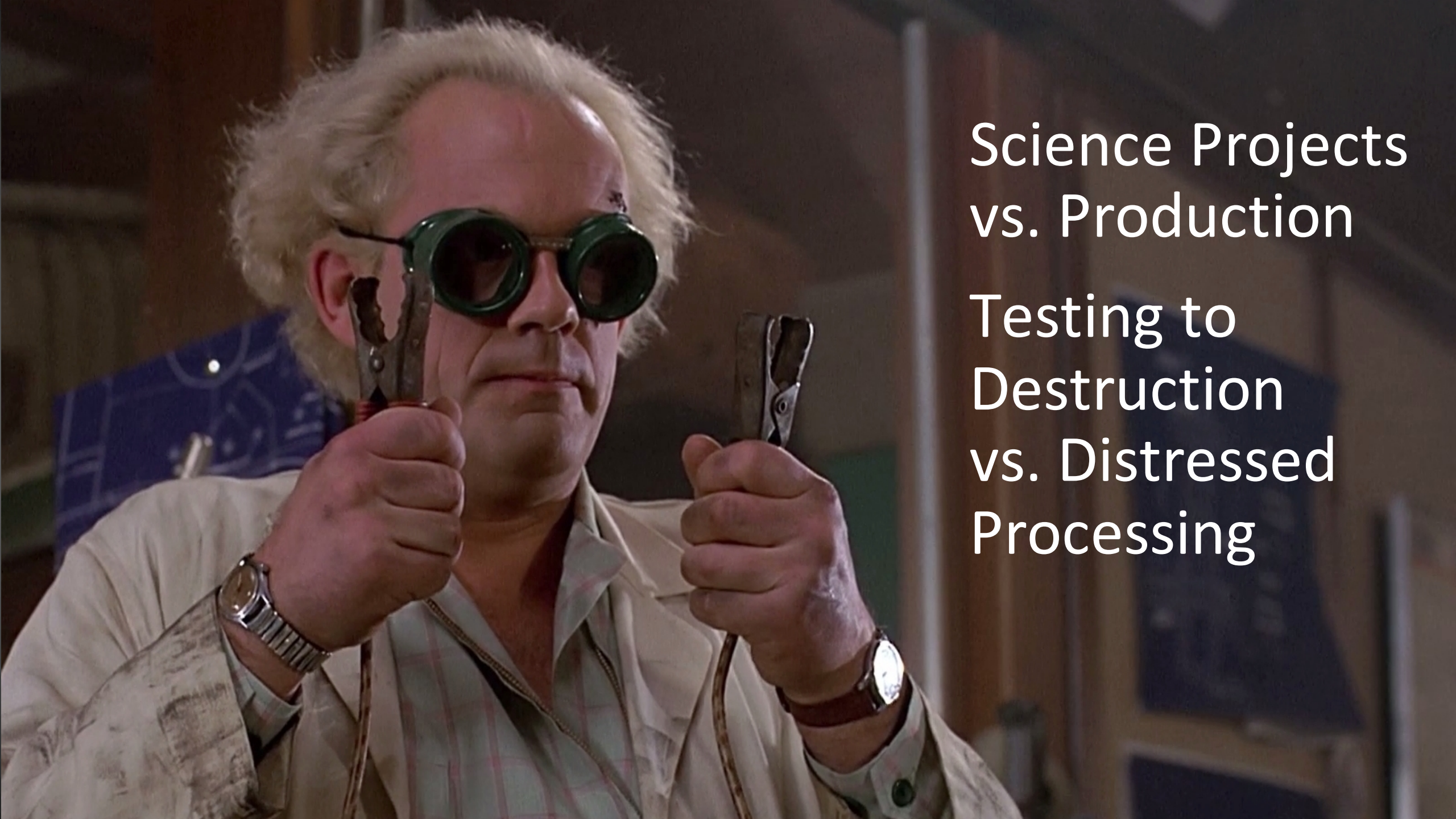
Lies, Damn Lies and Benchmarks:

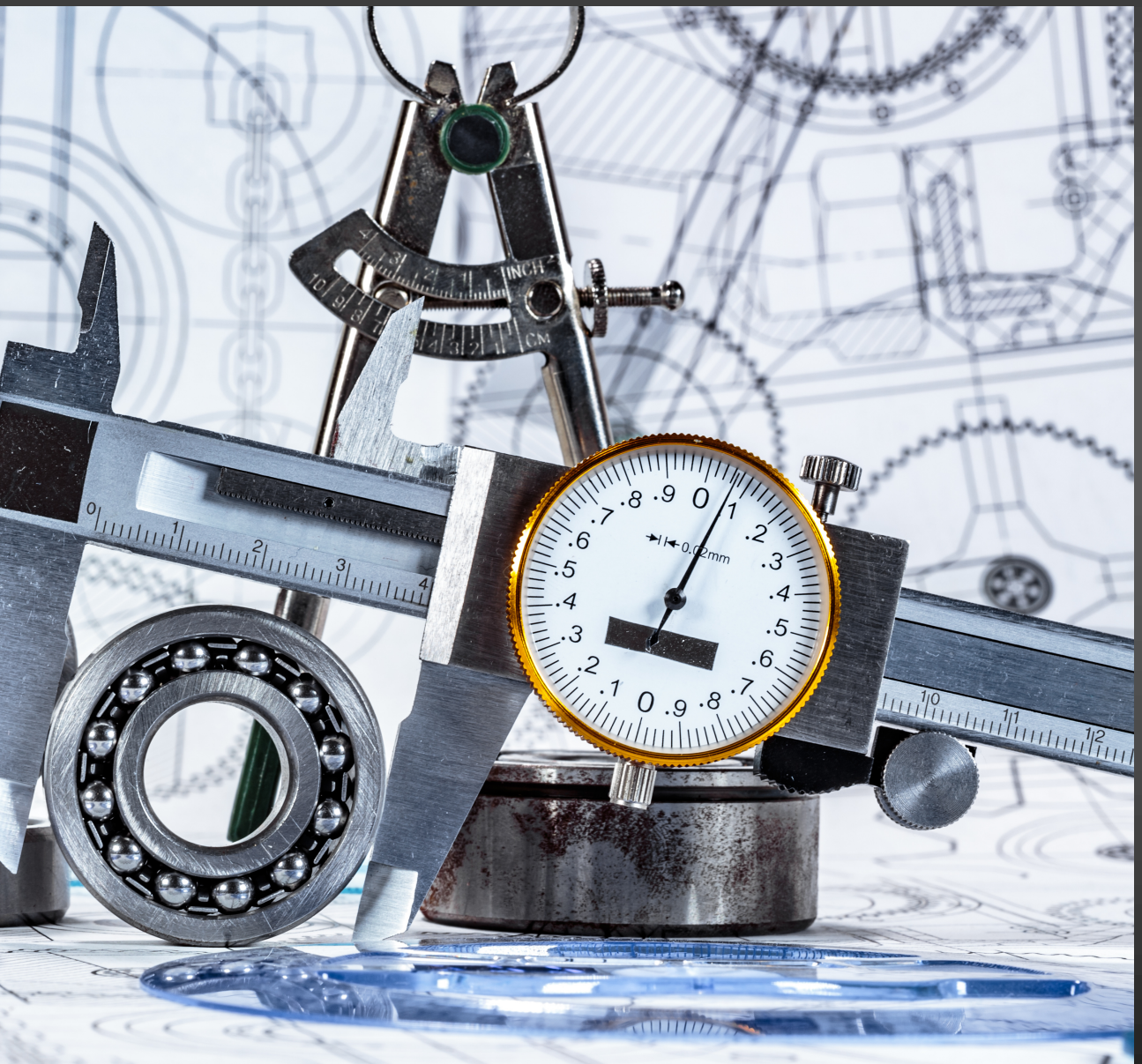
How to Accurately Measure Distributed Application Performance

Heinz Schaffner


Science Projects
vs. Production
Testing to
Destruction
vs. Distressed
Processing

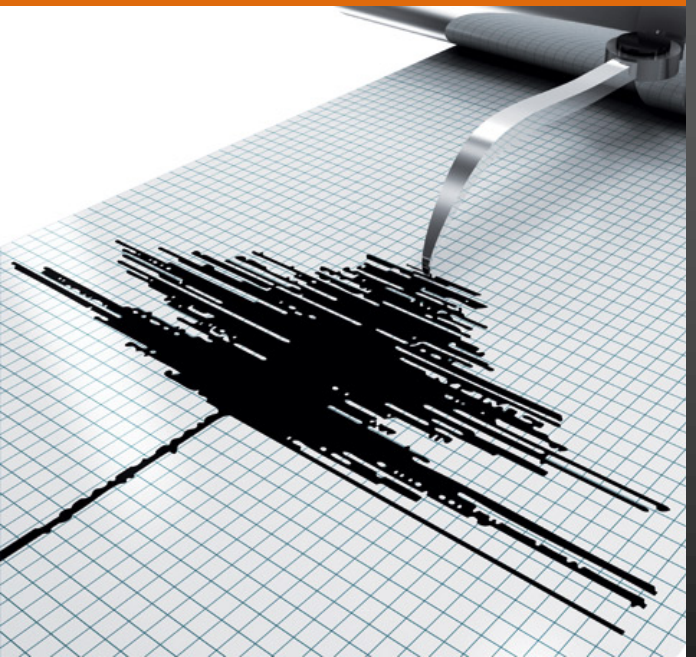
- Latency
- Schemes for generating test data
- Persistence Issues

Accuracy vs. Precision



- **Java nanoTime()**
  Nanosecond precision, not resolution
- **currentTimeMillis()**
  Granularity depends on OS
- **Network Analyzers**
  Measure latency over the wire not “API to API”

Myth:
"You cannot accurately get per Message Event Time Stamps"



- **Many vendors hide jitter**
- **C or JNI call will give precise time using TIC Register:**

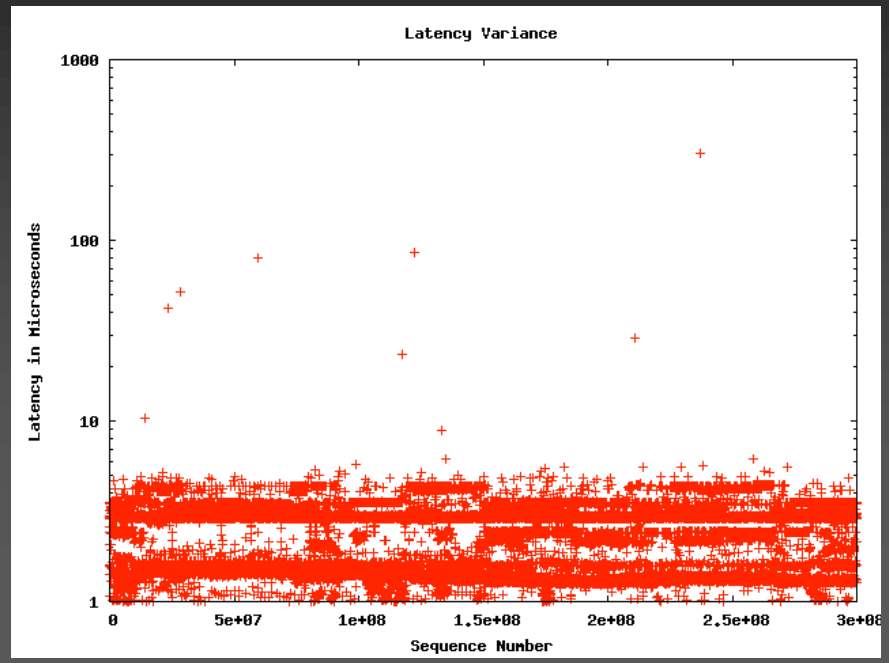
```
#elif _LINUX_X86_64
  if (USE_CLOCK_TICKS) {
     UINT32 hi, lo;
     __asm__ __volatile__ ("rdtsc" : "=a"(lo), "=d"(hi));
     return ( (UINT64)lo)|( ((UINT64)hi)<<32 );
  } else {
     return getTimeInUs();
```

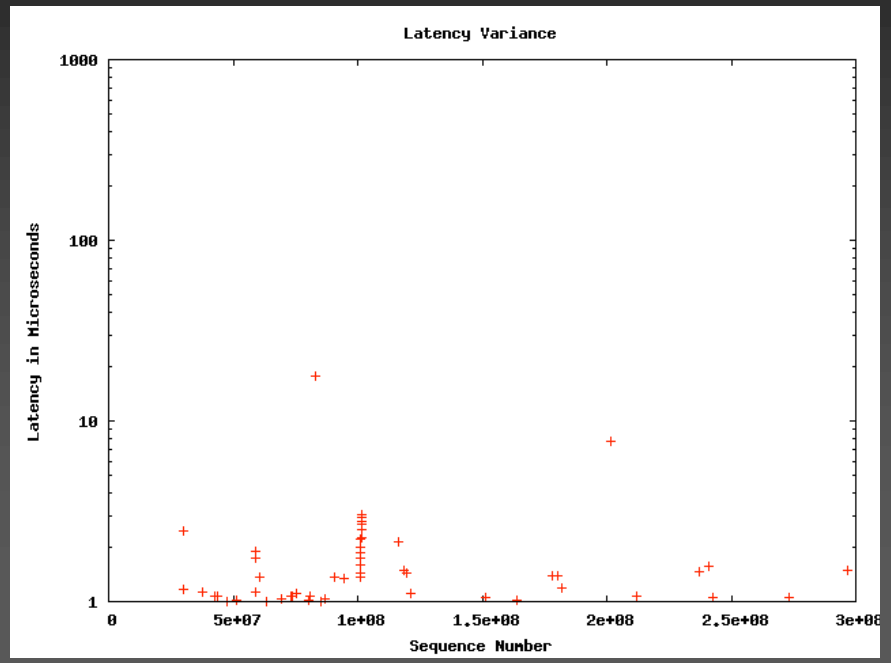
- **Only store values *of interest*** in pre-defined array and do calculations or save data at end of test (10-13 ns/per record)

Solace Systems®

Myth:

"Using Production Tuning to run benchmark will provide production comparison of vendors"



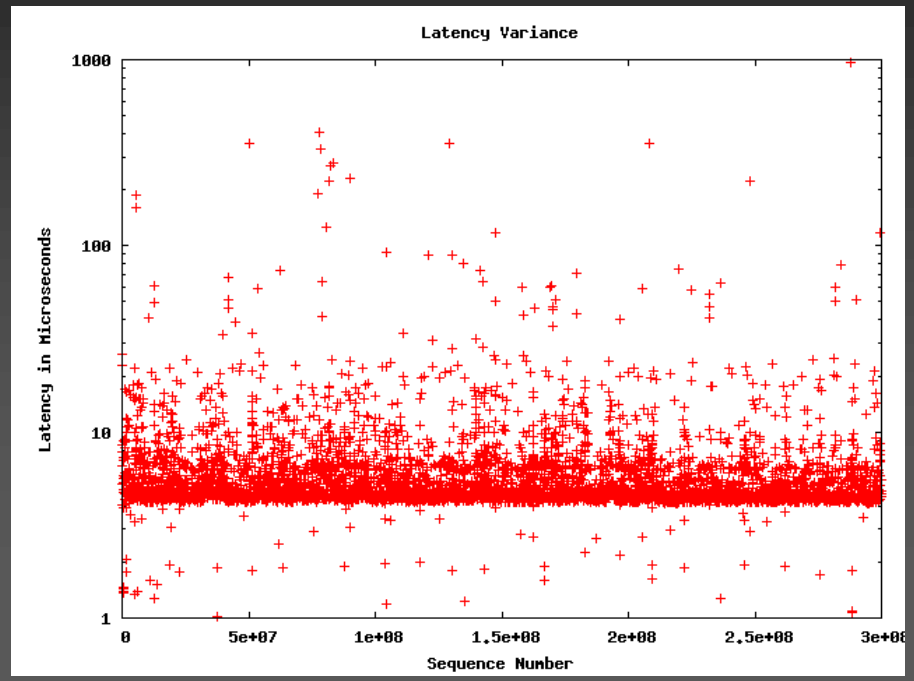


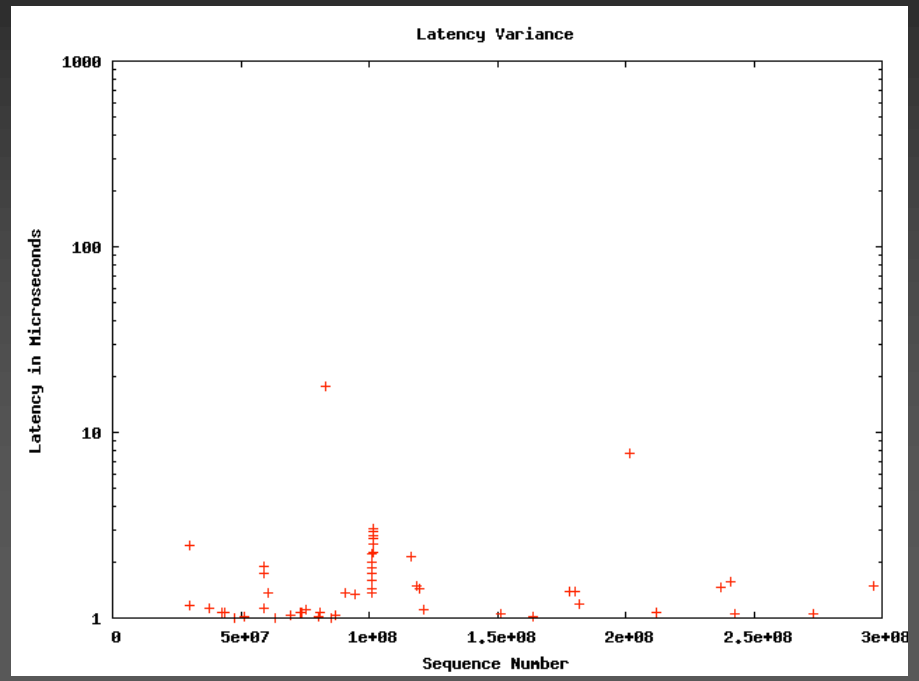
- Host setup and tuning have huge affect on system-induced processing overhead
- Accuracy affected by time-stepping hardware interrupts, etc.
- Use benchmarking as chance to review host and network tuning.

Solace Systems®

Myth:

"Applications in the cloud will have same performance as in bare metal hosts"



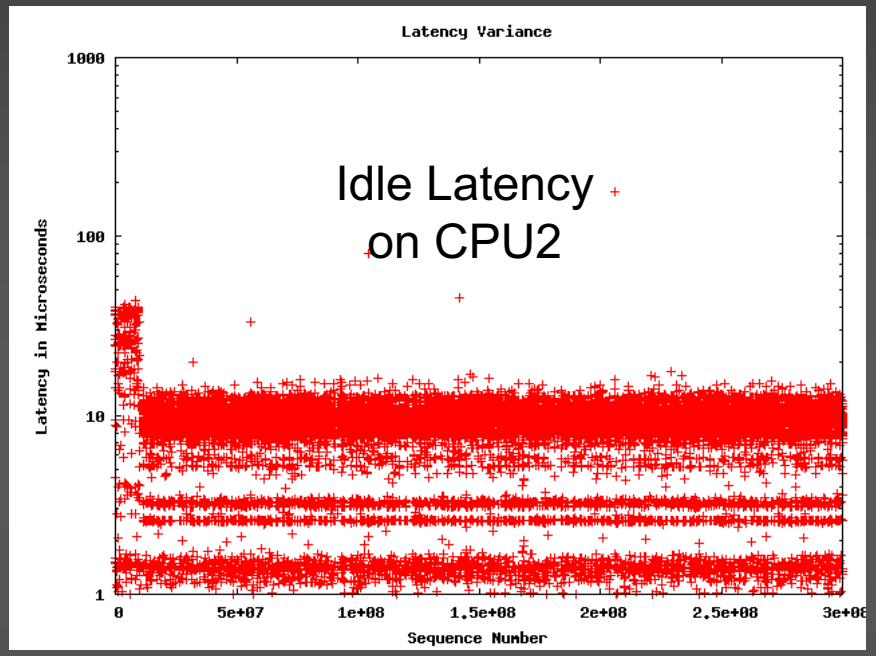


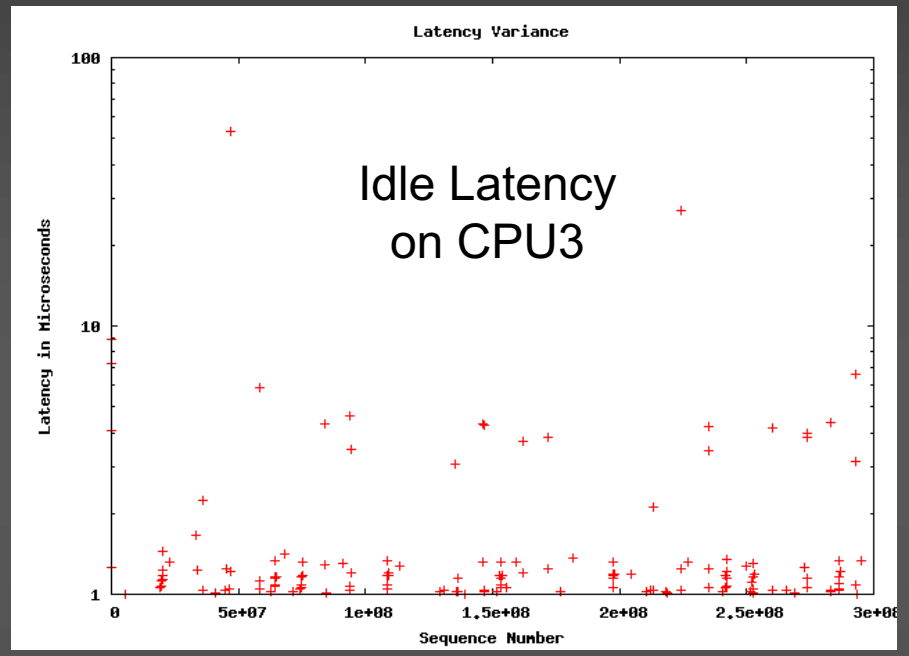
- OpenStack VM took 12% longer to complete test with 4 vCore and 8 Gig VM
- For this test Cloud hosts were idle
- Can't tune VM

Solace Systems®

Why does no one worry about CPU affinity when testing or in production – alternative to VM?

Idle latency test on CPU2 and CPU3 while CPU1 does file transfer

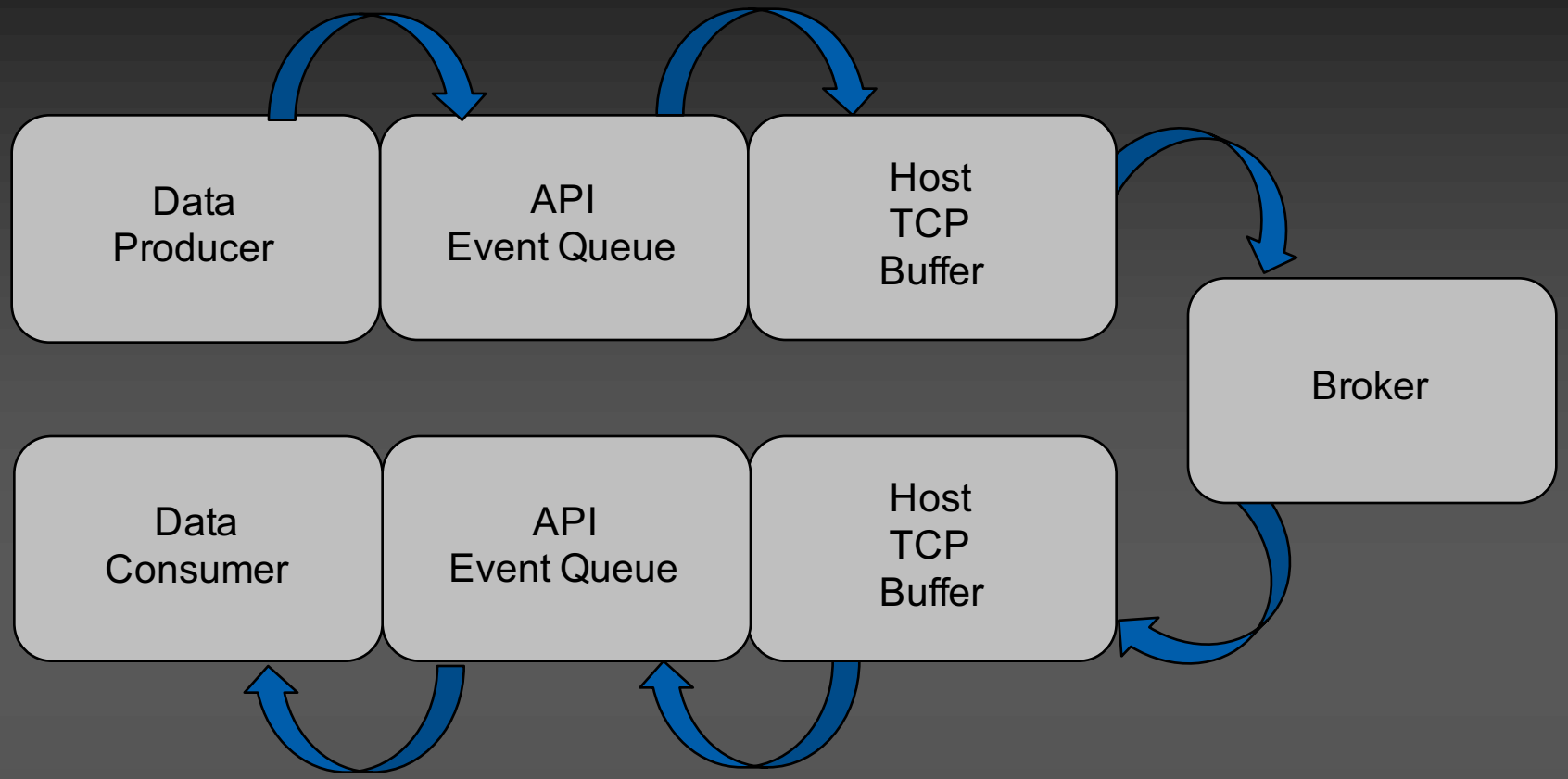




- "/proc/interrupts" shows all network hardware interrupts are on CPU2
- Don't run or benchmark distributed applications on high interrupt cores

Solace Systems®

Vendors love tests that Timestamp, send Test Data, Timestamp and divide total by Timestamp Delta



- Distributed systems = linked chain of queues and buffers.
- Each vendor provide customer processing
- Getting Timestamp Delta when Producer is done shows no jitter

Solace Systems®

Skewing of Results Using Timestamp Delta Technique

API Event Queue

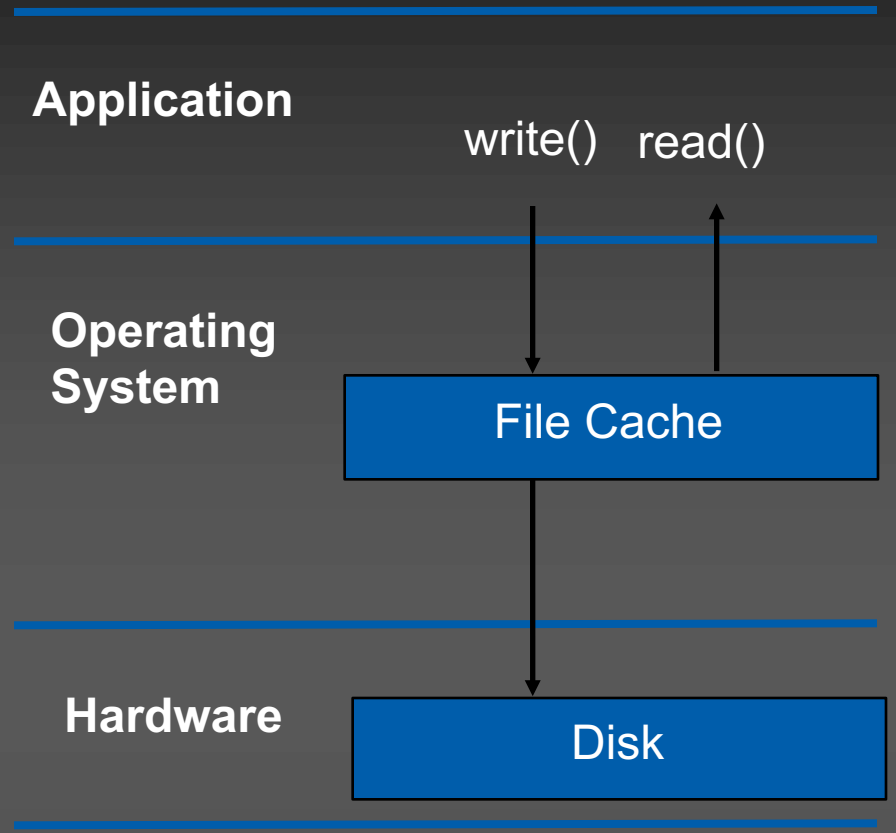
- **Huge Buffers**
  Varies by vendor and usually tunable, can show throughput/rate 50% higher than reality
- **Nagle-like Processing**
  Outbound delay buffering with vectored send. Only works if back-to-back sends and test to destruction

TCP and Broker Buffers

- **Massive Buffers**
  Leaves unprocessed data uncounted.
- **Persistent Queues**
  benchmarking with messages still in queue
- **Unidirectional tests**
  vs. bidirectional reality
- **Shared Broker Access**
  w/ no background load on the broker

Persistent Messaging's dirty little secret(s)

- Buffered write skews benchmarks
- Synchronous writes slower by 80%
- Read first from file cache and goes to disk if no cache hit
- File cache can affect hosts as memory becomes scarce

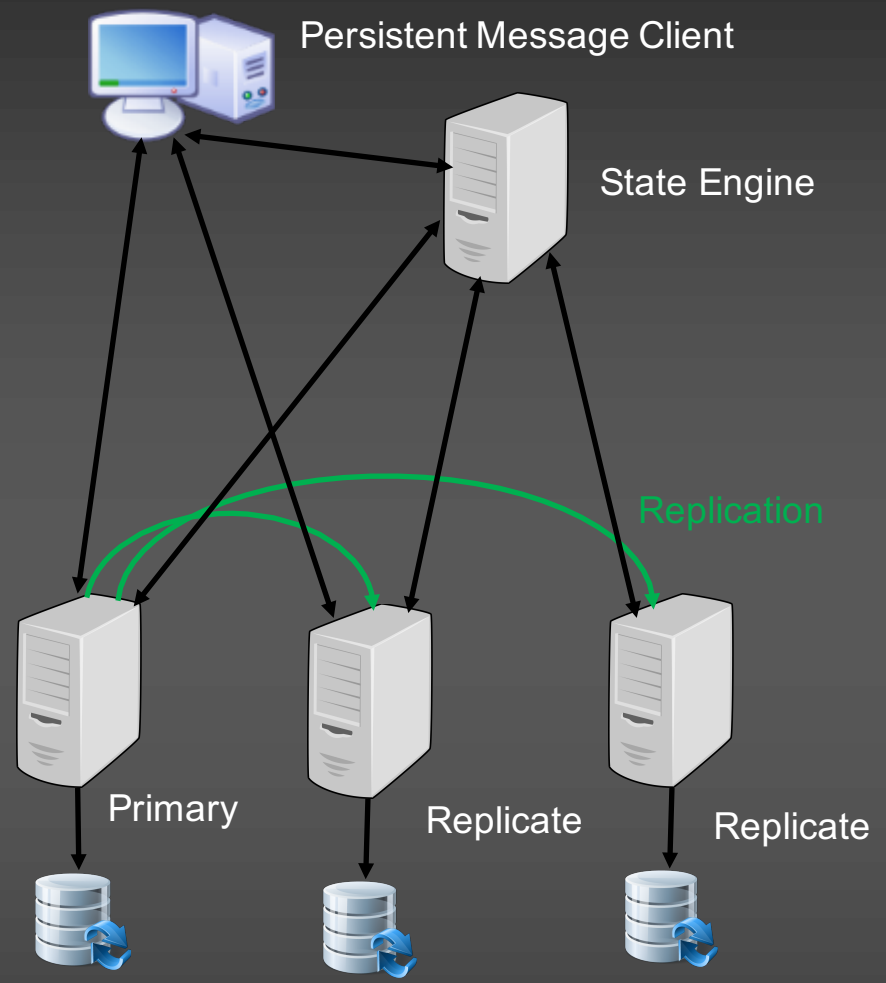


Benchmarking Message Broker and Persistence Issues

- Most vendors use buffer writes as default
- Slow consumers result in read cache misses
- Synchronous writes/reads
  - Pre-emptive
  - Interrupts
  - Context switches
- Testing Clients and Broker on same host eliminates I/O contention
- To increase synchronous disk write performance disk is pre-allocated and swap-like writes are used – test ungraceful crash – it is potential to lose all persisted data.

Distributed Quorum-based Persistence; Watch QoS setup!!

- Buffered writes locally mean better persistence throughput
- Requires replication, at expense of multiple network writes per message
- Replicates are check-pointed on timed basis to reduce overhead but can be 10 seconds out of sync
- QoS can define when and which members of the cluster ack
- Game of probabilities; faster persistence, lower QoS

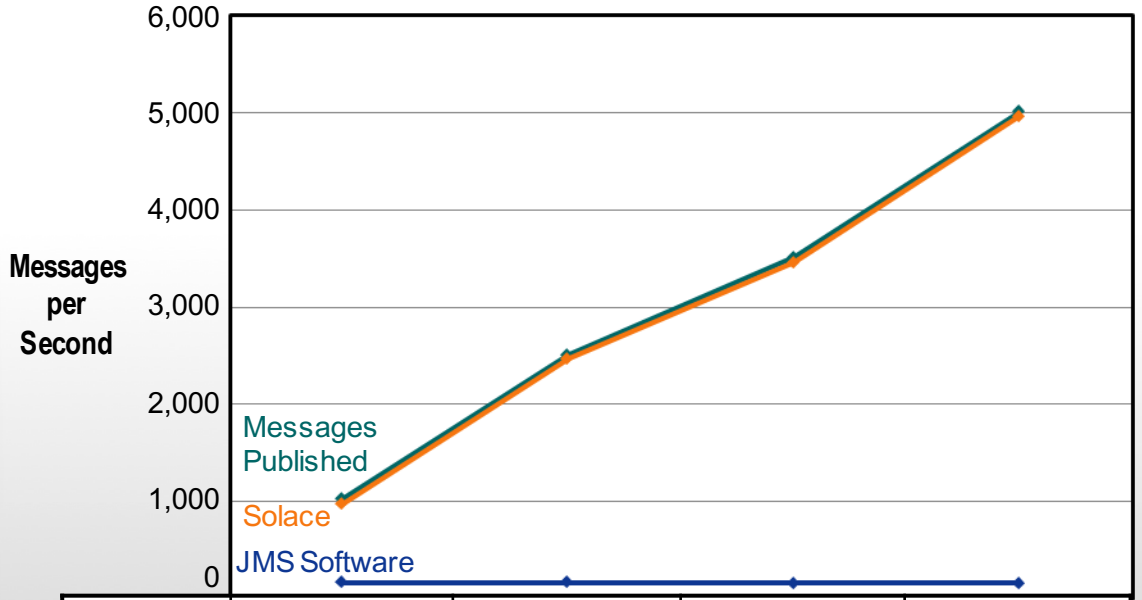


Benchmarking Big Data Applications & Infrastructure

- Benchmark generators don't capture duplication of persistent data to queues
- Duplicate (or more) sends greatly affect performance
  - 2x network traffic and hardware interrupts on brokers.
  - 2x file synch (or replicated) writes, and cache usage
  - If writing to HDFS then the slow consumer issue comes into play and you lose all read cache hits and slow performance of duplicate application queue.
  - Using of non-exclusive queues for scaling causes fan-out issues which affects broker performance.
- Big Data applications allow elastic scaling
  - For topic data this buys you nothing if one topic is over-used
  - Testing different topics on different applications instance is a science project

When the speed of light slows you down: WAN versus LAN

- Distance and TCP Slow Start
- Can't expect LAN speeds over WAN just because bandwidth is there
- WAN simulators = must use network errors and bandwidth throttling



Published	1,000	2,500	3,500	5,000
Solace Rate	1,000 (100%)	2,496 (99.8%)	3,498 (99.9%)	4,992 (99.8%)
JMS Rate	160 (16%)	160 (6.4%)	159 (4.5%)	159 (3%)

Solace Systems®

Some General Issues

- HA Clustering DR, Monitoring and Security/ACL
- Virus Checkers
- Equivalent hardware
- Don’t send back-to-back messages
- Disable Nagle’s
- Don’t bypass API event queue
- Not always fair to use same test with multiple vendors, but try
- Throw away first 30 second, (esp. w/ Java)
- Quick test runs may skew results

Questions?

If you don't have questions here is a quiz to fill the time while others ask questions. What is wrong with the following?

$A = B$	Initial Statement
$A^2 = AB$	Multiply both sides by A
$A^2 – B^2 = AB – B^2$	Subtract B^2 from both sides
$(A + B)(A – B) = B(A – B)$	Factor
$A + B = B$	Divide both sides by $(A – B)$
	What! I though A = B?
	What's wrong above?

Solace Systems®

goto; conference

Please

Remember to rate this session

Thank you!

follow us @gotochgo