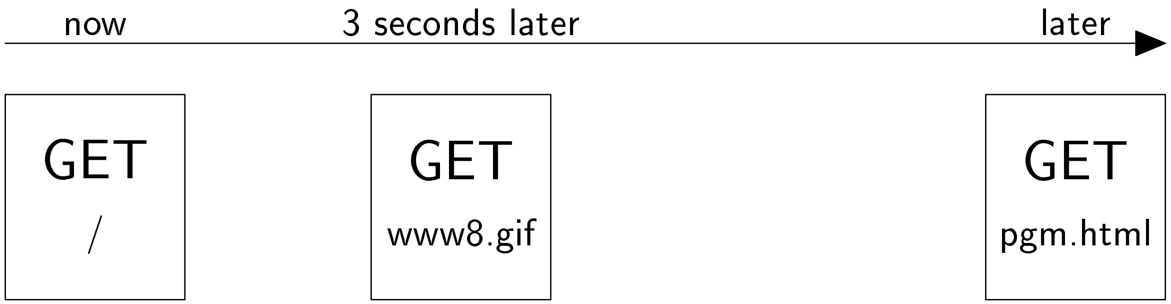
Managing TCP Connections Under Persistent HTTP

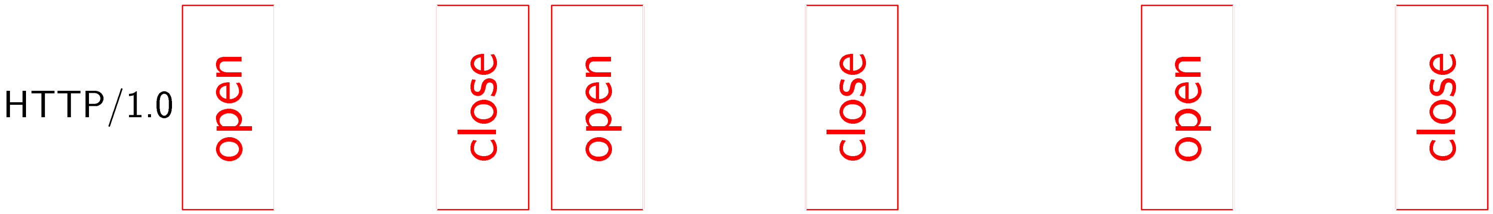
Edith Cohen Haim Kaplan
AT&T Labs—Research
{edith,hkl}@research.att.com

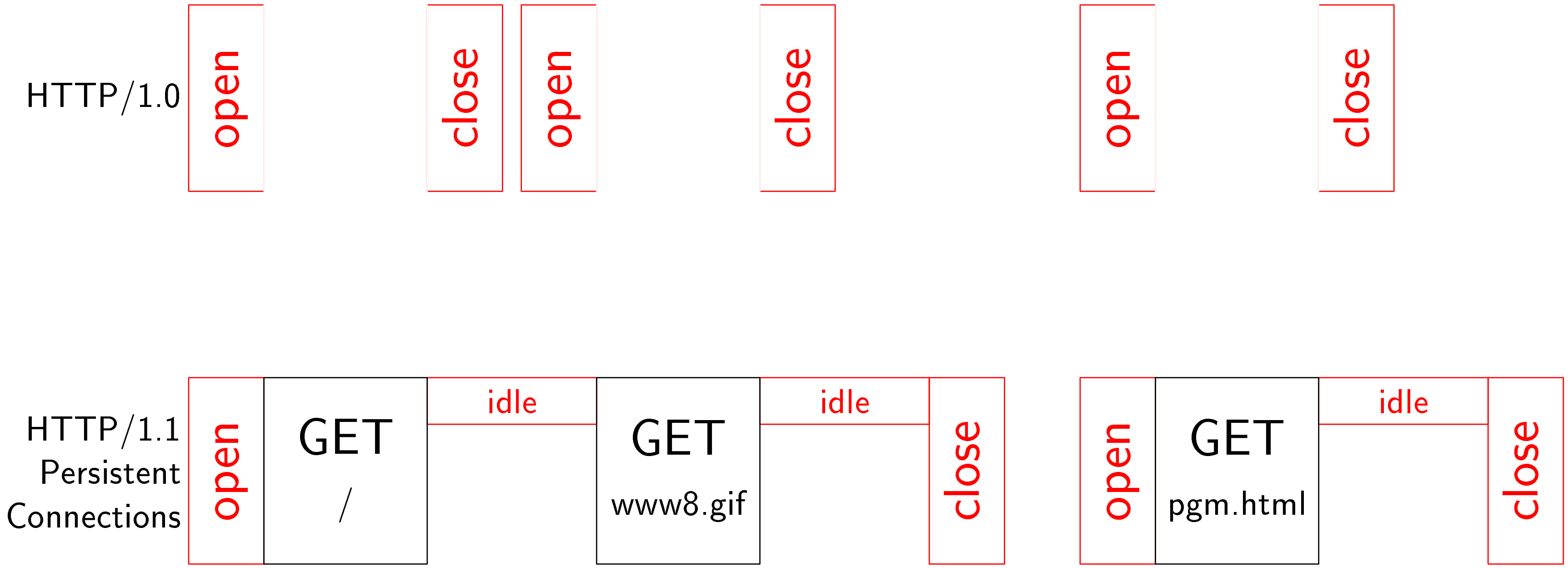
Jeffrey D. Oldham
Stanford University
oldham@cs.stanford.edu

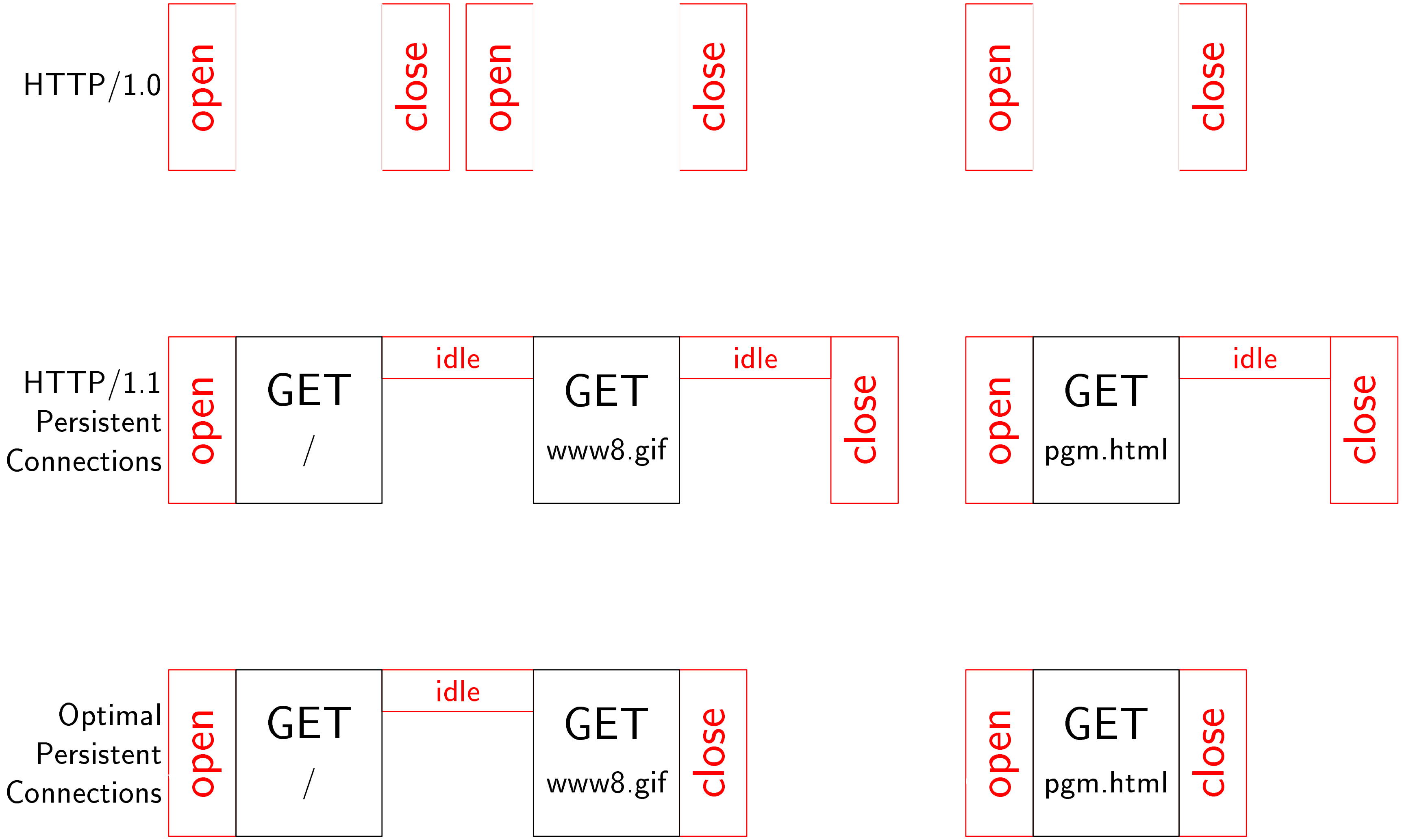
13 May 1999

An Example of Persistent HTTP Connections




HTTP/1.0
open
close
open
close
open
close


HTTP/1.0
open
close
open
close
open
close
HTTP/1.1
Persistent
Connections
open
GET
/
idle
GET
www8.gif
idle
close
open
GET
pgm.html
idle
close


HTTP/1.0
open
close
open
close
open
close
HTTP/1.1
Persistent
Connections
open
GET
/
idle
GET
www8.gif
idle
close
open
GET
pgm.html
idle
close
Optimal
Persistent
Connections
open
GET
/
idle
GET
www8.gif
close
open
GET
pgm.html
close

Persistent HTTP Advantages and Disadvantages

Advantages for User: faster response time

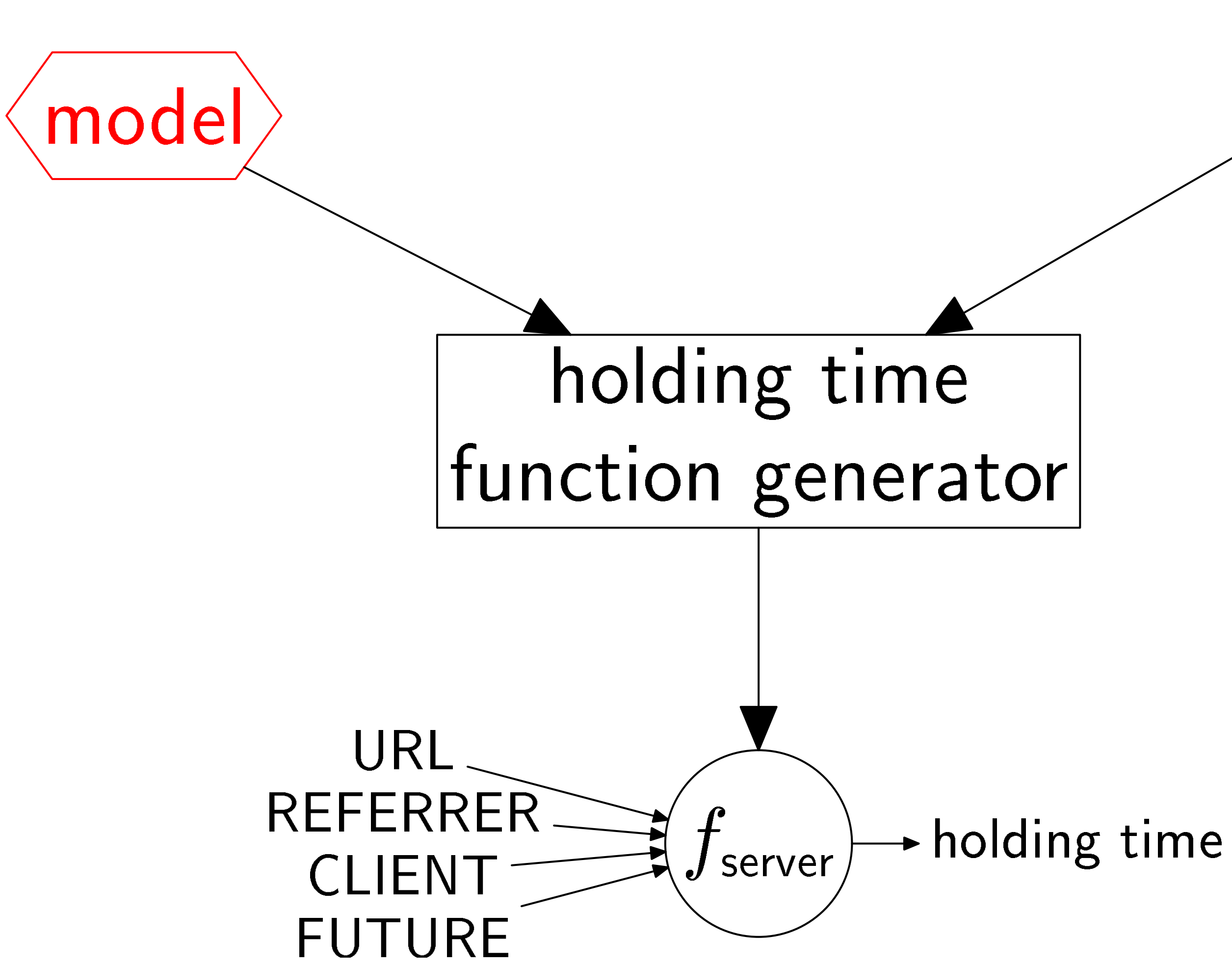
- fewer TCP handshakes
- can pipeline multiple requests
- TCP can better determine network congestion

Advantages for Server: less computation

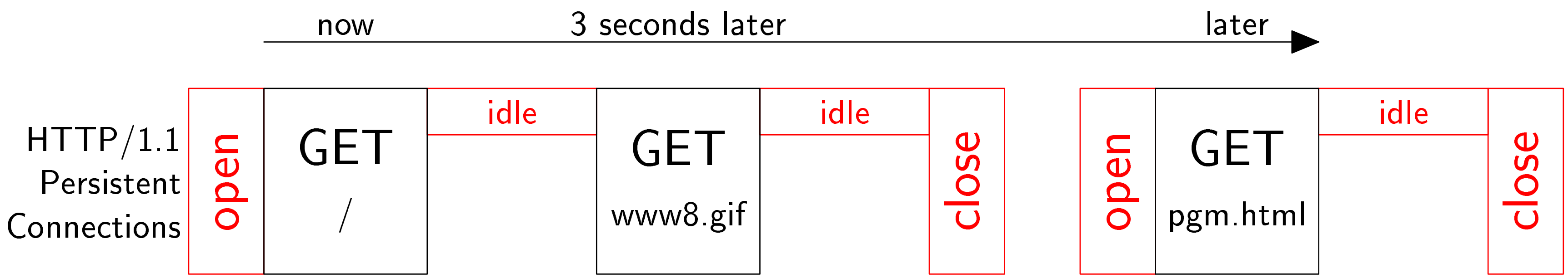
- less CPU time opening and closing TCP connections
- fewer simultaneous TCP control blocks

Disadvantages for Server: more idle open TCP connections

- TCP connections require memory
- many idle connections slow servers [Banga, Druschel, Mogul 1999]


model
server
log
holding time
function generator
URL
REFERRER
CLIENT
FUTURE
f_{server}
holding time

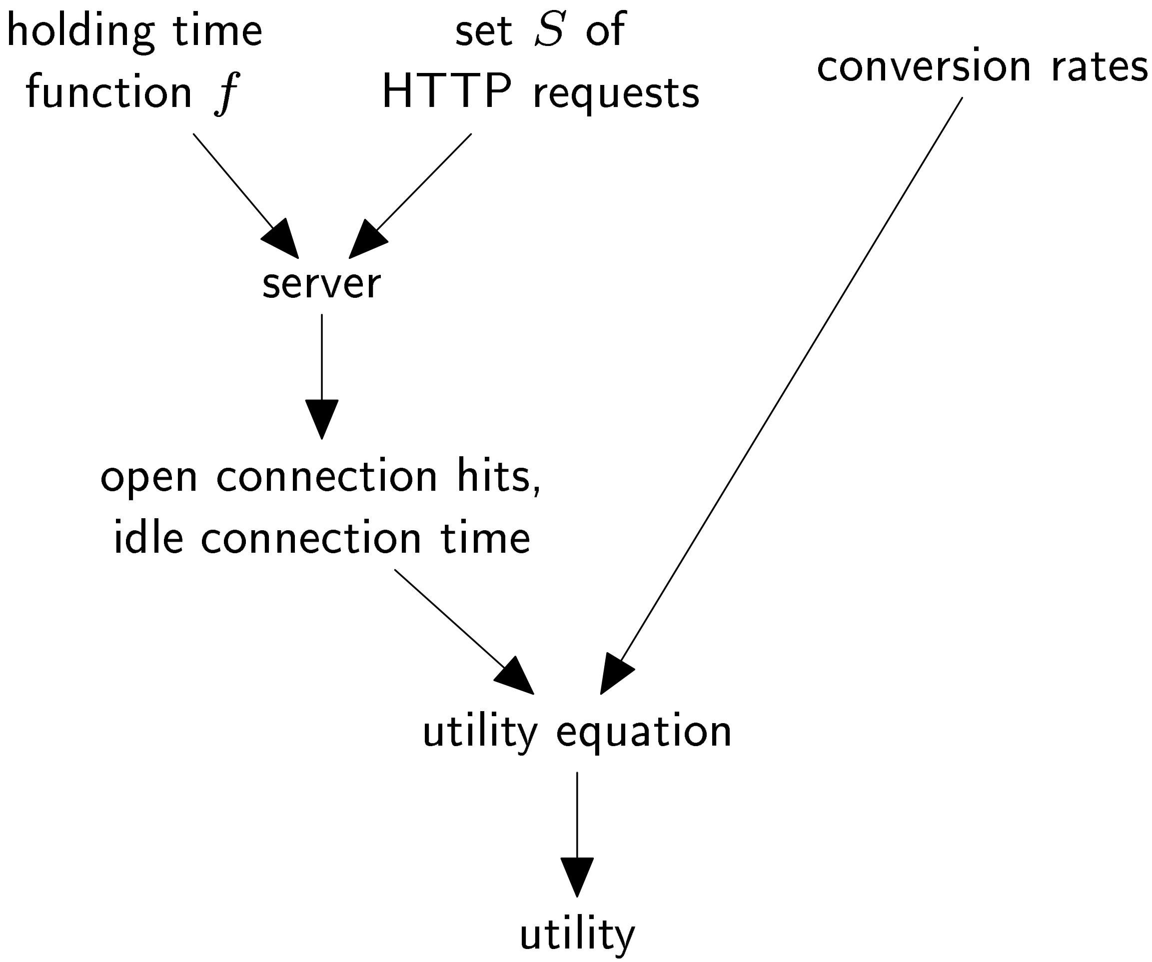
Holding Time Model



What is a good choice of holding times?

Utility Model

Goal: Maximize server's utility.



Utility Equation

$$
\begin{aligned}
\text{utility} &= \text{revenue} - \text{cost} \\
\text{revenue} &= \text{hits} \cdot C_{h \to \$} + \text{ open-hits} \cdot C_{oh \to \$} \\
\text{cost} &= (\text{hits} - \text{open-hits}) \cdot \text{TCP setup} \cdot C_{TCP \to \$} + \text{ idle-time} \cdot C_{it \to \$}
\end{aligned}
$$

Conversion Rates

$$
\begin{aligned}
C_{h \to \$} &= \text{value of HTTP request} \\
C_{oh \to \$} &= \text{value of HTTP request to open connection} \\
C_{TCP \to \$} &= \text{cost of establishing TCP connection} \\
C_{it \to \$} &= \text{cost of idle connection}
\end{aligned}
$$

$$\text{utility} \approx C + C_o \cdot \ \text{open-hits} - C_i \cdot \ \text{idle-time}$$

Utility Equation

Which terms depend on holding time function?

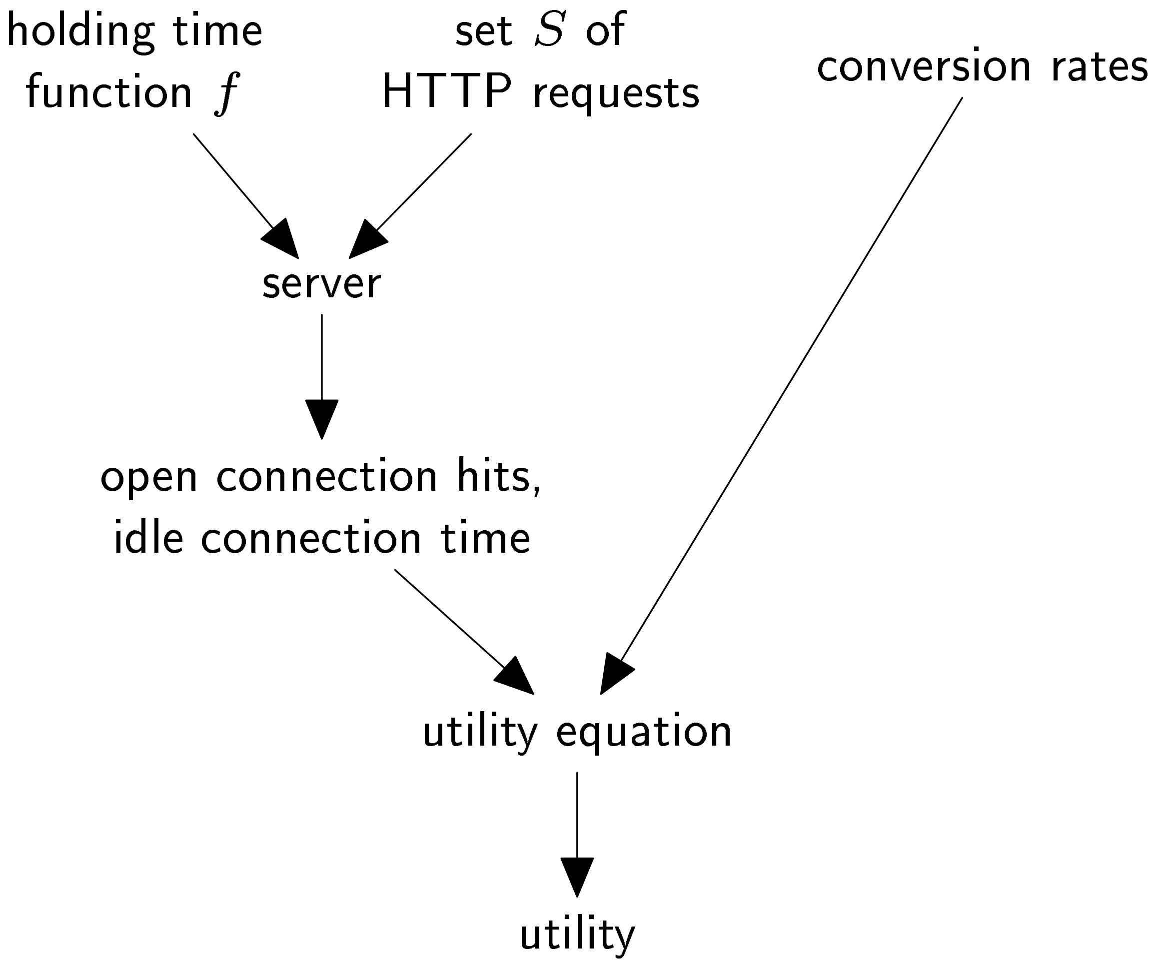
$$\begin{aligned}
\text{utility} &= \text{revenue} - \text{cost} \\
\text{revenue} &= \text{hits} \cdot C_{h \to \$} + \text{open-hits} \cdot C_{oh \to \$} \\
\text{cost} &= (\text{hits} - \text{open-hits}) \cdot \text{TCP setup} \cdot C_{TCP \to \$} + \text{idle-time} \cdot C_{it \to \$}
\end{aligned}$$

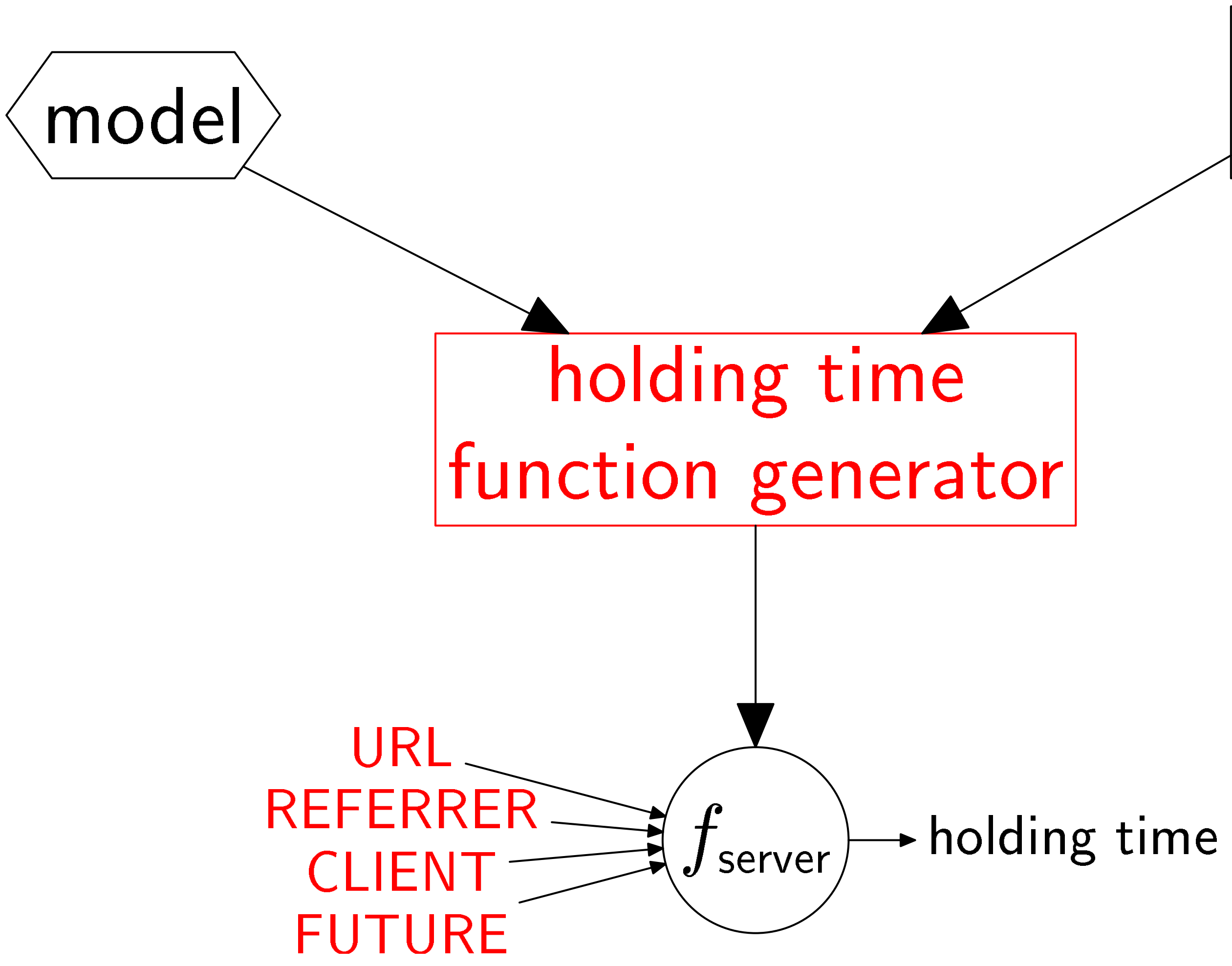
equivalent to

$$\text{utility} \approx C + C_o \cdot \text{open-hits} - C_i \cdot \text{idle-time}$$

Utility Model

Goal: Maximize server's utility.




model
server log
holding time function generator
URL
REFERRER
CLIENT
FUTURE
f_{server}
holding time

Basing Holding Times on HTTP Request Attributes

Current Page has Embedded Images

- probable request within few seconds
- short holding time

Client Predictability:

- request stock market information every 15 minutes
- request Toronto weather information

Types of Attribute-Based Functions

Relevant Parameters

RESOURCE: use current URL
REFERRER: use source for current URL's address
CLIENT: use current client
FUTURE: use knowledge of future requests

OPT: uses FUTURE

- closes connection if idle time more costly than next open-hit
- best possible function
- unimplementable

FIXED: ignores all parameters

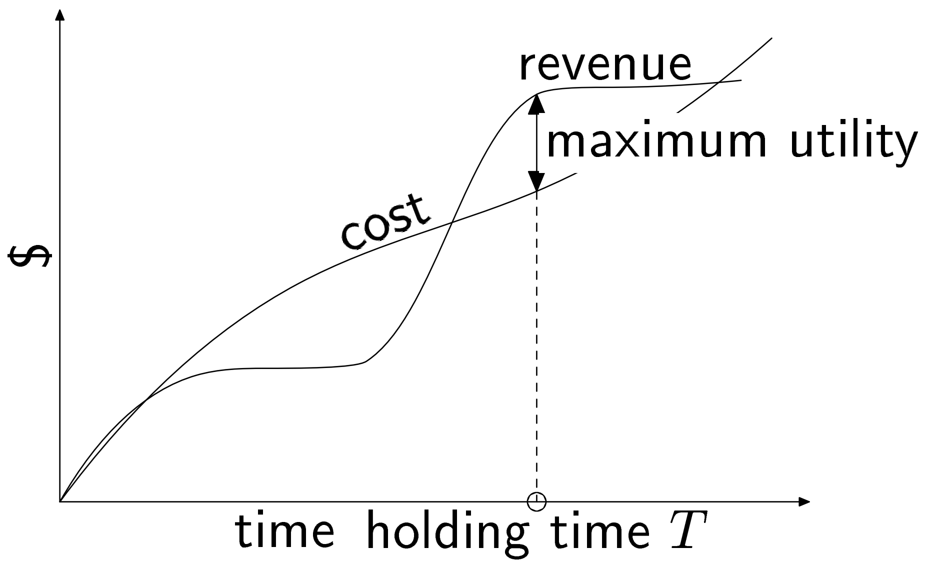
- every holding time equals t seconds
- simple to implement
- HTTP/1.1 suggests this function
- Apache v.1.3 uses 15 seconds

Computing Most Beneficial RESOURCE

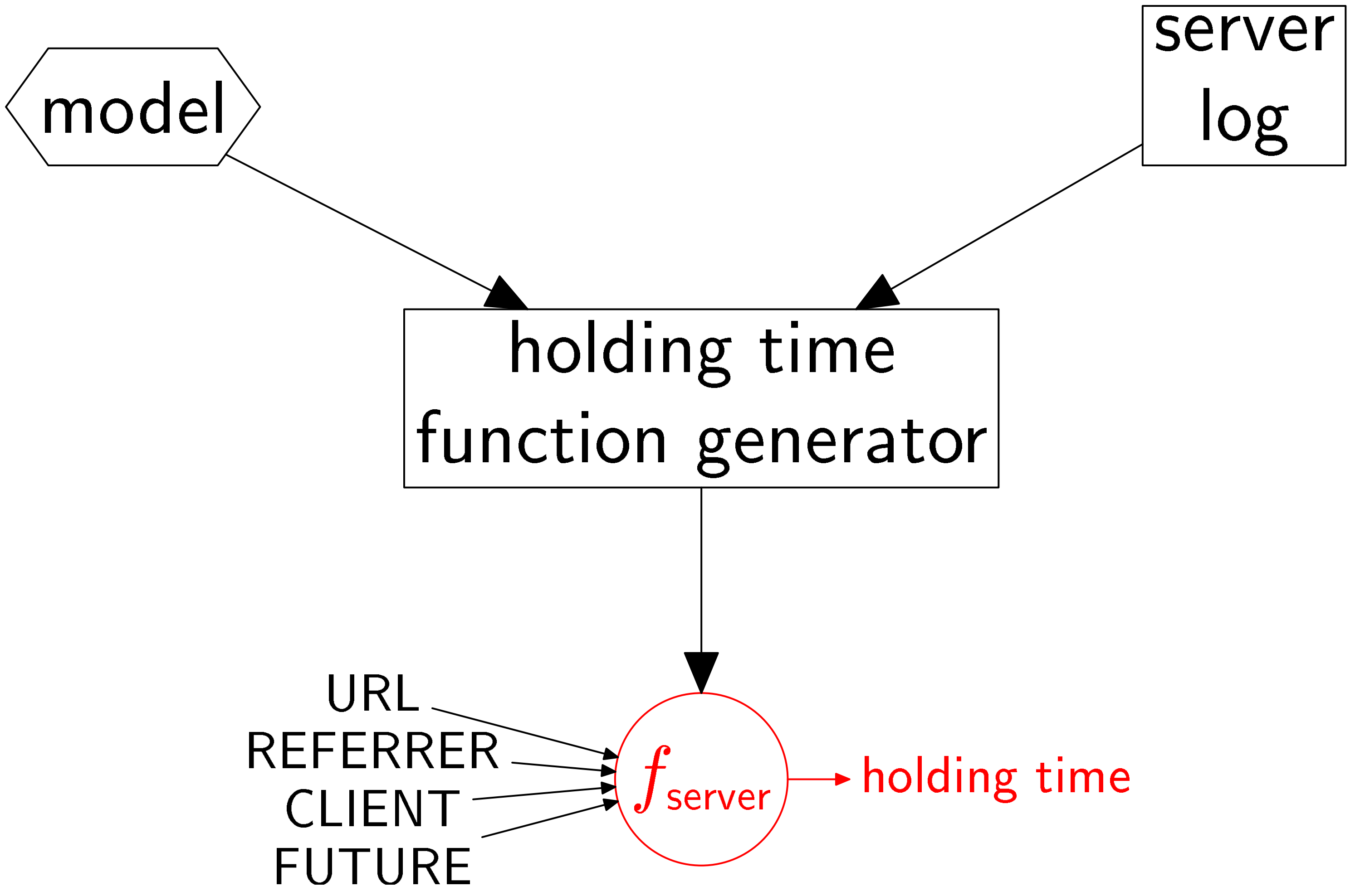
$$\text{utility} \approx C + C_o \cdot \text{ open-hits} - C_i \cdot \text{ idle-time}$$

Server logs: future requests may reflect past requests

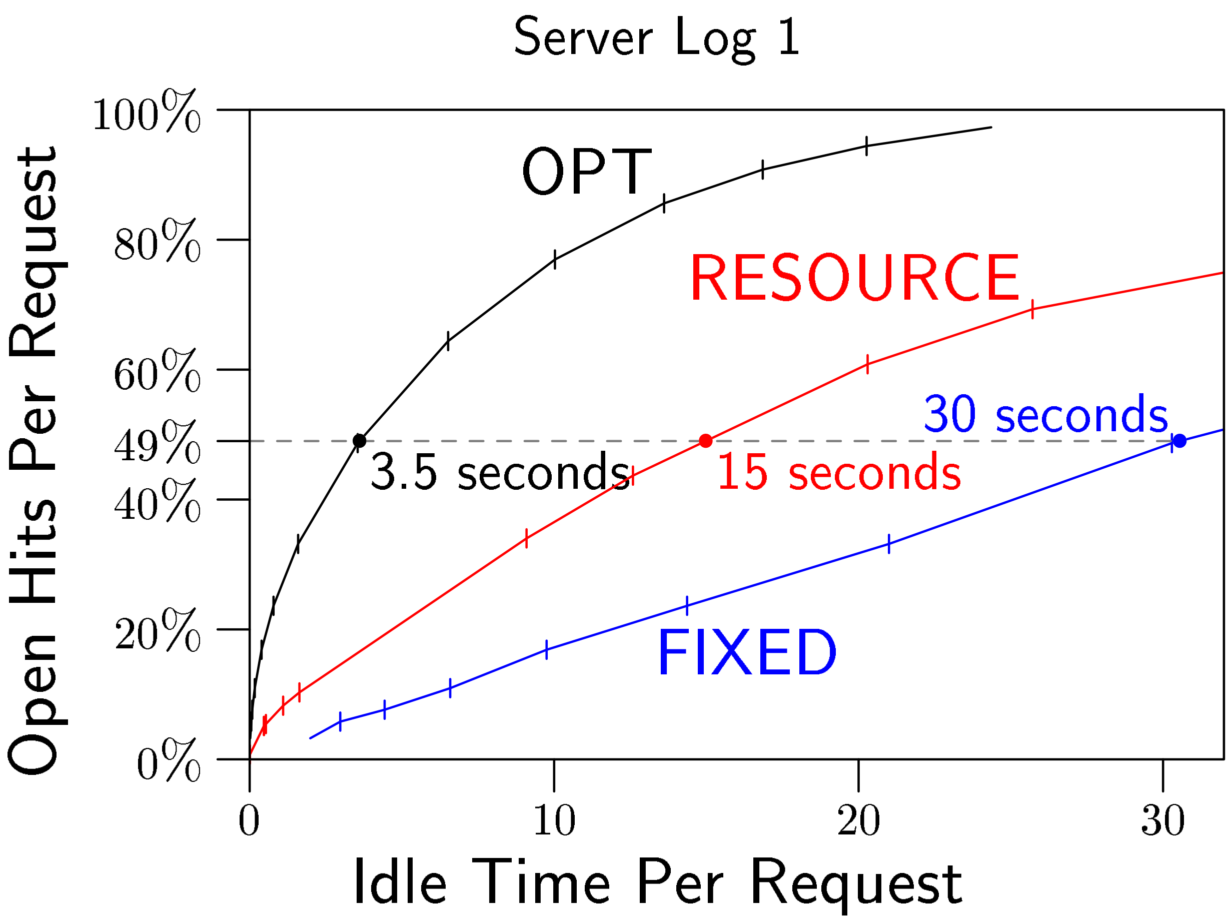
Compute utility: for each resource



Holding time: choose time with greatest utility

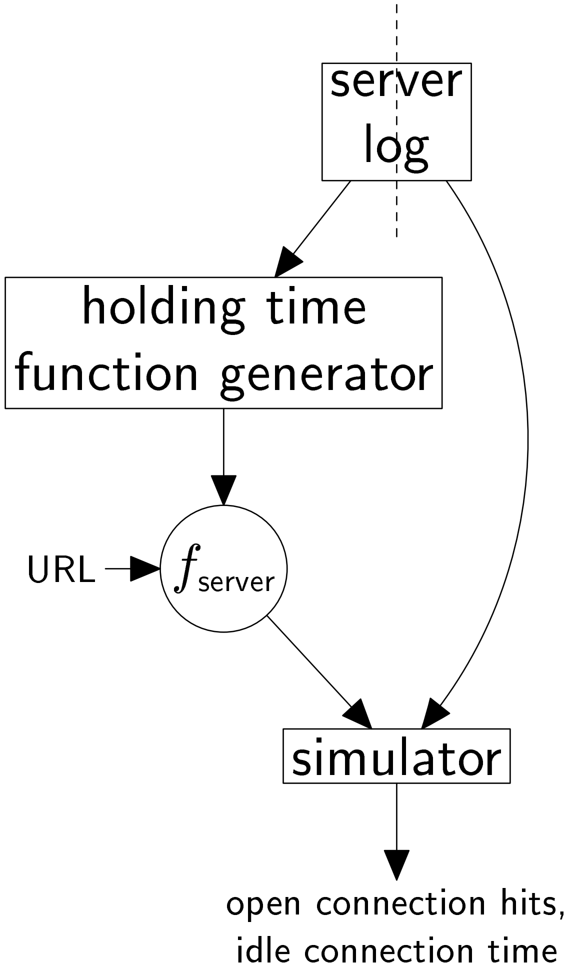

model
server log
holding time function generator
URL
REFERRER
CLIENT
FUTURE
f_{server}
holding time

Predicting Using RESOURCE

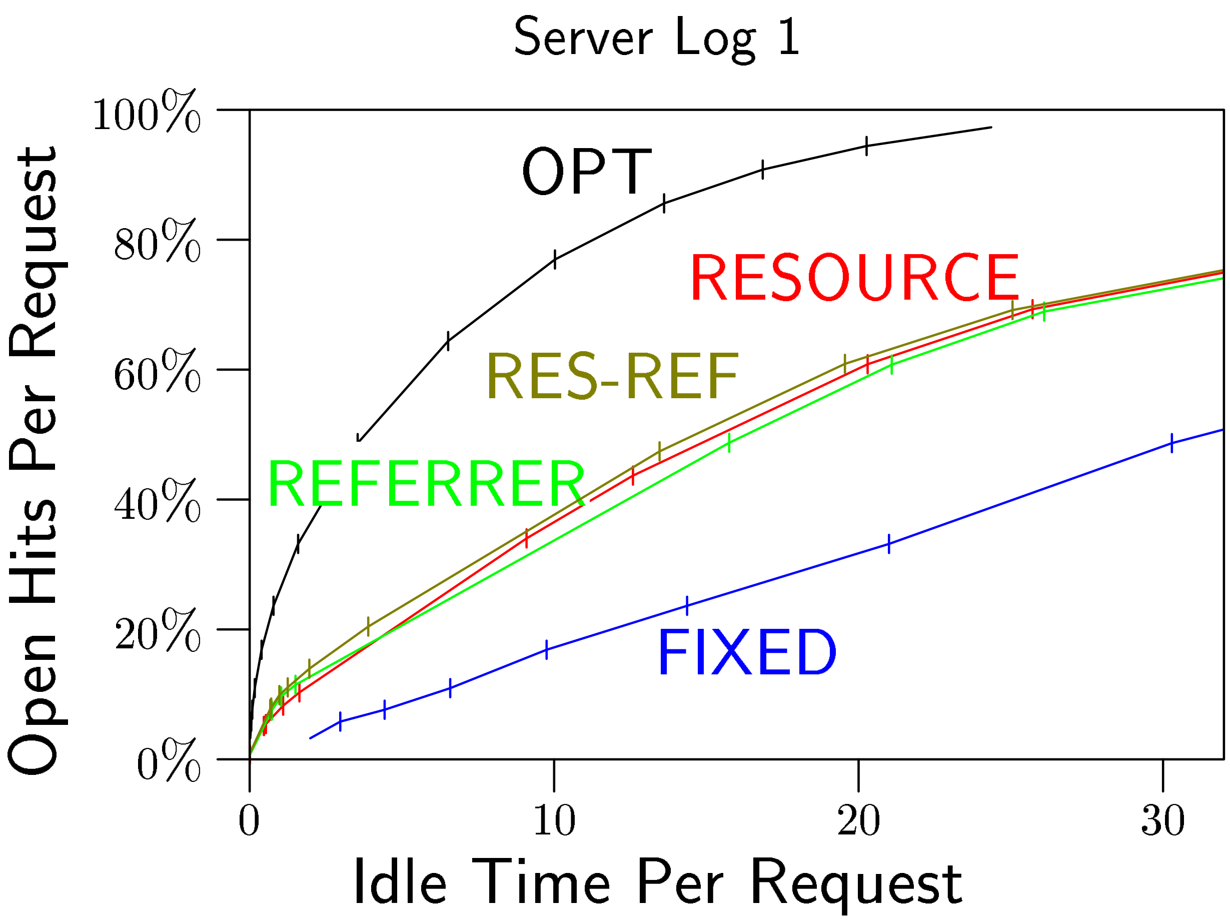


$$\frac{\text{utility}}{\text{connection}} = C_o \frac{\text{open-hits}}{\text{connection}} - C_i \frac{\text{idle-time}}{\text{connection}}$$

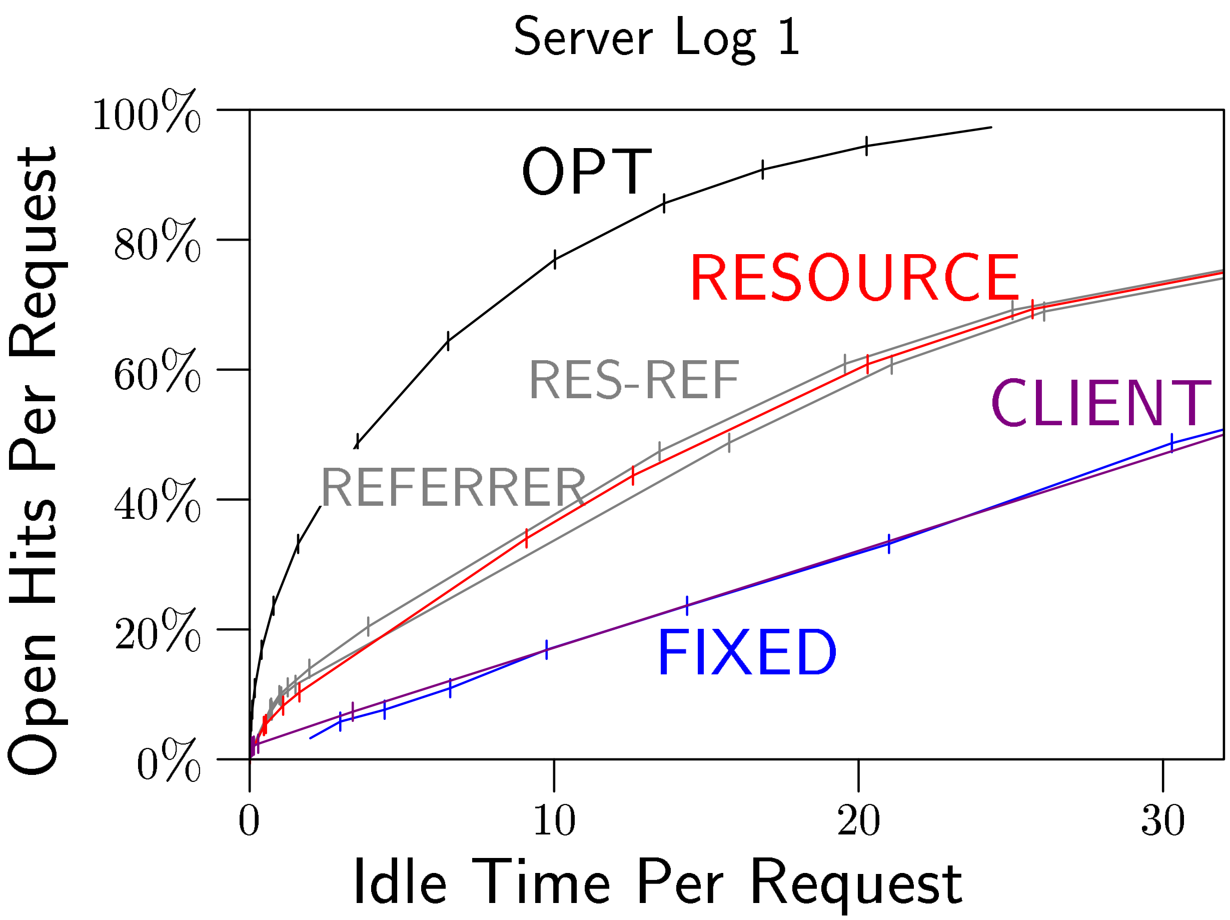
Experimental Method



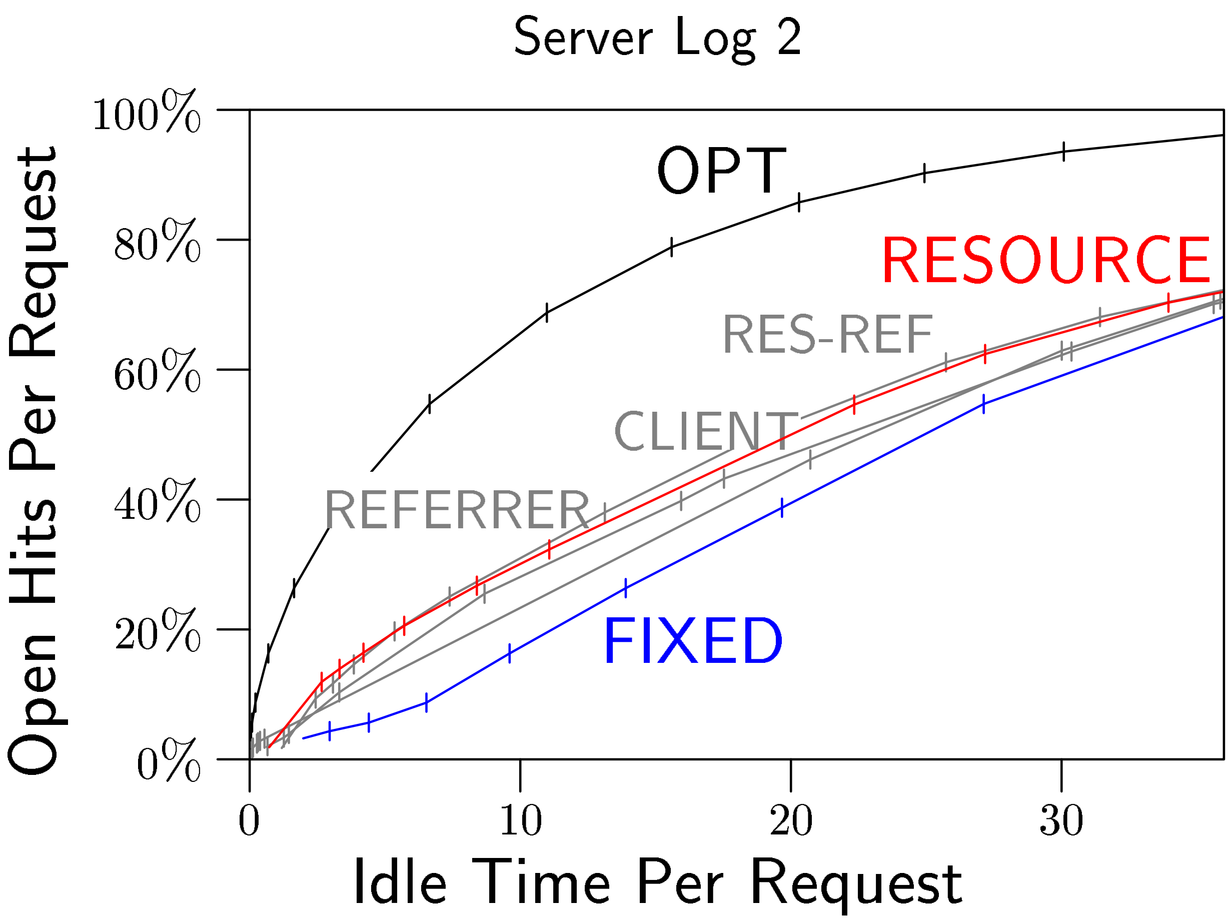
Comparing RESOURCE, REFERRER, RES-REF

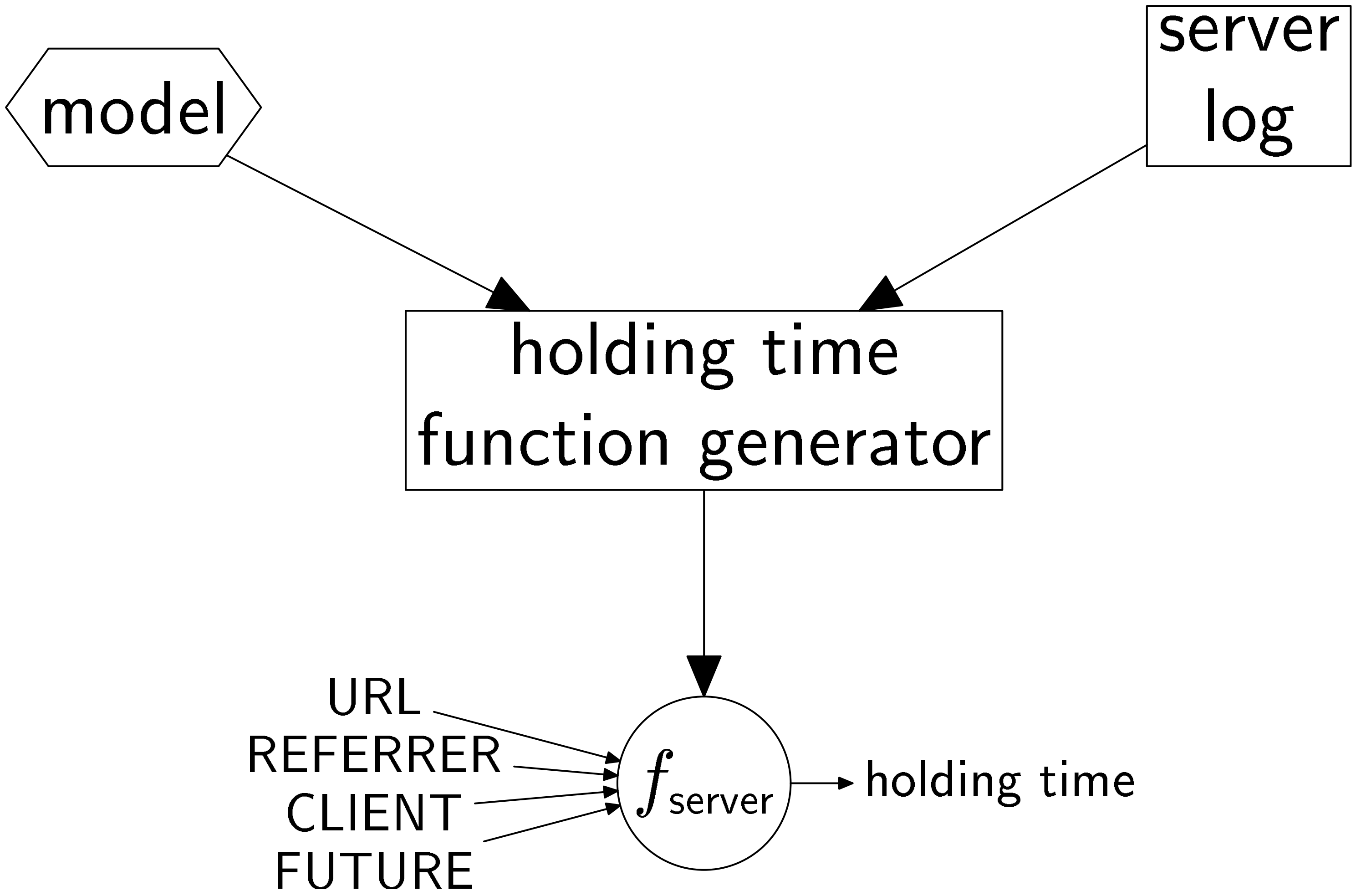


Predicting Using CLIENT



RESOURCE Reliable Attribute




model
server log
holding time function generator
URL
REFERRER
CLIENT
FUTURE
f_{server}
holding time

Conclusions

Use talk outline slide.

- Introduced model for evaluating persistent HTTP connections.
- Presented technique to produce beneficial holding time functions.
- Experimentally showed our holding time functions beneficial.
  - reduce idle connection time up to 50%
- Experimentally evaluated relevance of attributes to holding times.

Future Work

- are conversion rates constant?
- holding times' effects on number of hits?
- effects of clients' closing connections?
- connection management functions for proxies