# Computer Execution Statistics As a Source of Big Data

Kai Shen



## Computer Execution Statistics

- Computers don't just process/analyze data, their executions also leave a trail of data that can be useful
- Software system logs
  - Application logs, OS kernel logs, network message traces, error traces, …
  - High data volume over time across many machines
- Hardware events
  - Instruction execution rates, floating point operations, cache accesses/misses, memory accesses, …
  - Very high data volume (at high sampling rate) even on one machine
- Others like the power traces



## Computer Execution Statistics

- Computer execution statistics
  - Software system logs
  - Hardware events
  - Others like the power traces
- What are they useful for?
  - Performance analysis
  - Reliability assessment
  - Provenance tracking
  - Understand workload/application patterns
  - Privacy and security



## Performance Analysis

- Network message traces

```
#128 <1.732364sec> NET SEND: PID:13661 HOST_ADDR -> 128.151.67.29:41800 REMOTE_ADDR -> 128.151.67.228:8080 SIZE:177
#129 <1.734737sec> NET RECV: PID:13661 HOST_ADDR -> 128.151.67.29:41800 REMOTE_ADDR -> 128.151.67.228:8080 SIZE:1615
#130 <1.736060sec> NET RECV: PID:13661 HOST_ADDR -> 128.151.67.29:41800 REMOTE_ADDR -> 128.151.67.228:8080 SIZE:684
#131 <1.738076sec> NET RECV: PID:13661 HOST_ADDR -> 128.151.67.29:41800 REMOTE_ADDR -> 128.151.67.228:8080 SIZE:1448
#132 <1.738398sec> NET SEND: PID:13661 HOST_ADDR -> 128.151.67.29:41800 REMOTE_ADDR -> 128.151.67.228:8080 SIZE:600
#133 <1.738403sec> NET RECV: PID:13661 HOST_ADDR -> 128.151.67.29:41800 REMOTE_ADDR -> 128.151.67.228:8080 SIZE:568
#134 <1.738421sec> NET SEND: PID:13661 HOST_ADDR -> 128.151.67.29:41800 REMOTE_ADDR -> 128.151.67.228:8080 SIZE:363
#135 <1.752501sec> NET RECV: PID:13661 HOST_ADDR -> 128.151.67.29:41800 REMOTE_ADDR -> 128.151.67.228:8080 SIZE:12
```

- To understand request/response performance
  - Identify matching send/receive events and compute delay
  - #128 → #129 delay 2.4 msecs
  - #132 → #133 delay 5 usecs **TOO SHORT!**
  - #132 → #135 delay 14.1 msecs
- Inference is useful, by imprecise

## Performance Analysis

- Analyze network message traces in a distributed system
  - Match send event at S with receive event at R
  - There are many send events and many receive events, how to find matching pairs?
- Statistical correlation
  - Assume that the network delays within a short period of time are largely stable, find a time shift such that the send events will best align with the receive events [Aguilera et al. 2003]



## Performance Analysis

- Performance problems occur in a complex IT system with many components ⇨ very hard to find the causes and fix
- Black-box machine learning [Cohen et al. 2004]
  - Collect a ton of system logs and traces (as much as you can)
    - CPU time, disk I/O, amount of swap space on machine, network activities, Apache web server statistics, database events, Javascript warnings, file systems alerts, … …
  - Build a large history of such traces and problem manifestations
  - Train a classifier (Bayesian, decision tree, etc.) that link system metrics to problems ⇨ help problem diagnosis
  - Link advanced system metrics to problems ⇨ predict problem in future



## Reliability Assessment

- Component (disk, memory, processor, etc.) reliability assessments are hard
  - They are largely reliable so for sufficient error statistics, such assessment requires very large-scale data collection
- Google accumulated significant data on disk failures and made some statistical analysis [Pinheiro et al. 2007]
  - Correlation with age and utilization (Figure 3)
    - Bathtub reliability curve
  - Correlation with temperature (Figure 5)
- Correlation does always mean causality!

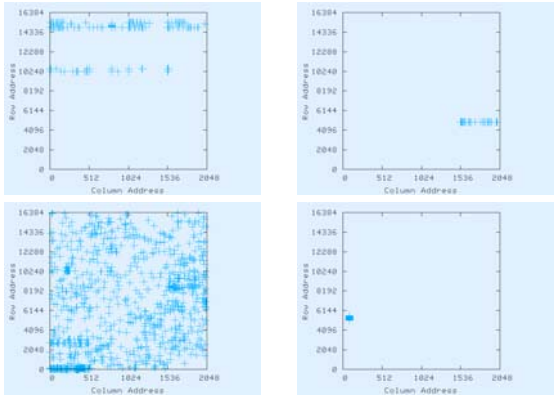


## Reliability Assessment

- Memory errors
  - Transient errors (particle strikes, cosmic rays, etc.)
  - Permanent chip defects
- Understanding memory error rates helps the computer system reliability management and devise countermeasures
- We monitored 212 machines (4GB each) at Ask.com by enforcing hourly memory scrubbing on each chip [Li et al. 2007]
  - Over 9 month, we found two transient errors
  - With 99% probability, the error rate is less than 0.56 FIT/Mb, orders of magnitude lower than 200-5000 FITs/Mb reported previously

## Reliability Assessment

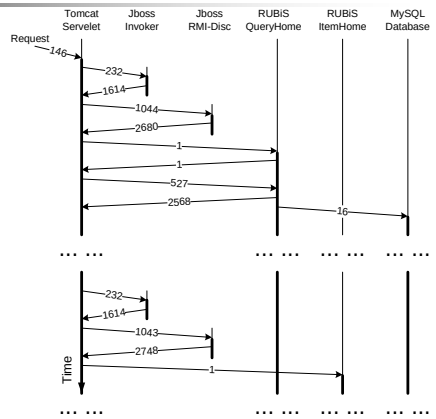
We found 9 chips with permanent errors





## Provenance Tracking

- Information processing flows through multiple components in a complex system (Web server, application server, databases, …)
  - The flows can be tracked by monitoring computer systems events (network sockets, process forking, etc.)



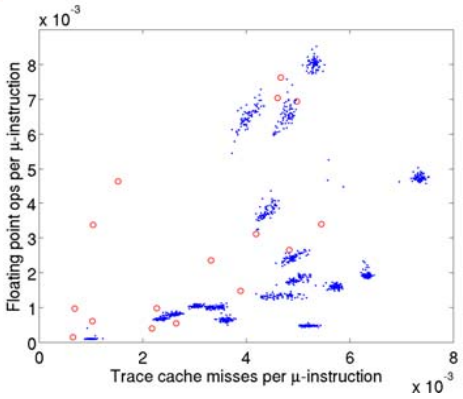



## Processor Hardware Statistics

- Hardware counters on modern processors
  - Instruction mix, rate of execution, branch prediction accuracy, cache/ memory activities, and many with unclear semantics to software people
  - Vast amount of ("big") data accumulated (a thousand samples per second per metric)
  - Free-of-cost
  - Provide rich information on system performance, power, and normal/ abnormal patterns
- Utilization
  - harnessing these hardware counters for workload modeling, performance assurance, and security



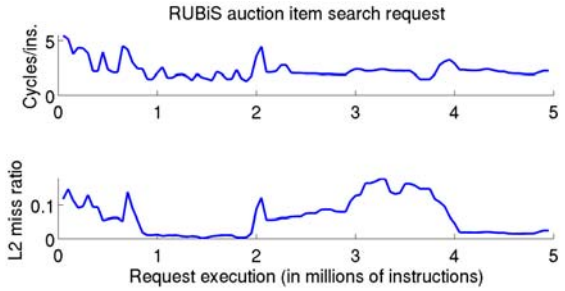
## Request Classification and Anomaly Detection [ASPLOS'08]




- Blue dots – normal TPC-H requests
- Red circles – anomalies (simulating SQL injection attacks by removing conditional statement in WHERE clause)
- Metrics of the first 10msec execution

## Fine-Grained Behavior Variations [ASPLOS'10]

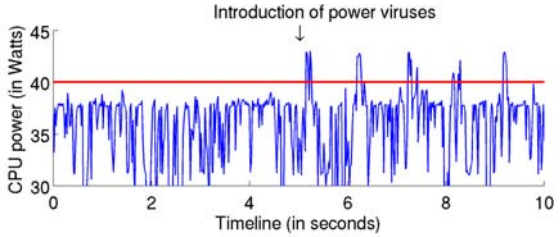
- Fluctuating behaviors over the course of one request execution
  - Opportunties for signature identification and performance analysis






## Power Virus Containment [ASPLOS'13]

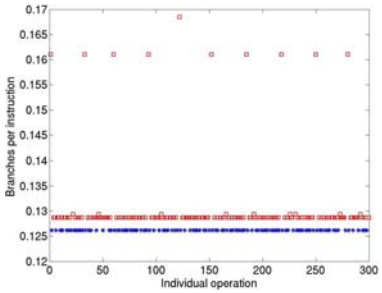
- Power viruses – unusually high power tasks, by stressing multiple power-consuming resources simultaneously
- Simply-constructed (200 lines of Java code) power viruses injected to a Google App Engine workload






## Stealing the RSA key in OpenSSL [HotOS'07]

- $X^d$ is decomposed into a series of square and multiply operations, e.g., $X^{11}$ is decomposed into $((X^2)^2*X)^2*X$
- The execution order of the square and multiply operations can be learned by reading the processor hardware counters






## Power Traces

- Power measurement is fairly easy (compared to collecting processor hardware statistics)
- Medical devices run critical tasks yet they are vulnerable to computer viruses like normal computers do
  - Researchers collect data on power traces and virus infections [WattsUpDoc 2013]
  - Pattern correlation and machine learning tell the power behaviors that indicate virus infections