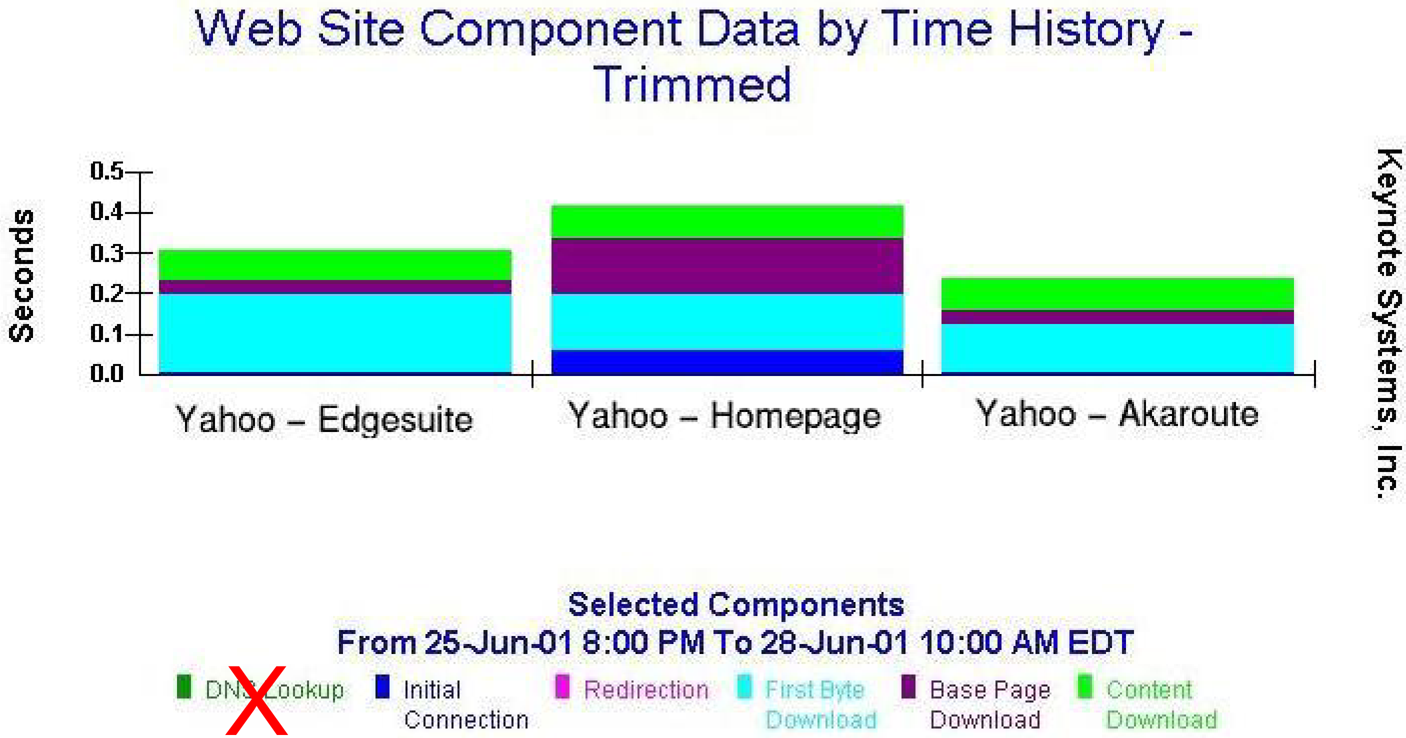
Keynote: component times



Results

- Ignoring embedded (cacheable) content:
  - EdgeSuite/ESI was 31% faster than Direct.
  - Akarouting/ESI was 55% faster than Direct.

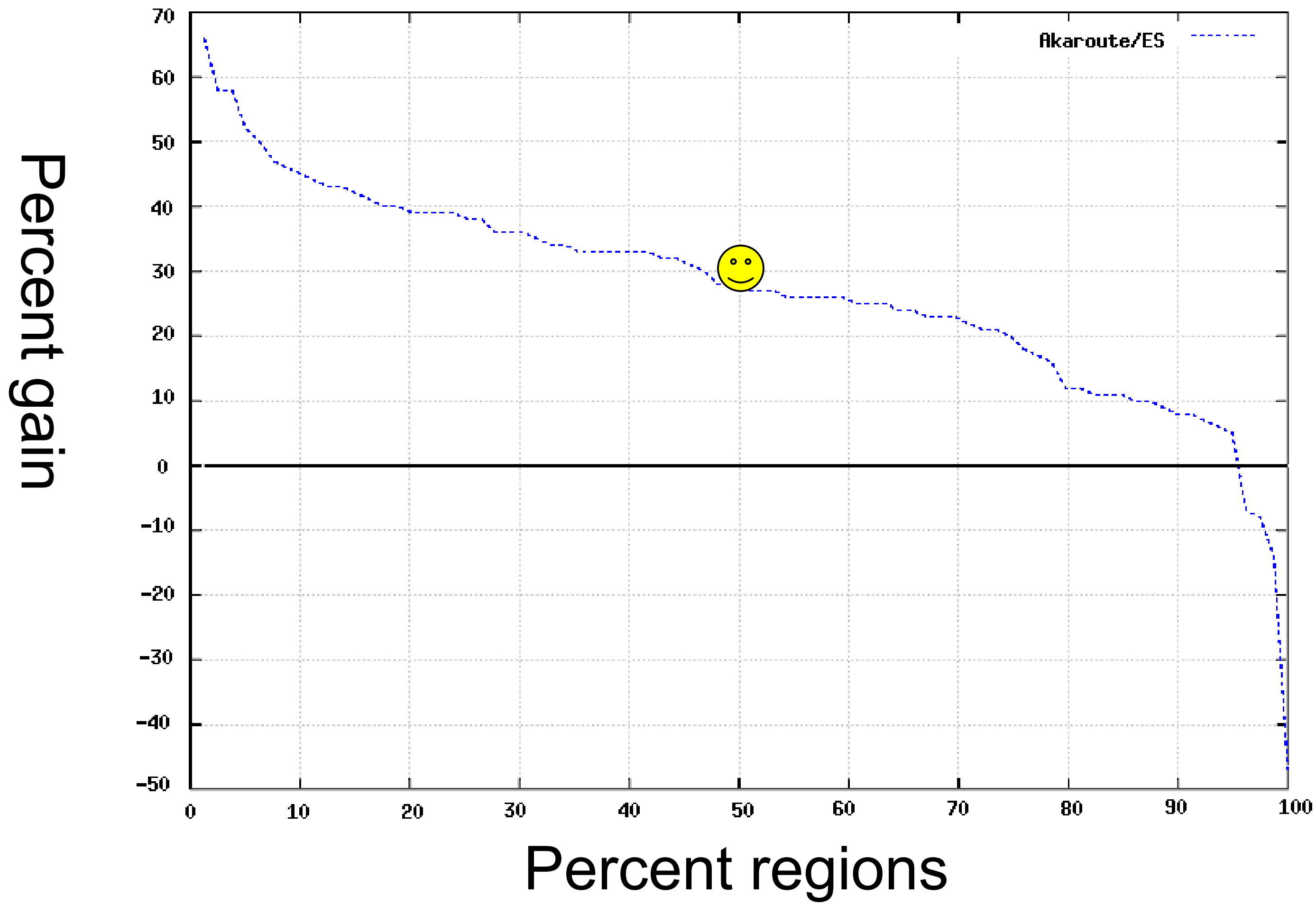
Keynote: component times



Results

- ESI improves performance.
  - EdgeSuite was 27% faster than basic FreeFlow.
  - Akarouting was 44% faster than basic FreeFlow.

Keynote: per agent averages



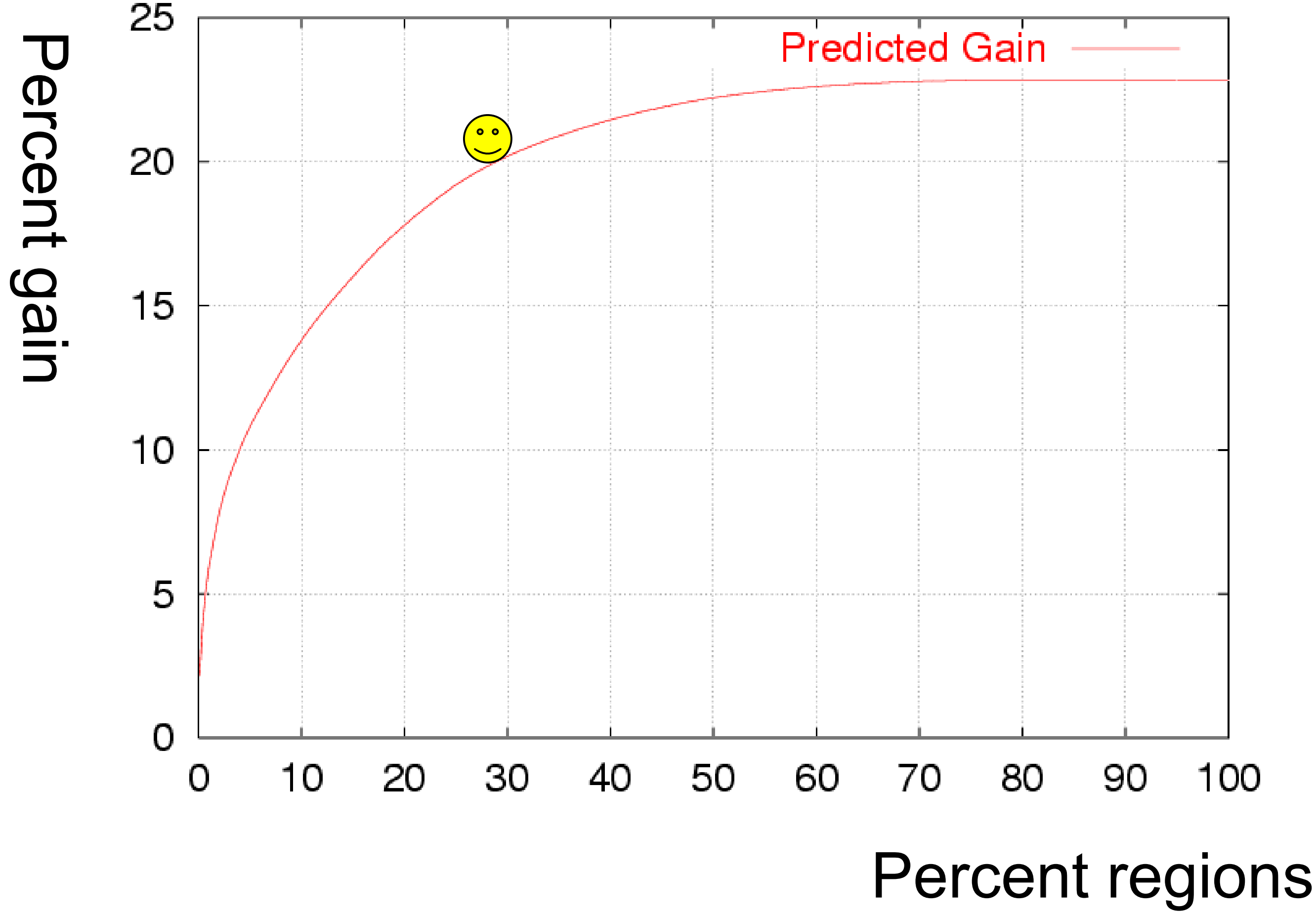
What is the cost?

- Akarouting should provide most of the speed benefits with 25-30% of regions going indirect.
- What percentage of traffic is dynamic, specially with ESI ? (2% for front page)
- Reliability is crucial!

Possible Application for Akarouting

- SSH (Original)
- Streaming Network
- Akamai Powered Web browser
- Voice over IP
- VPN
- EdgeSuite

Two-hop x direct ping times



Gain

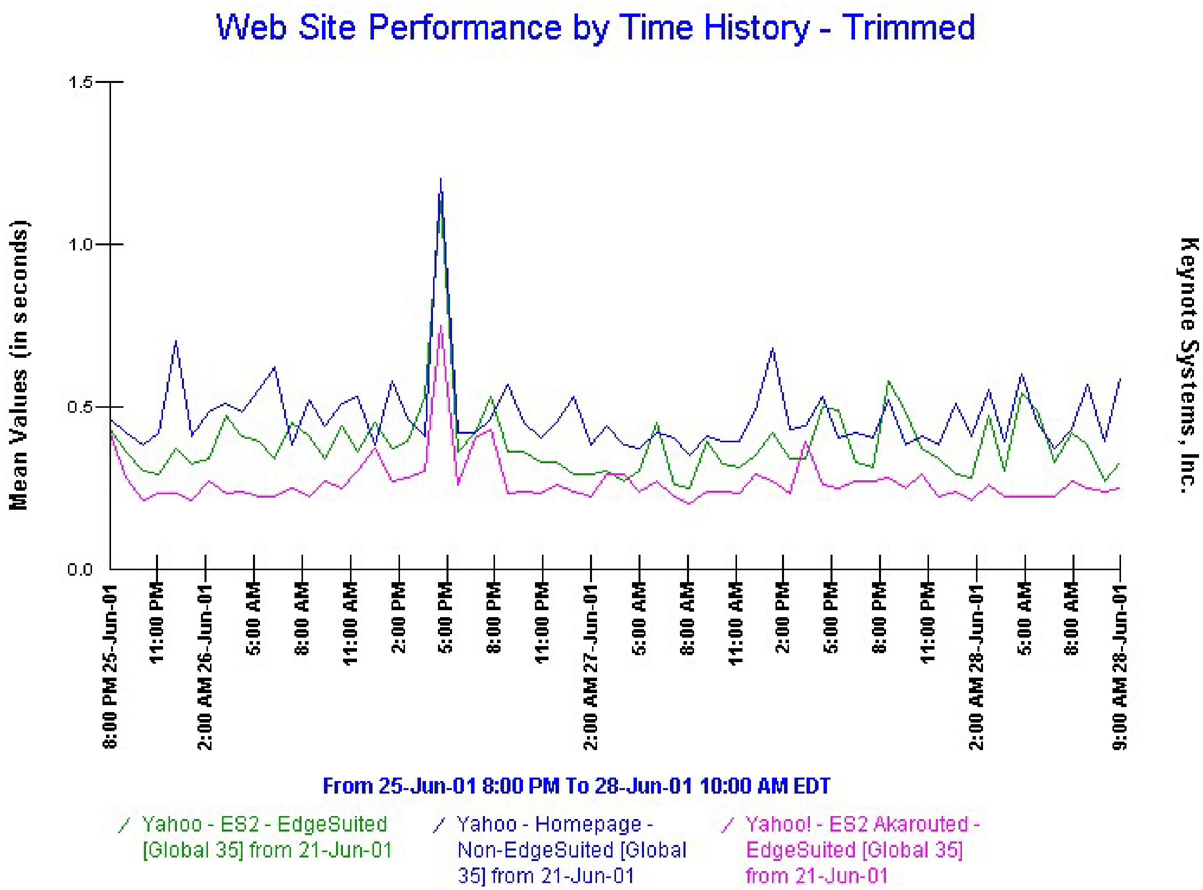
- Direct average-ping-time:
  - Average over all Akamai regions of minimum ping time to either Yahoo-east or Yahoo-west.

- Two-hop average-ping-time:
  - Average over all Akamai regions of minimum two-hop ping time to either Yahoo-east or west.

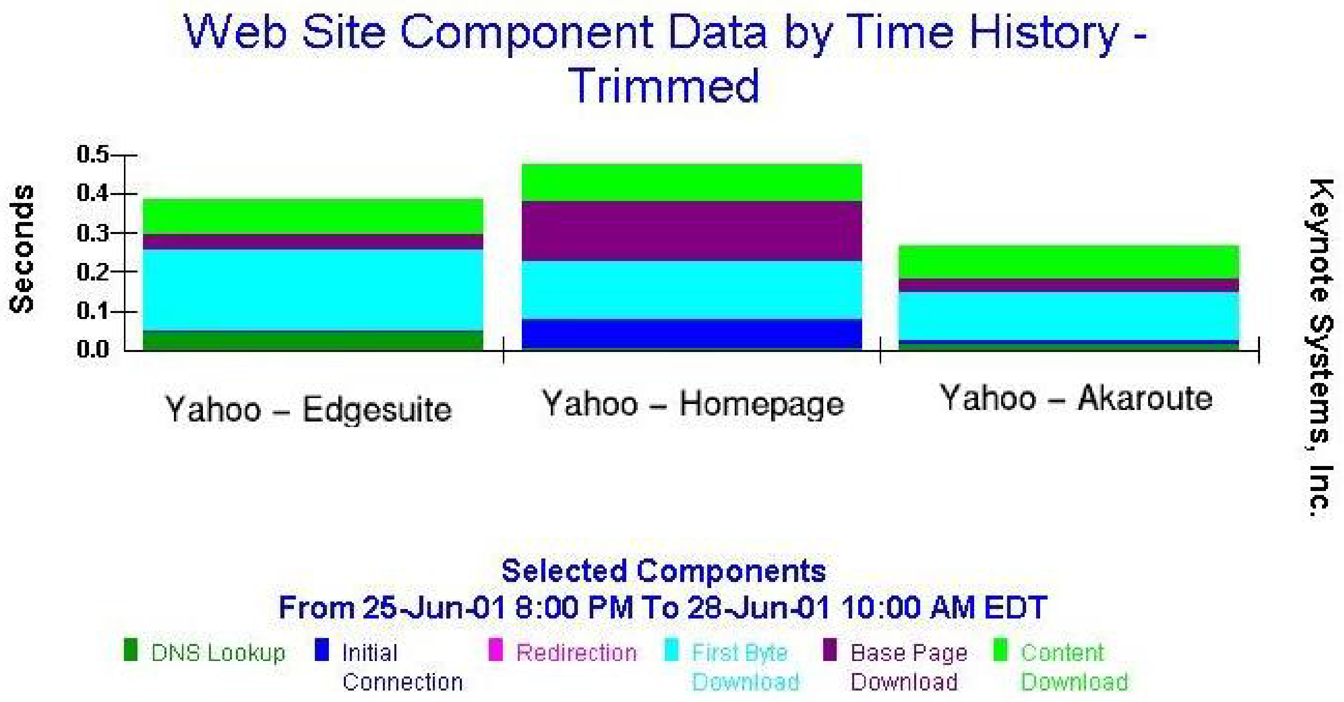
Keynote: component times

All Time Ranges	Yahoo - ES2 - EdgeSuited		Yahoo - Homepage - Non-EdgeSuited		Yahoo! - ES2 Akarouted - EdgeSuited	
Component	Avg. Time (secs.)	%	Avg. Time (secs.)	%	Avg. Time (secs.)	%
DNS Lookup	.04	12.06			.01	5.94
Initial Connection			.06	14.41		
Redirection						
First Byte Download	.19	55.00	.14	32.92	.12	46.86
Base Page Download	.03	9.86	.14	32.95	.03	13.54
Content Download	.07	21.37	.08	18.55	.08	31.29
Count	17991		19234		18052	
Average Total Bytes	28529		29118		28696	
Average Bytes/Sec.	82181.29		69288.49		113945.60	
Total Measurement Time	.35		.42		.25	
Trimmed Data Points	19		34		11	

Keynote results: time history



Keynote results: components



Keynote results: components

All Time Ranges	Yahoo – ES2 – EdgeSuited		Yahoo – Homepage – Non-EdgeSuited		Yahoo! – ES2 Akarouted – EdgeSuited	
Component	Avg. Time (secs.)	%	Avg. Time (secs.)	%	Avg. Time (secs.)	%
DNS Lookup	.04	11.03			.01	5.59
Initial Connection			.07	15.77		
Redirection						
First Byte Download	.21	54.05	.15	31.31	.13	47.25
Base Page Download	.04	9.50	.15	32.60	.03	12.74
Content Download	.09	23.58	.09	19.29	.08	31.31
Count	18009		19265		18059	
Average Total Bytes	28528		29116		28696	
Average Bytes/Sec.	73892.51		61792.42		107207.25	
Total Measurement Time	.39		.47		.27	
Trimmed Data Points	1		3		4	

Constraints

- Cost of tunneling:
  - Configurable settings allows trading off speed and indirect traffic.
- Cost of quality checks:
  - Small absolute cost (for low usage).
  - Small percentage of traffic from the provider (for larger loads).

MapMaker

- Prunes route choices based on global view.
- Short paths from our edge regions to CPs are computed every 15 to 30 minutes.

How are routes chosen?

- MapMaker suggests routes.
  - The Guide ranks routes.
  - Is able to respond faster.
  - Based on real download times.

Default settings

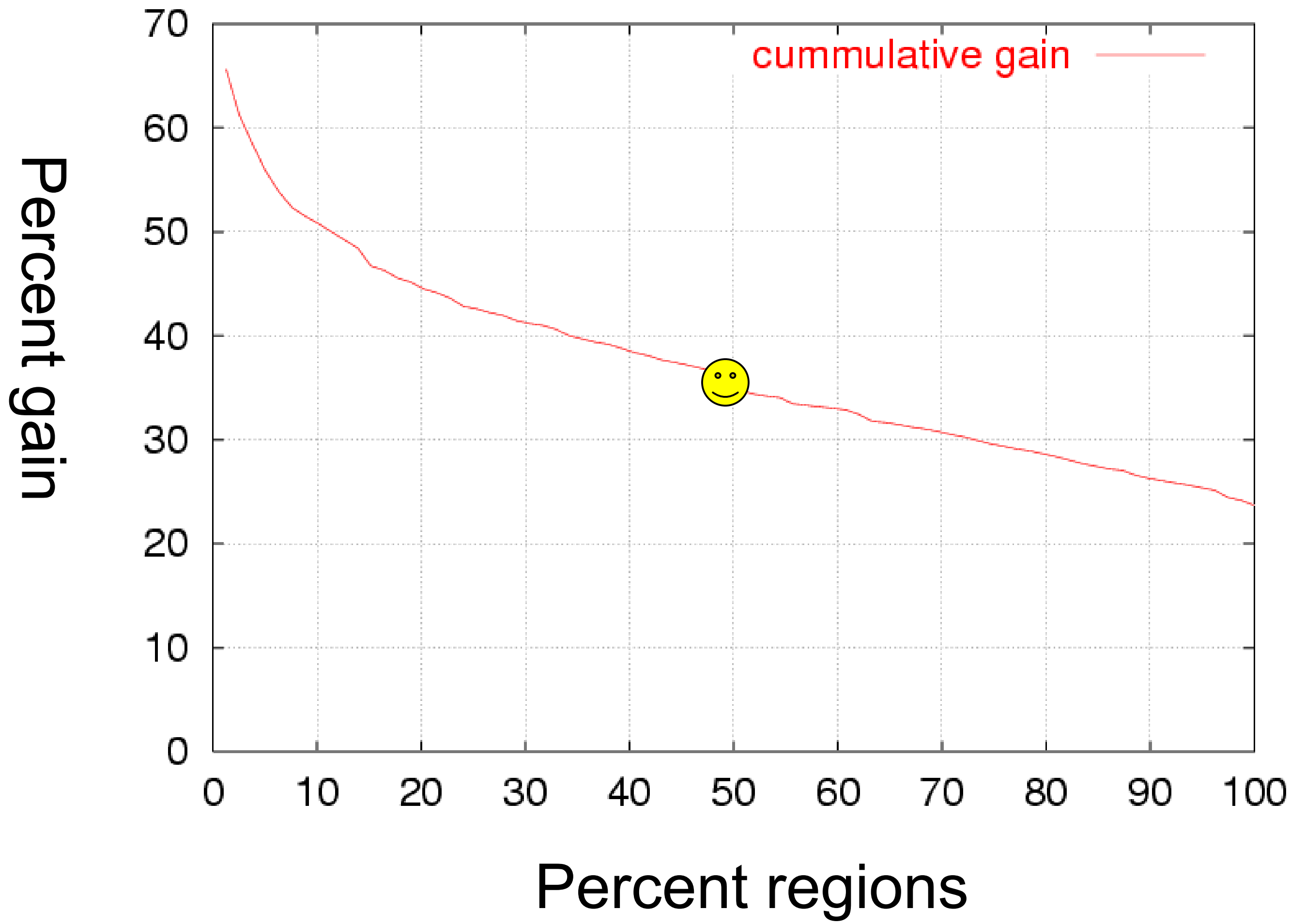
TimeBetweenRaces = 5 - 15 min

TimeBetweenTests = 5 - 15min

AdditiveThreshold = 30ms

(about 25% of regions go indirect)

Keynote: per agent averages



Guide – Route ranking

Guide needs fresh download data.

– Every so often (configurable) the ghost does simultaneous downloads (races), to locally rank the routes.

Akarouting

- To be used in conjunction with Edgesuite
  - Faster, more reliable downloads
  - Works by tunneling content through intermediate regions when necessary