# Web Workload Characterization at a Micro-architectural Level

Ramesh Radhakrishnan, Lizy John and Freeman L Rawson III*
Laboratory for Computer Architecture
University of Texas, Austin.

*Austin Research Laboratory, IBM
Austin, TX.

# Goals of this study

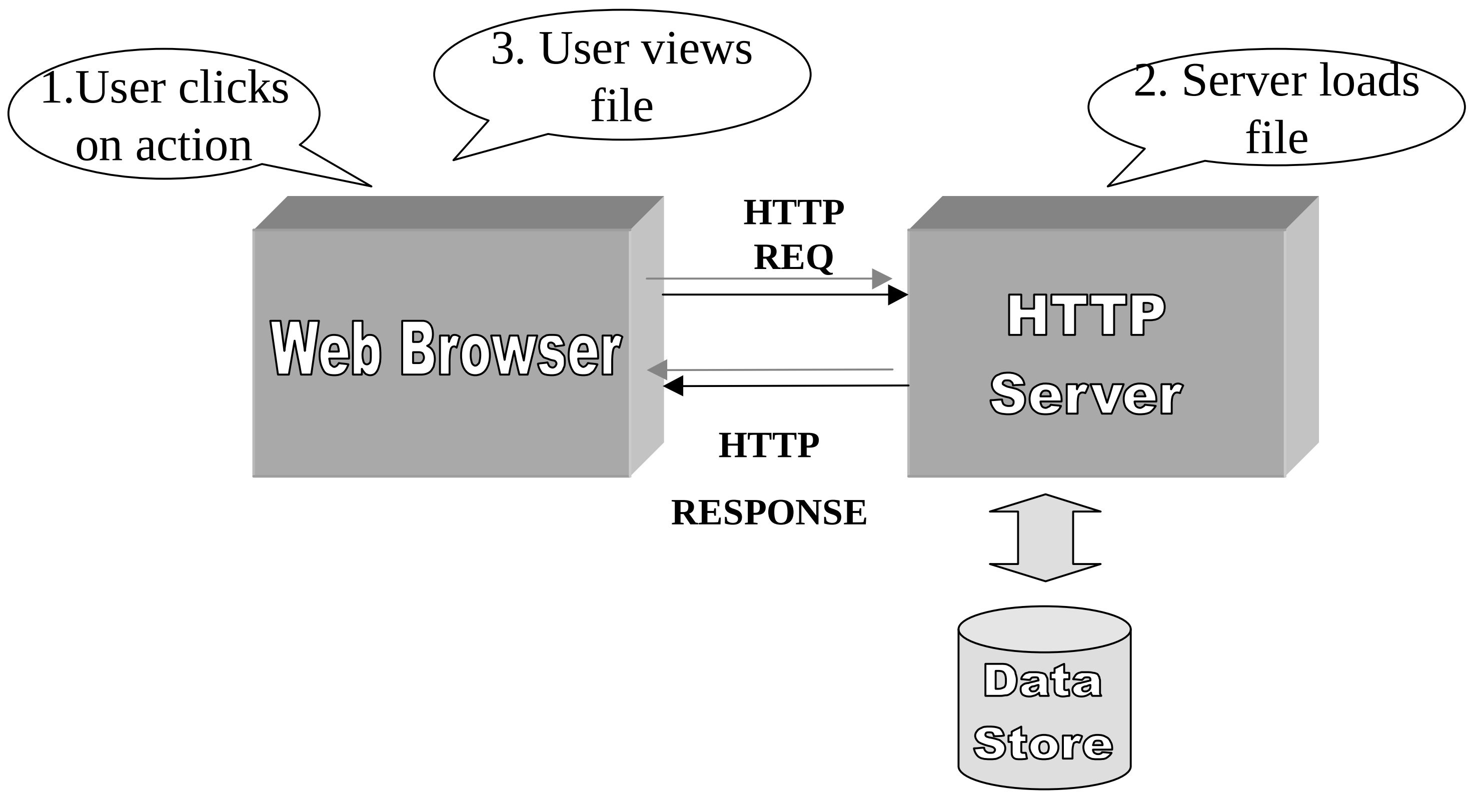
- We study the behavior of modern Web server applications programs to understand how they interact with the underlying Web server, hardware and OS under realistic workloads.
- We monitor and evaluate the performance of the system using hardware performance counters for different workloads, and identify bottlenecks based on collected data

# Workloads

- Our workloads include static requests which form a major percentage of the web traffic, as well as dynamic requests in the form of CGI (Common Gateway Interface) scripts and Java servlets.
- CGI and servlets are gaining popularity and form a significant part of today's Web traffic.

# World Wide Web (WWW)

- The WWW is a distributed hypertext based information system based on the client-server model.



# Web Performance

- Three primary factors that affect Web performance are:
  - The end user whose browser issues a request
  - The network infrastructure that communicates requests and results
  - The Web server (software and underlying hardware)
- Web benchmarks only measure the performance of the Web server.
- Increasing Web server performance means reducing the response time to service a request.

# Web Server Performance

- The ability of a server to respond to requests is influenced by several factors
  - Speed of the network
  - Web Server software
  - Hardware on which the sever is running
- Measuring performance is not trivial for Web servers.

# Web Workloads on current Micro-architectures

- Previous studies have shown that servers spend around 85% of their cycles executing OS code, as opposed to 9% by the SPEC95 suite.
- This raises questions about the performance of servers on current micro-architectures, since many of the recent innovations have been justified on the basis of increasing SPEC95 performance.

# Generating Realistic Web Workloads

- Web benchmarks are used to test the performance of servers.
- Benchmark suites available are:
  - SPECWeb96 from Standard Performance Evaluation Corporation.
  - WebStone from Silicon Graphics.
  - NetPerf from Hewlett-Packard.
- These benchmarks are based on the model that all requests made by a client are for static pages.

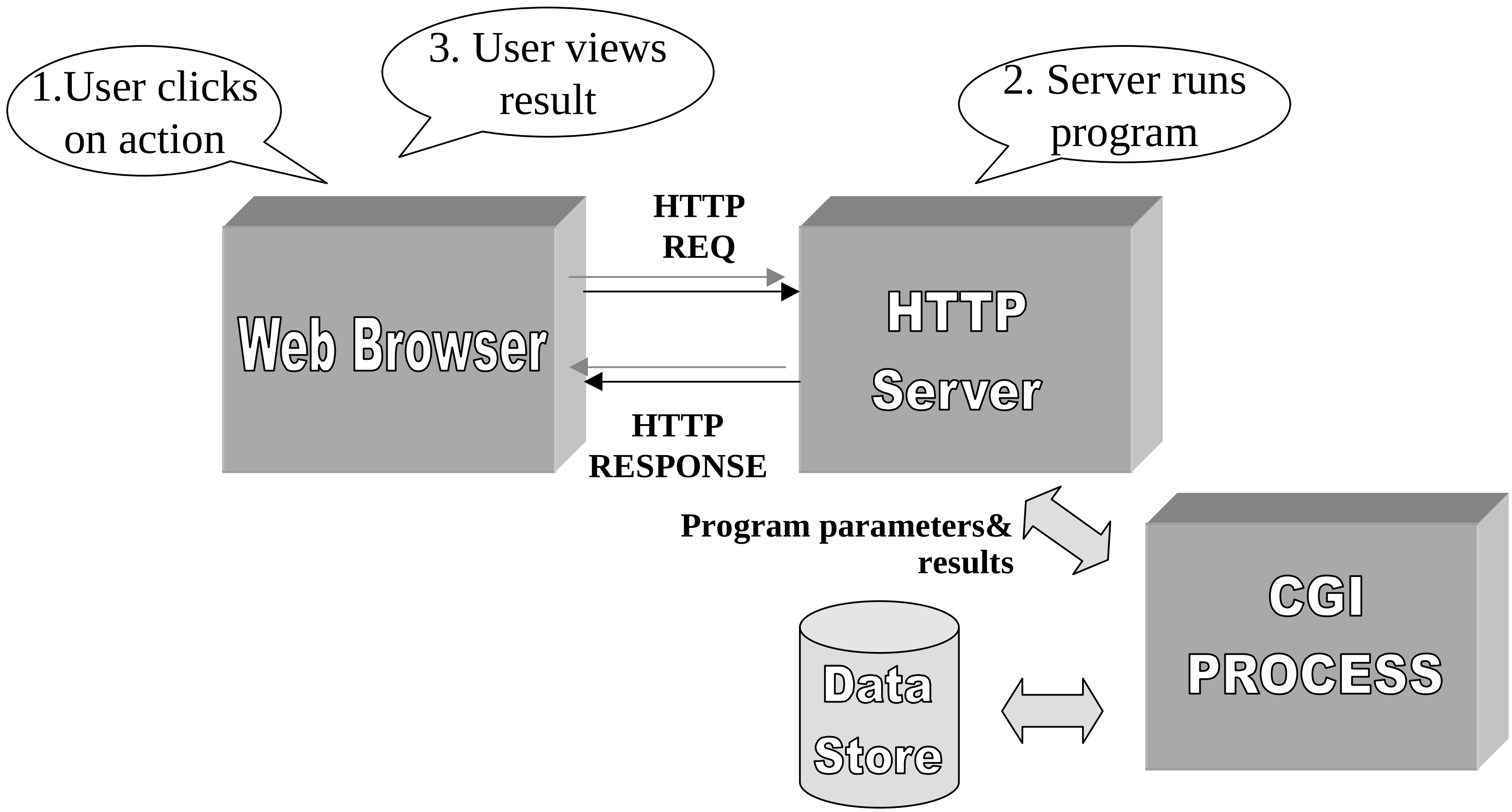
# Web is becoming more dynamic

- The evidence points to the Web becoming more dynamic, especially with *Dynamic HTML*,*CGI Scripts*, *ActiveX*, *Java Servlets* and numerous other Web technologies aimed at increasing the interactivity on the Web.

- Although SPECWeb96 can be perceived to be more unbiased than other vendor-based and maintained benchmarks, it does not match current real-world web workloads.

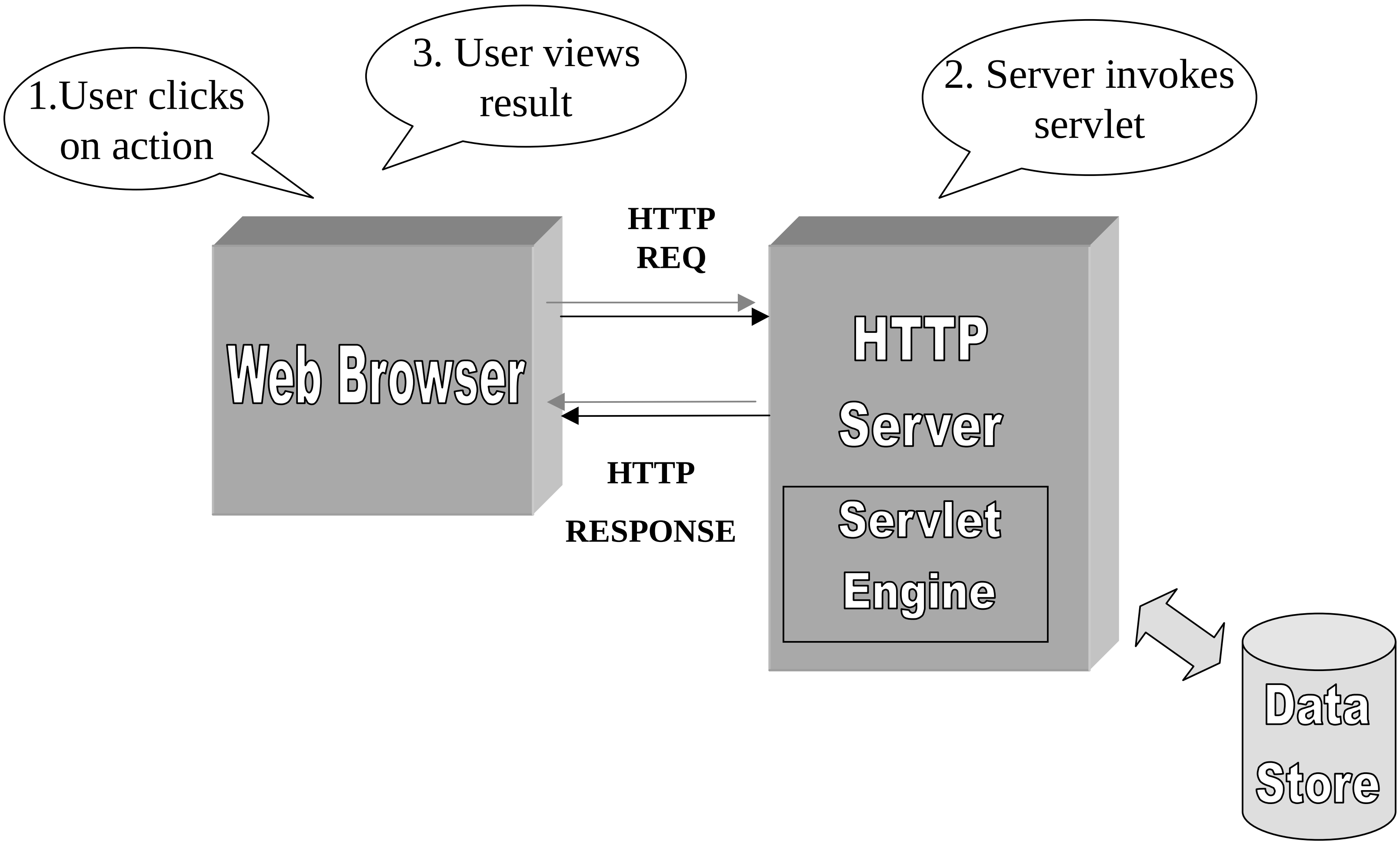
# CGI

- CGI enables the server to create information on the fly, and is executed in real time to output dynamic information.



# Servlets

- Servlets are the Java equivalent of CGI programs.



# Hardware Configuration

- The underlying hardware used for running the server is an UltraSPARC-II processor.

| | |
|---|---|
| Model | Sun Ultra5 (SPARC station 5) |
| Processor | UltraSPARC-II |
| Memory<br>Size | Split Instruction and Data Cache<br>16KB, 2-way set associative I-Cache<br>16Kb direct mapped D-Cache<br>Unified External Cache (L2) |
| Instruction Issue | Up to 4 instructions/cycle |
| Branch Prediction | Dynamic branch prediction scheme,<br>based in two-bit history |
| Instruction Completion | Limited out of order execution |

# Software Configuration

- Apache 1.3.0 is the server engine used to service the requests.
- The server was run on the Solaris 2.6 Operating System.
- *servletrunner* web server engine provided with the JSDK 2.0 was used to service servlet requests.
- Java JDK 1.6 Virtual Machine executes the servlets.

# Experimental Methodology

- Static workload consisted of:
  - 1. Request for an HTML page containing text
  - 2. Request for an image (GIF file)
- Dynamic workload consisted of:
  - 1. Call to an executable (*printenv*) invoked through a CGI interface
  - 2. Request to a Java servlet

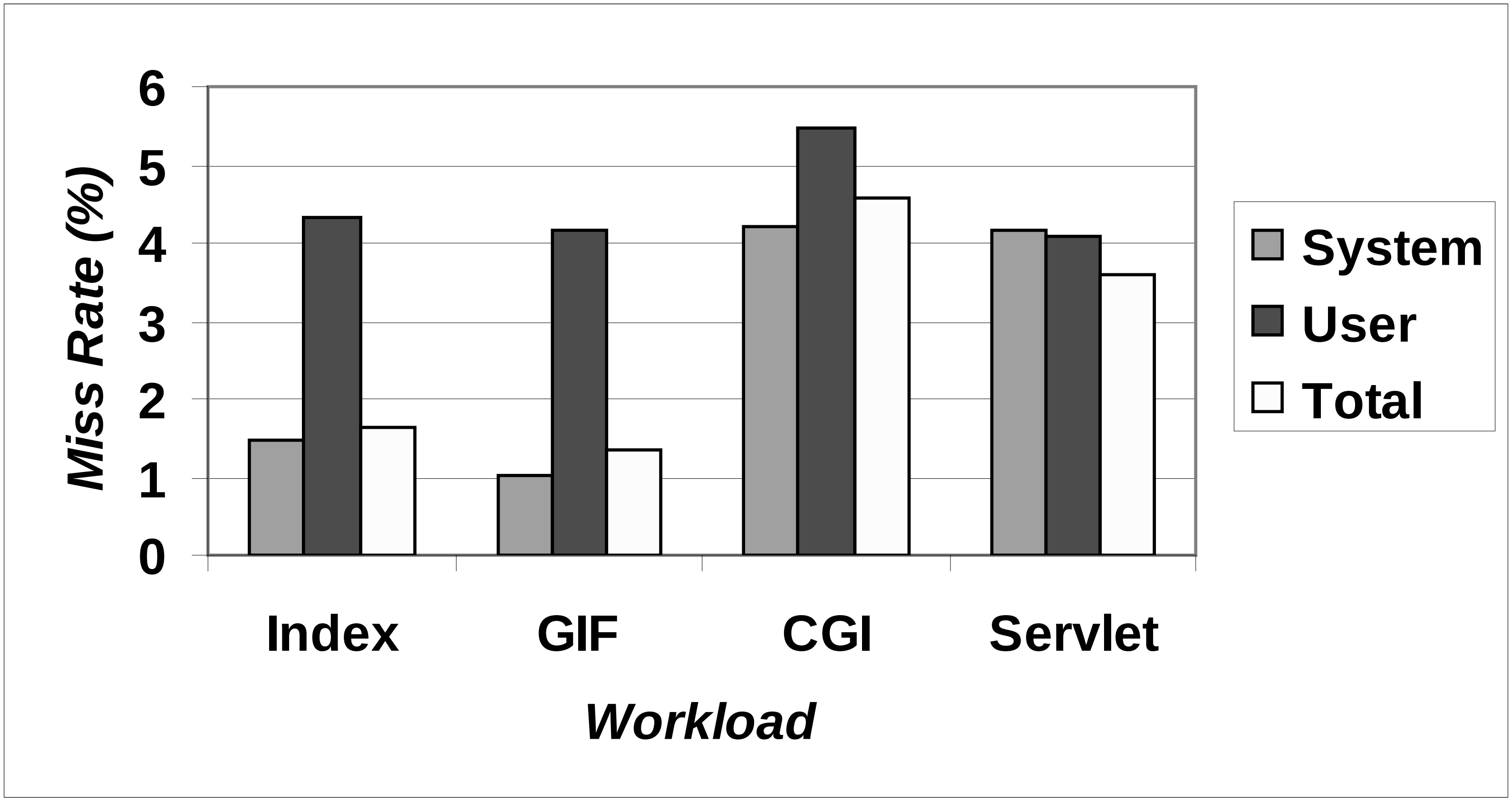
- Each run consisted of 20 requests, made by a dummy client program running on the same machine.

# Performance of Different Workloads

| Static Workload | Instruction | Cycles | CPI |
|---|---|---|---|
| Index (system) | 152159904 | 327325408 | 2.15 |
| Index (user) | 11424319 | 23748612 | 2.07 |
| Index (both) | 159348832 | 350395584 | 2.13 |
| GIF (system) | 208139824 | 433525344 | 2.08 |
| GIF (user) | 10934707 | 21821480 | 1.88 |
| GIF (both) | 207659920 | 442459808 | 2.04 |

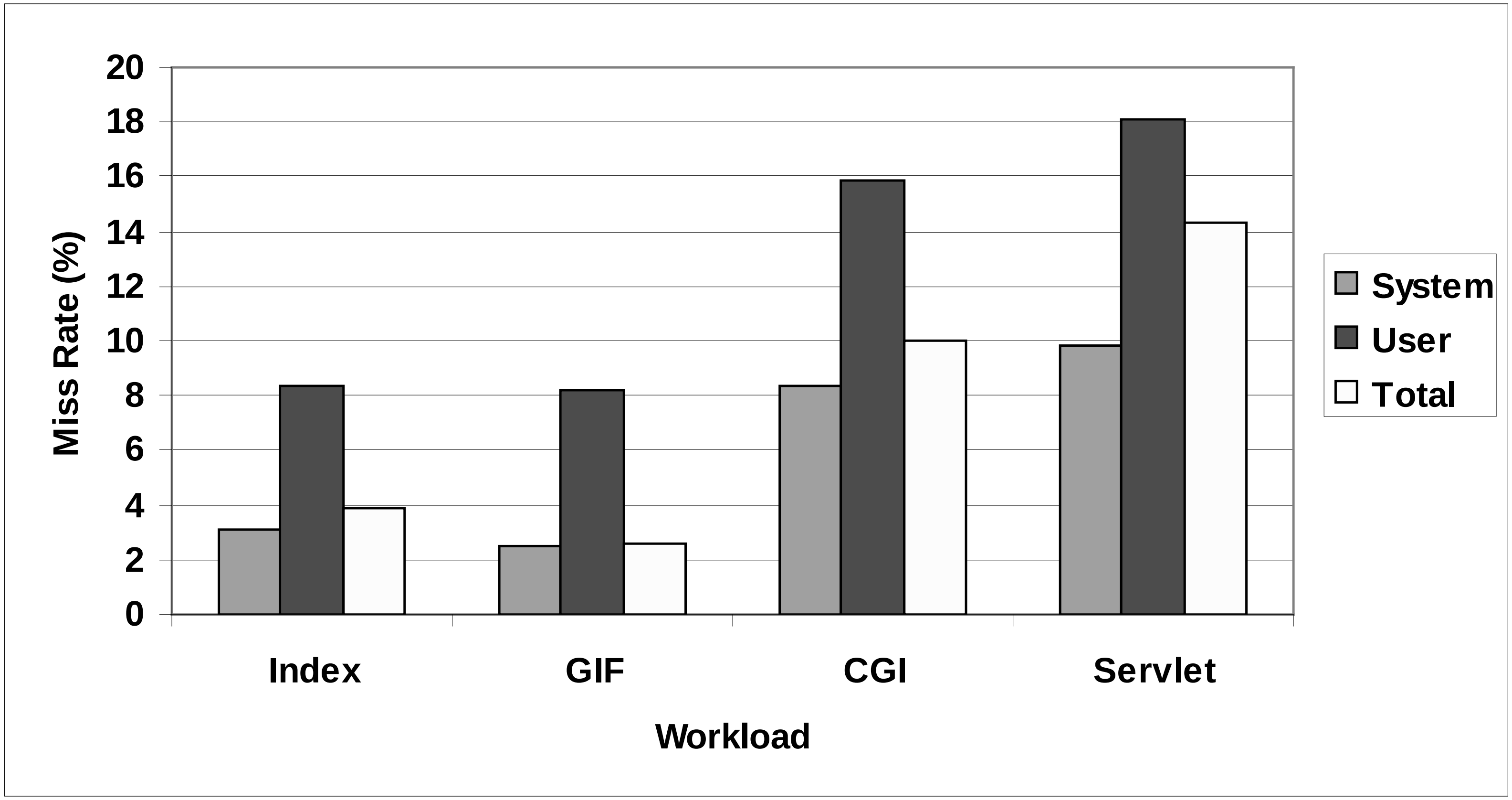
| Dynamic Workload | Instruction | Cycles | CPI |
|---|---|---|---|
| CGI (system) | 141329088 | 399676160 | 2.82 |
| CGI (user) | 30277060 | 80925312 | 2.67 |
| CGI (both) | 178319120 | 482511360 | 2.7 |
| Servlet (system) | 151229216 | 424606304 | 2.8 |
| Servlet (user) | 140923376 | 271369728 | 1.92 |
| Servlet (both) | 287816608 | 686206592 | 2.38 |

# Instruction Cache Performance



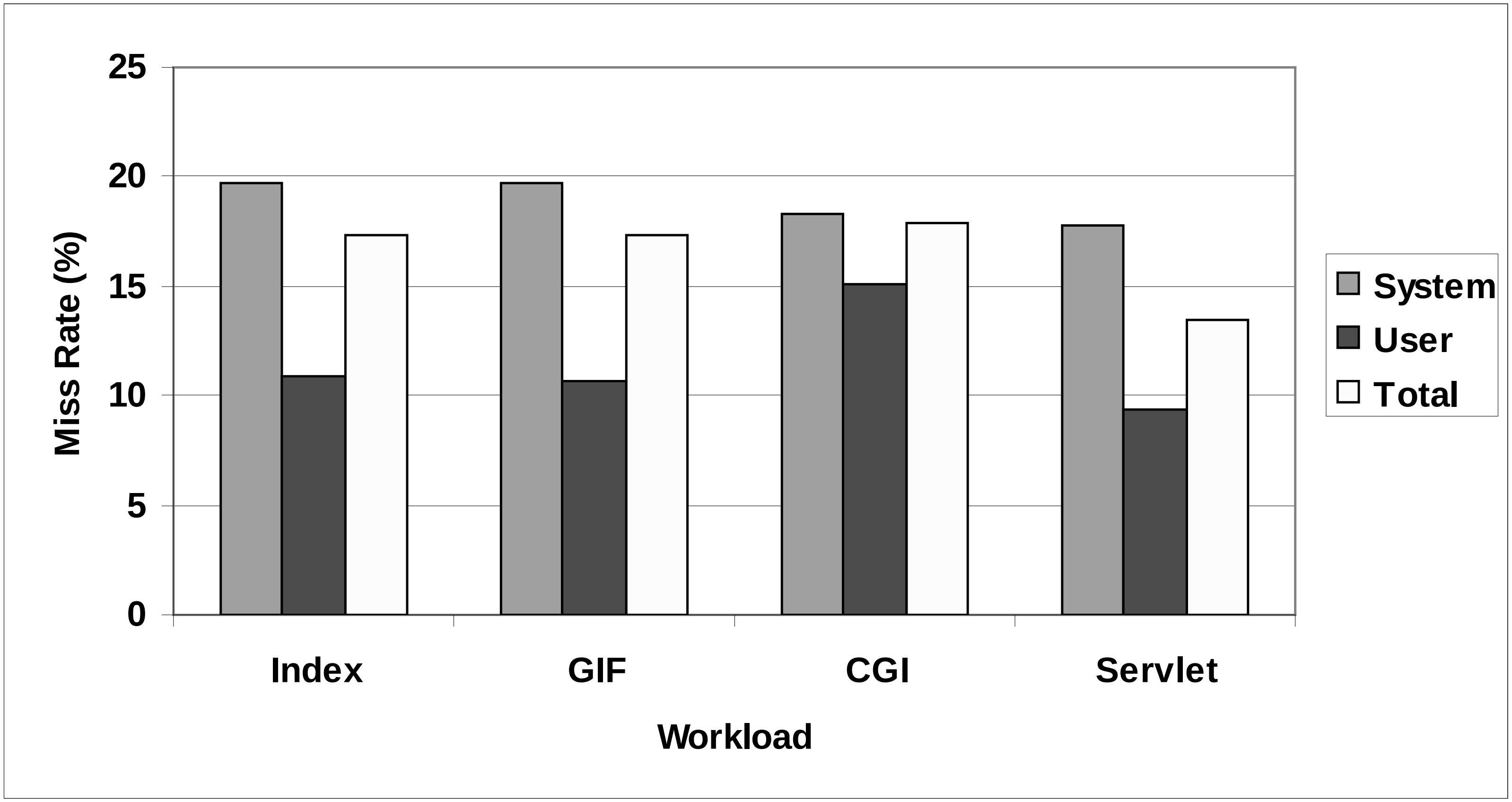

Dynamic workload is seen to have higher miss rates

# Data Cache Performance




Dynamic workload is seen to have higher miss rates.

# External Cache Performance




External cache misses are high for both workloads

# I-Buffer Stalls

- IC miss: % of cycles that the I-Buffer is empty from an IC miss
- branch misprediction: % of cycles that the I-Buffer is empty from branch mispredictions

| | IC-miss | Branch |
|---|---|---|
| index | 8.64 | 10.72 |
| gif | 6.91 | 13.12 |
| cgi | 17.98 | 6.76 |
| servlet | 17.77 | 5.89 |

Stalls are shown as % of Total cycles

# Other Pipeline Stalls

- Store buffer full: Percentage of cycles that the first instruction in the group is a store and the store buffer cannot hold additional stores

- Load Stall : Percentage of cycles that an instruction in the execute depends on an earlier load result that is not yet available.

- Load RAW: Percentage of cycles that there is a load in the execute stage and there is RAW hazard on the oldest outstanding load.

# Other Pipeline Stalls

| | Store Buffer full | Load Load Stall | Load RAW |
|---|---|---|---|
| index | 0.62 | 16.09 | 0.65 |
| gif | 0.49 | 19.02 | 0.49 |
| cgi | 2.47 | 22.70 | 1.10 |
| servlet | 2.97 | 20.77 | 0.94 |

Dynamic workloads exhibit higher pipeline stalls as compared to static workloads.

# Conclusions

- We studied  the behavior of modern web  server applications to understand the interactions with the underlying hardware and operating system.
- Additional functionality provided by the dynamic behavior  of the server is achieved at the cost of a high overhead.
- Dynamic workloads incurred higher cache misses and pipeline stalls, resulting in a higher CPI as compared to static workloads.