Using Latency to Evaluate Interactive System Performance

Yasuhiro Endo, Zheng Wang, J. Bradley Chen,
and Margo Seltzer

{yaz, zhwang, bchen, margo}@eecs.harvard.edu

Harvard University
Division of Engineering and Applied Sciences

Overview

- Study in interactive system performance.
- Latency is the appropriate metric for measuring.
- To this end, we
  - develop techniques for measuring latency and
  - apply the techniques to measure and compare two versions of Windows NT and Windows 95 using real applications.

Slow systems irritate users.

Satisfying the Users

- We improve performance to make users happy.
- Users must be able to notice the improvement.
- We need a performance metric that correlates with user perceived performance.

Outline

- Problems with conventional methodology.
- Techniques for measuring latency.
- Results

Conventional Technique

- Drive the system as hard as possible.
  - SPEC
  - Winstone
- Measures time to complete a series of requests.
  - Measures throughput.

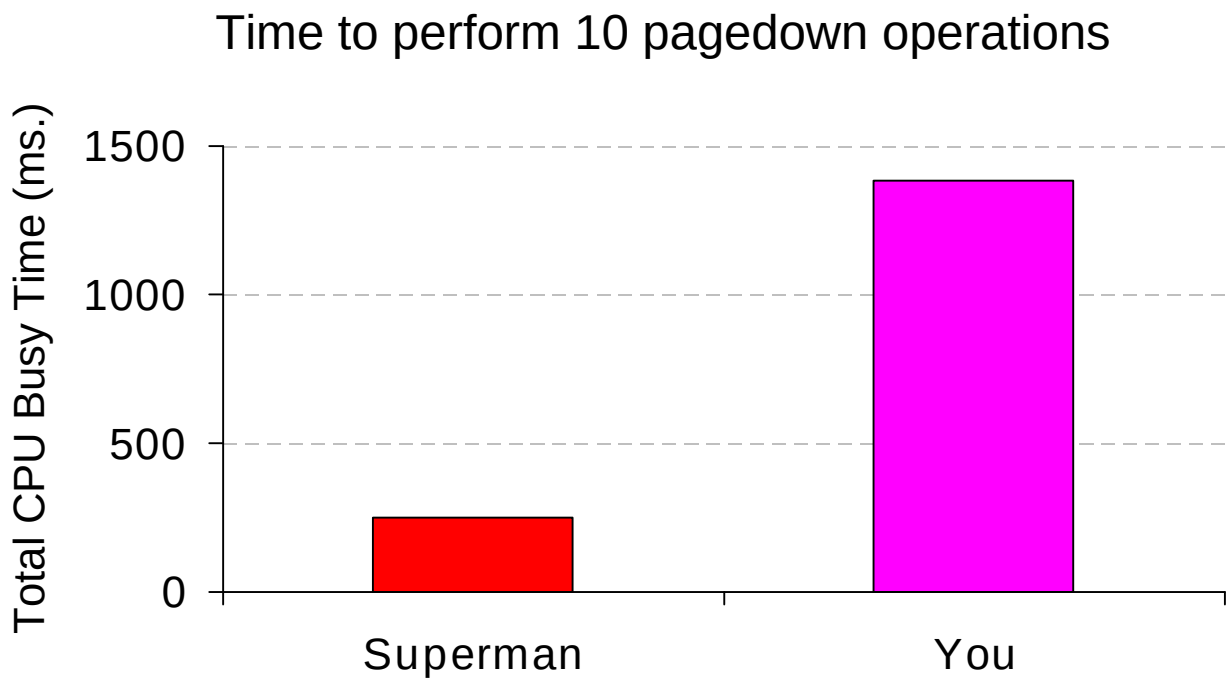
Problems with throughput measurements

- Interactive users
  - are sensitive to variance in latency.
  - cannot feel delays shorter than 100 ms.
  - are slow.
- Do not provide information about the variance in response time.
- Measure activities that do not contribute to user-perceived performance.
- Make inappropriate assumptions.

Batching

When requests are fed quickly, the system often perform less work.

- by batching multiple requests into one
- by not performing tasks overridden by the following task



Methodology

- Use Microsoft Test to simulate user input.
- Drive system as an ordinary user would with pauses between user-initiated events.
- Measure latency of each event rather than the elapsed time.
- Present results graphically.

Hardware

- 100 MHz Pentium CPU
  - 64-bit cycle counter
  - Two 40-bit programmable event counters
- Intel Premiere II motherboard
- PCI bus SCSI (NCR 825 controller)
- One 1GB Fujitsu M1606SAU SCSI hard disk dedicated for each OS.
- Diamond Stealth 64D video adapter
- SMC-8013W ISA Ethernet card

Measuring Latency

- Challenges
  - Interactive events are short in duration.
  - Measurement scheme must have short turn-around time.
  - No access to the source code to either applications or the operating systems.
- Techniques
  - Monitor CPU activities.
  - Monitor message exchange.

Observations

- Interactive systems are idle most of the time.
- A burst of CPU activity is observed when the system handles a request (e.g. keystroke, interrupt, etc.).
- We can infer latency of many events by capturing the duration of such bursts.

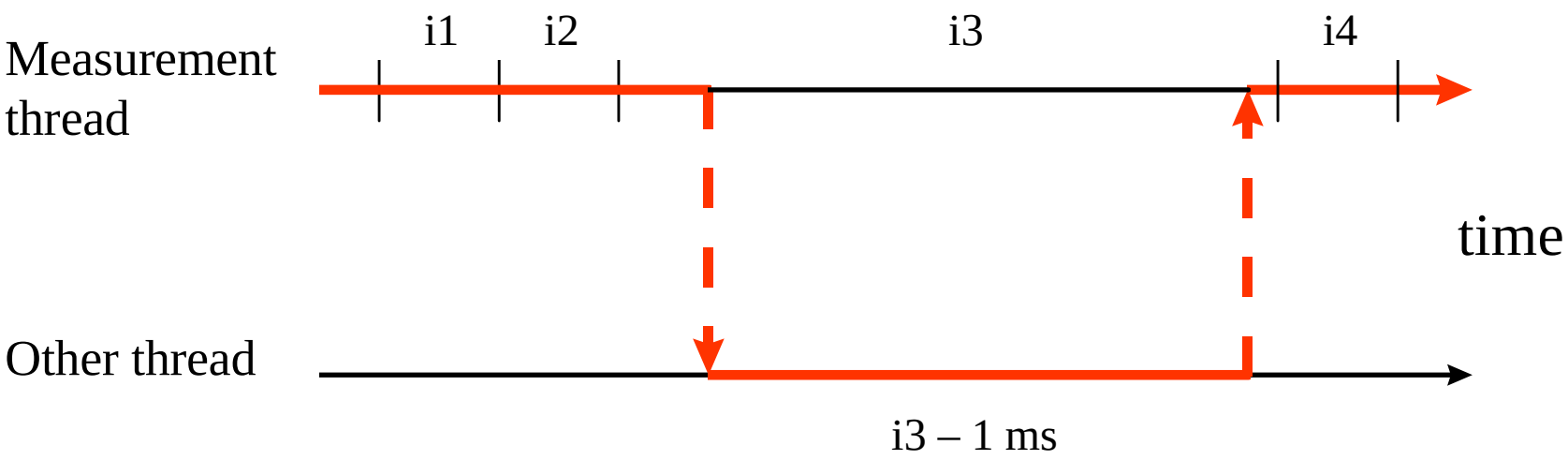
Detecting CPU Activities

- Replace idle loop with a low-priority thread.

```
while (space_left_in_the_buffer) {
        for (i = 0; i < N; i++) /* wait 1 ms */
                ;
        generate_trace_record;
}
```

- N is set such that a record is generated every 1 ms. when system is idle.
- Any non-idle time in the system will show up as elongated time interval between records.

Measuring Lost Time



- Each interval contains 1 ms of idle time.
- i3 contains 1 ms of idle time and i3 – 1 ms. of system busy time.

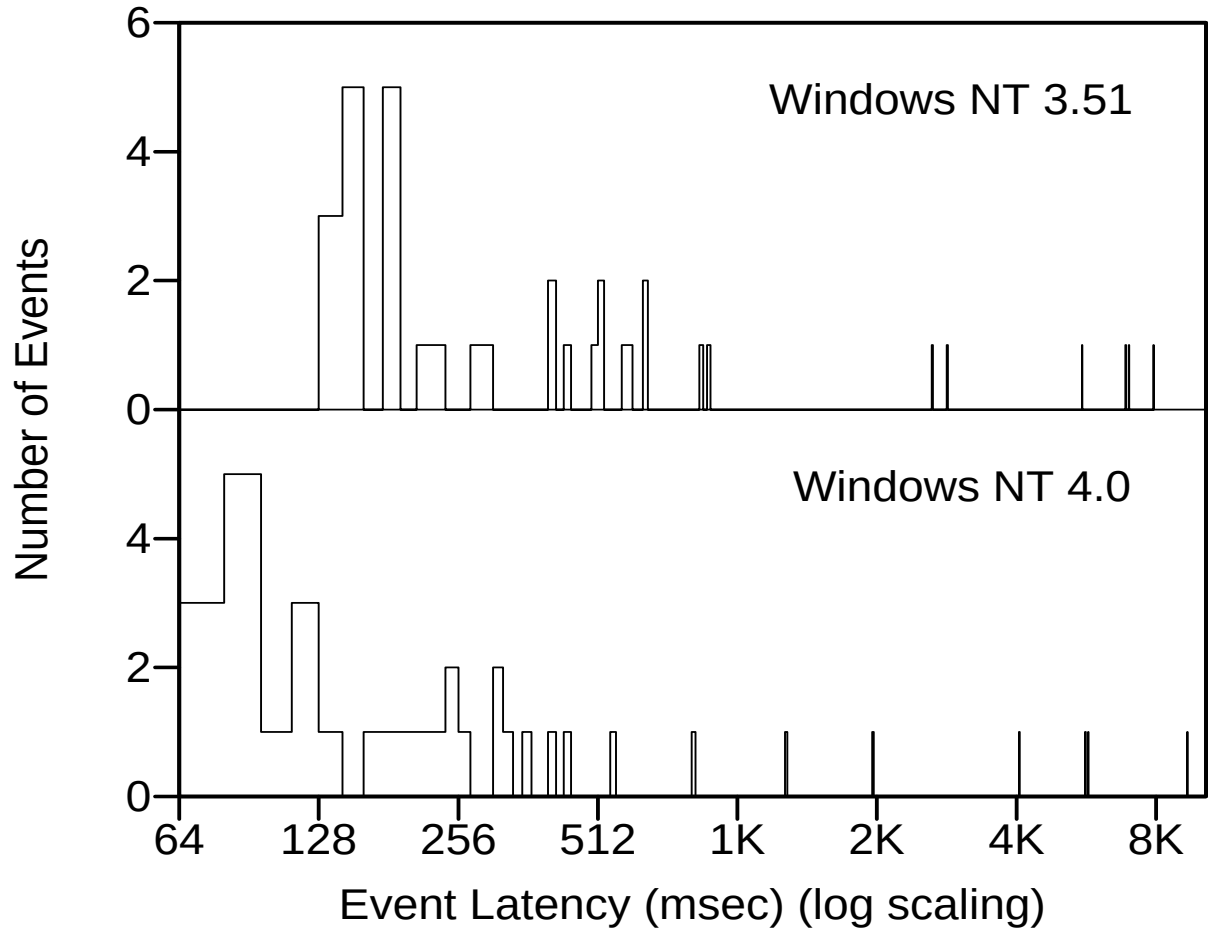
Message Queue Monitoring

- Windows applications receive user input by making several API calls.
  - GetMessage()
  - PeekMessage()
- Monitor messages to pinpoint the beginning and ending of events.
- Intercept and log important API calls.

Task Oriented Benchmark - Powerpoint

- Model an user accomplishing a certain task.
- Edit embedded graphs in Powerpoint.
  - Involves
    - Starting the application.
    - Edit OLE embedded objects.
- Insert delays to model a “normal” user.
- Capture latencies of individual user request.

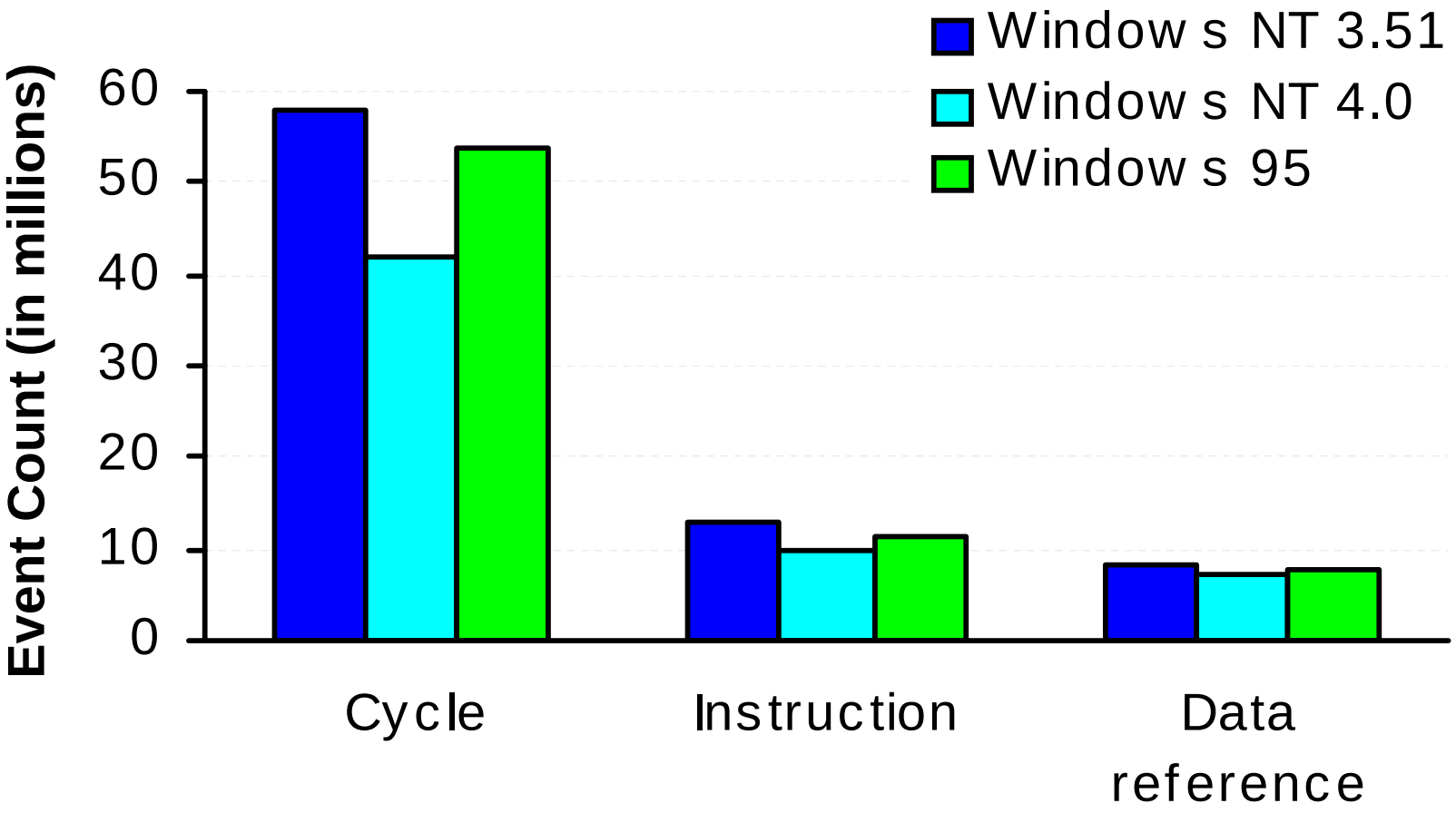
Powerpoint Benchmark Results



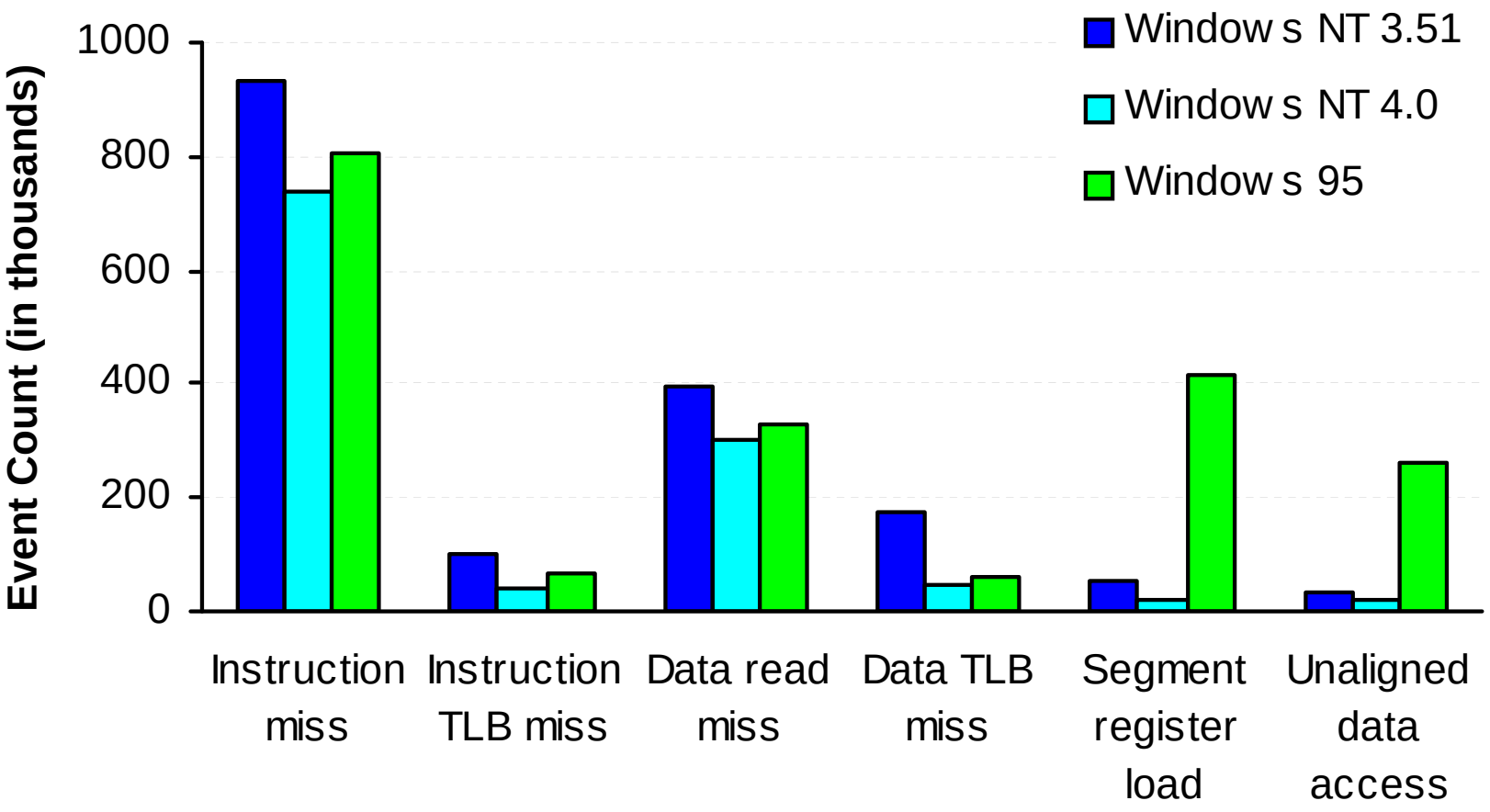
Application Microbenchmarks

- Task oriented benchmarks identify slow operations.
- We should concentrate on improving these.
- Isolate and study them in detail.

OLE Edit


Window s NT 3.51
Window s NT 4.0
Window s 95
Event Count (in millions)
60
50
40
30
20
10
0
Cycle
Instruction
Data reference

OLE Edit


Event Count (in thousands)
1000
800
600
400
200
0
Windows NT 3.51
Windows NT 4.0
Windows 95
Instruction miss
Instruction TLB miss
Data read miss
Data TLB miss
Segment register load
Unaligned data access

Conclusion

- We study performance to make users happy.
- To direct our efforts in the right direction, we need a metric that accurately reflects user-perceived performance.
- Latency of each event **can** be measured.

What Now?

- Develop a better/simpler metric
  - Integrate with human factors research.
    - Simplify results.
    - How do expectations affect perception?
- Automate data collection process.
- Develop a better optimization methodology
  - Improve user-perceived performance.
- Capture unusual situations.

Using Latency to Evaluate Interactive System Performance

Yasuhiro Endo, Zheng Wang, J. Bradley Chen,
and Margo Seltzer

{yaz, zhwang, bchen, margo}@eecs.harvard.edu

Harvard University
Division of Engineering and Applied Sciences