Web Performance Characterization

Anja Feldmann

AT&T Labs-Research; Florham Park, NJ
http://www.research.att.com/~anja/
anja@research.att.com

Why care

Web is the dominant application

- ~ 72% flows, ~ 55% packets, ~ 56% bytes

Web is based on other protocols

- TCP, DNS

Network mechanism that impact the Web

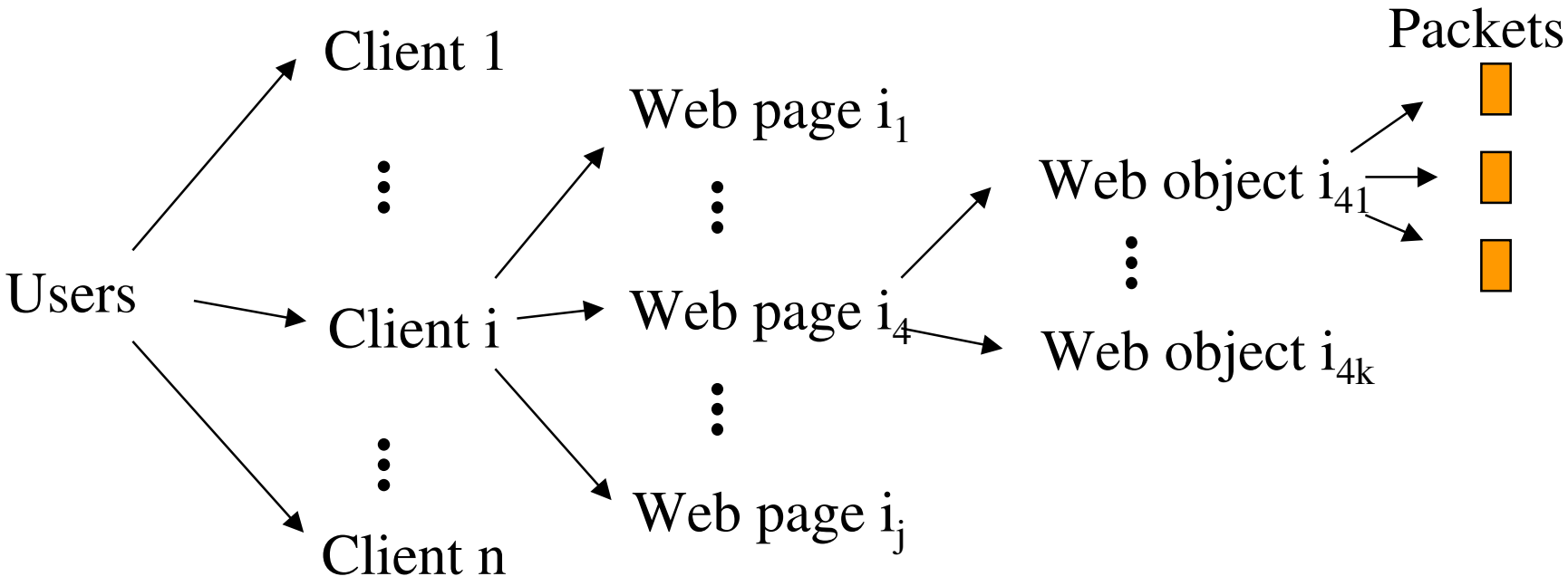
- Routing, caching

Web is undergoing changes

- HTTP 1.0 -> HTTP 1.1, multimedia

How to...

Web: high variability at all levels



Web pages per session

Images per page

Resource size (bytes, packets)

Think time (time between clicks)

Web and TCP

TCP connection

- Transfers one or more Web objects

Parallel TCP connections

- Simultaneous rendering of imbedded images
- Improved download speed
- Independent state - fairness?

Aborted Web downloads

- Click ahead, abort button
- Terminates the TCP connection

Web workload

Arrival process

- Web sessions: Poisson process
- TCP connections: bursty (self-similar)
- Packets: bursty (self-similar)

Persistent TCP connection

- Reuse of TCP connections is increasing
- Reduce burstiness of TCP connection arrivals
- Does not eliminate burstiness of TCP arrivals

Web: mice vs. elephants

Large number of small transfers

- TCP connection setup time is significant
- TCP connections stay in slow start

Small transfers introduce burstiness

- Aggregate traffic acts unconstrained
- Aggravated by bursty request arrival process

Small number of large transfers

- Carry large number of bytes
- Dynamics driven by "background" traffic

Workload models need to include variability

- **SURGE [BC99], NS2 workload [FGHW99]**

Web and DNS

Hostname to IP address mapping

- One to one; one to many; many to one

Resolution of hostname in URL

- Client, proxy, or none (page in client cache)

Server reverse lookup

- Logging, authentification, targeted advertising

Hosting

- Load balancing, content distribution

Impact

- **Lookup is larger portion of delay**
- **DNS cache misses are more likely**

Network mechanism that impact the Web

Cache placement

- Near client, middle of network, near server

Directing request to the cache

- Transparent, explicit, reverse

Packets may not follow a single path

- Transparent caching -> denial of service

Reverse path differs from forward path

- Caching changes traffic patterns

Peering with Web hosting companies

- Use BGP to control traffic entry point

Effectiveness of caching

	1997	1999	
Cookie present	30%	47 %	51 %
URL with '?'	10 %	16 %	21 %
Server Cache-Control	2 %	8 %	13 %
Client Cache-Control	8 %	2 %	2 %
URL with 'cgi-bin'	5 %	5 %	5 %
Not GET or HEAD	2 %	2 %	2 %
Authorization present	2 %	1 %	1 %

Limited cache hit rates, e.g. modem traffic:

- 1997 54 % (35 % without cookies)
- 1999 49% (27 % without cookies)

HTTP 1.0 vs. HTTP 1.1

		Requests	Responses
Nov 1999	1.1 (1.0)	71% (29%)	71% (29%)
Jul 1998	1.1 (1.0)	33% (67%)	52% (47%)

HTTP response headers

	HTTP 1.1	(1.0)
connection	73%	(39%)
keep-alive	32%	(1%)
accept-ranges	58%	(20%)
etag	64%	(17%)
last-modified	60%	(42%)
cache-control	12%	(14%)

request headers

	HTTP 1.1	(1.0)
host	100%	(98%)
connection	99%	(91%)
if-mod	17%	(21%)
if-none-match	11%	(.5%)
extension	10%	(11%)
range, if-range	1%	(1%)

Web characterization

Caveats

- Metrics are hard to define and measure
- Measurement techniques
- Datasets

  Integrity? Completeness?

  Representative? Diversity?

W3C Web characterization working group

WG: http://www.w3.org/WCA/

Metrics: http://www.w3.org/1999/05/WCA-terms/

Data repository: http://purl.org/net/repository

The Web

Dominant application:

- Wednesday 11 Nov. 3:50pm to 4:50pm
  - ~ 72% of flows to/from an ISP
  - ~ 55% of packets, ~ 56% of bytes
- Wednesday 11 Nov. 3:50am to 4:50am
  - ~ 60 % of flows, ~ 48 % of packets, ~ 49 % of bytes

Secure Web:

- Afternoon: 4% of flows, wee hours: 1% of flows

DNS:

- Afternoon: 9% of flows, wee hours: 13% of flows