# “Unifying *Event Tracing* and *Statistical Sampling*”

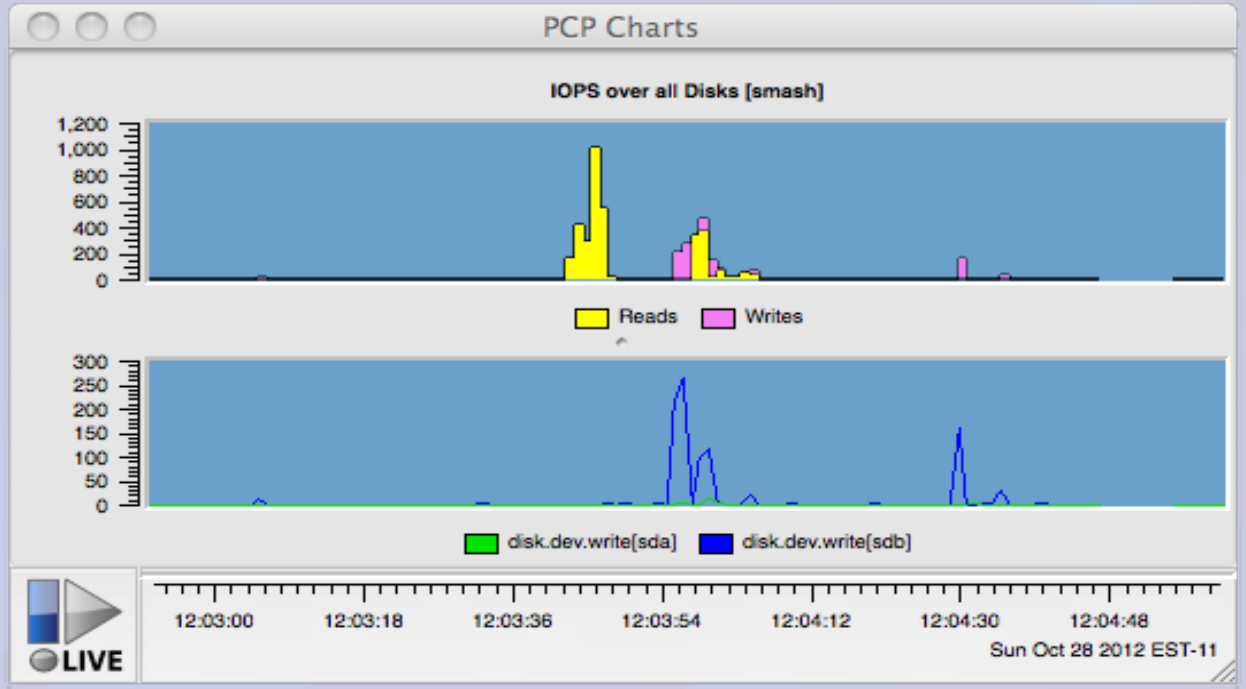
**Nathan Scott**

# Overview

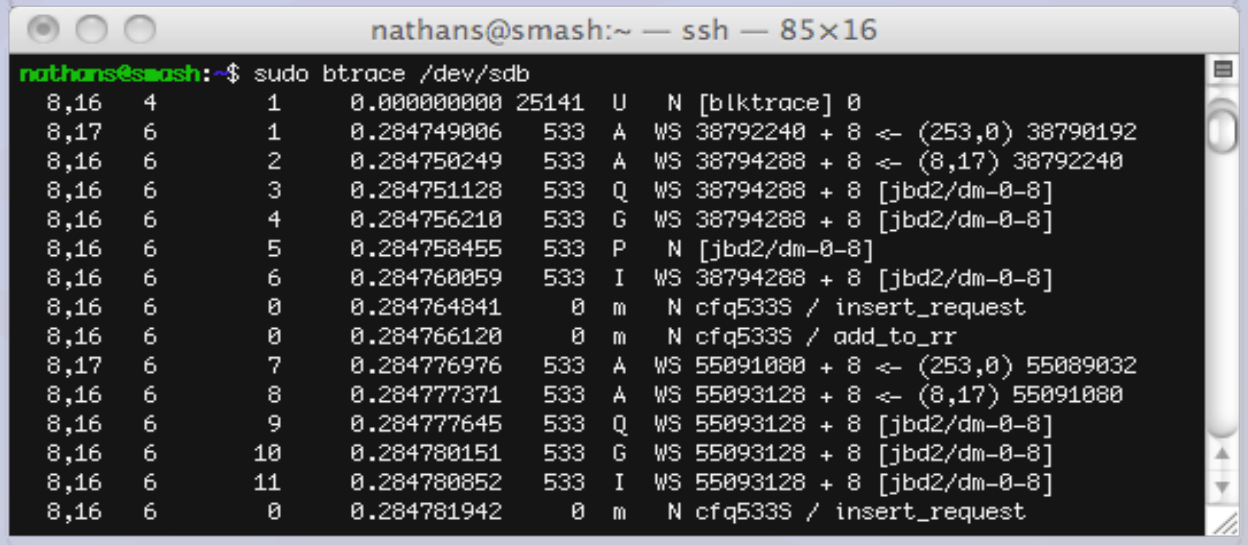
- Terminology
- Problem space
- Review
- Approach
- Case studies
- Conclusions
- Crash & Burn – Demo!

# Some Terminology...

“Sampled Values”

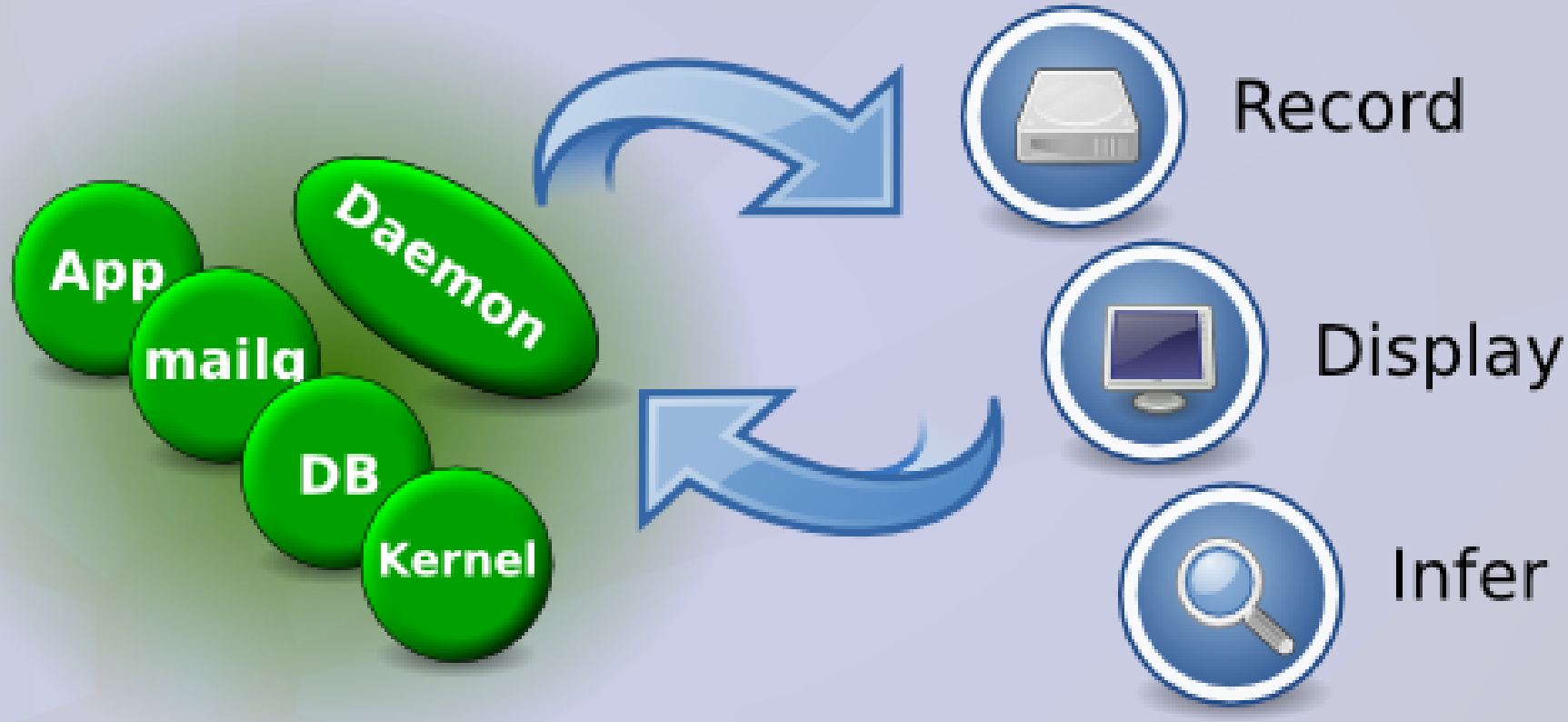



“Event Traces”



# More Terminology...

"Independent Domains"




"**Collector**" (server)

"**Monitor**" (client)

# What's The Problem?

- System-level performance analysis
- Predominantly sampling based today
  - Flexible  (retrospective, modelling, generic)
  - Granularity becomes an issue
- Tracing gaining popularity
  - Active research area  (relatively)
- Discontinuity
  - Disjointed tools, analysis “flow” broken

# Review

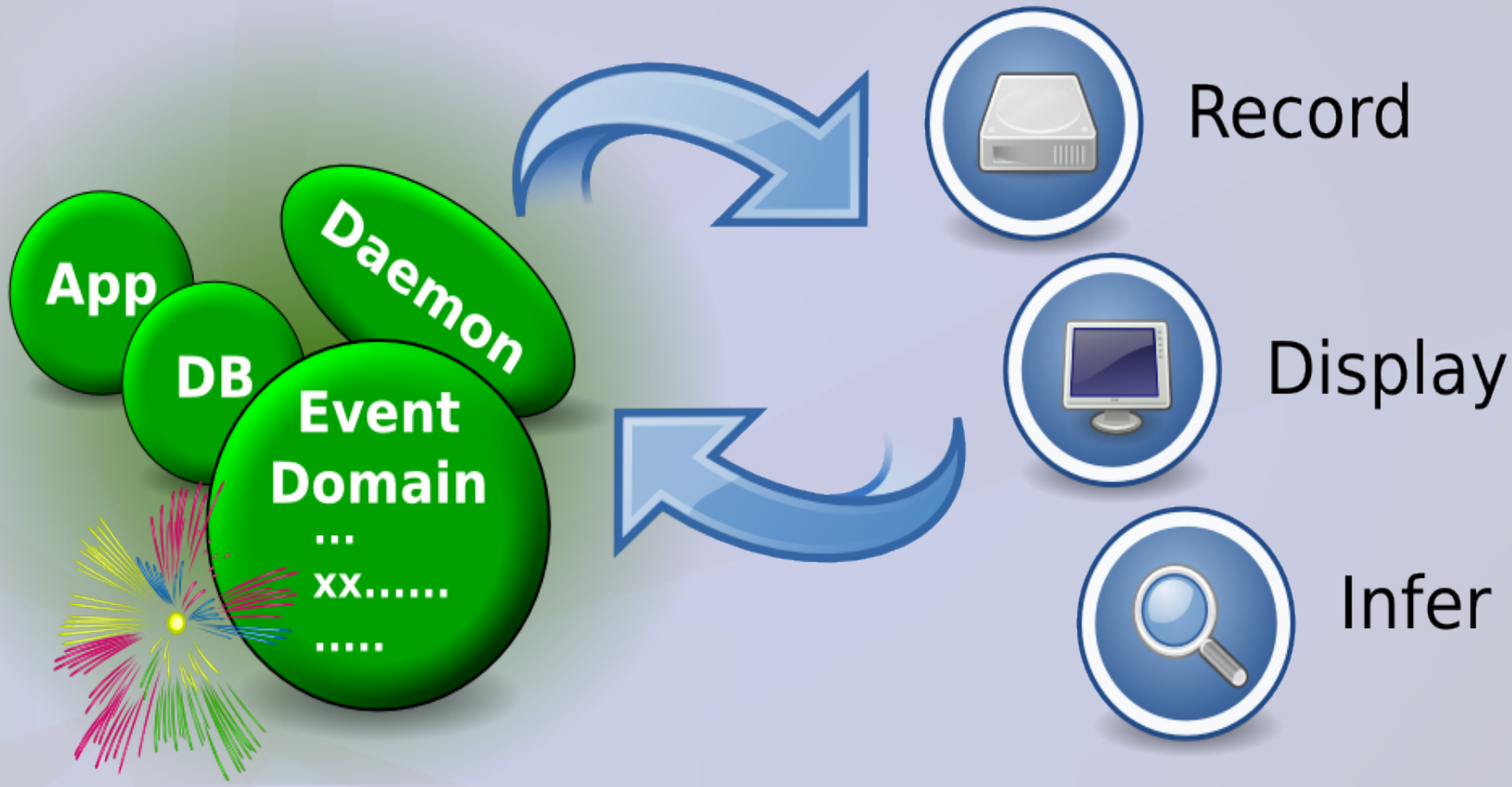
- Magpie and ETW
- DTrace and SystemTap
- X-Trace
- Dapper
- Performance Co-Pilot

- Production environments
- Visualisation techniques

# Approach

- Building on PCP:
  - Existing sampling-based toolkit
  - With recent tracing extensions
- Case-study based research
  - Multiple instrumentation points for each case
  - Extend/refine monitor tools based on findings

# Event Queues



# Case Studies

- Distributed Log Streaming
  - Sampling and tracing agent
  - Importance of server-side event filtering
- Data Warehouse Import
  - Shell script instrumentation
  - Visualisation
    - Charts with event trees and spans
    - Event selection

# Conclusions

- Event Parameters
  - Security model impact
  - Identifier significance
- Event filtering
  - Server side explored
  - Client side needed (and for archives)
- Visualisation
  - Trace structure explored (identifiers)
  - Selection model differences

# Quick example – parent shell

```
#!/bin/bash
. /etc/pcp.sh

wired()
{
        ./test-wired.sh $@
        sleep $1
}

pcp_trace on
for (( count=0 ; count < 10 ; count++ ))
do
        wired 2 $count
        [ $count -eq 5 ] && ./test-flood.sh $count
done
```

# Quick example – children

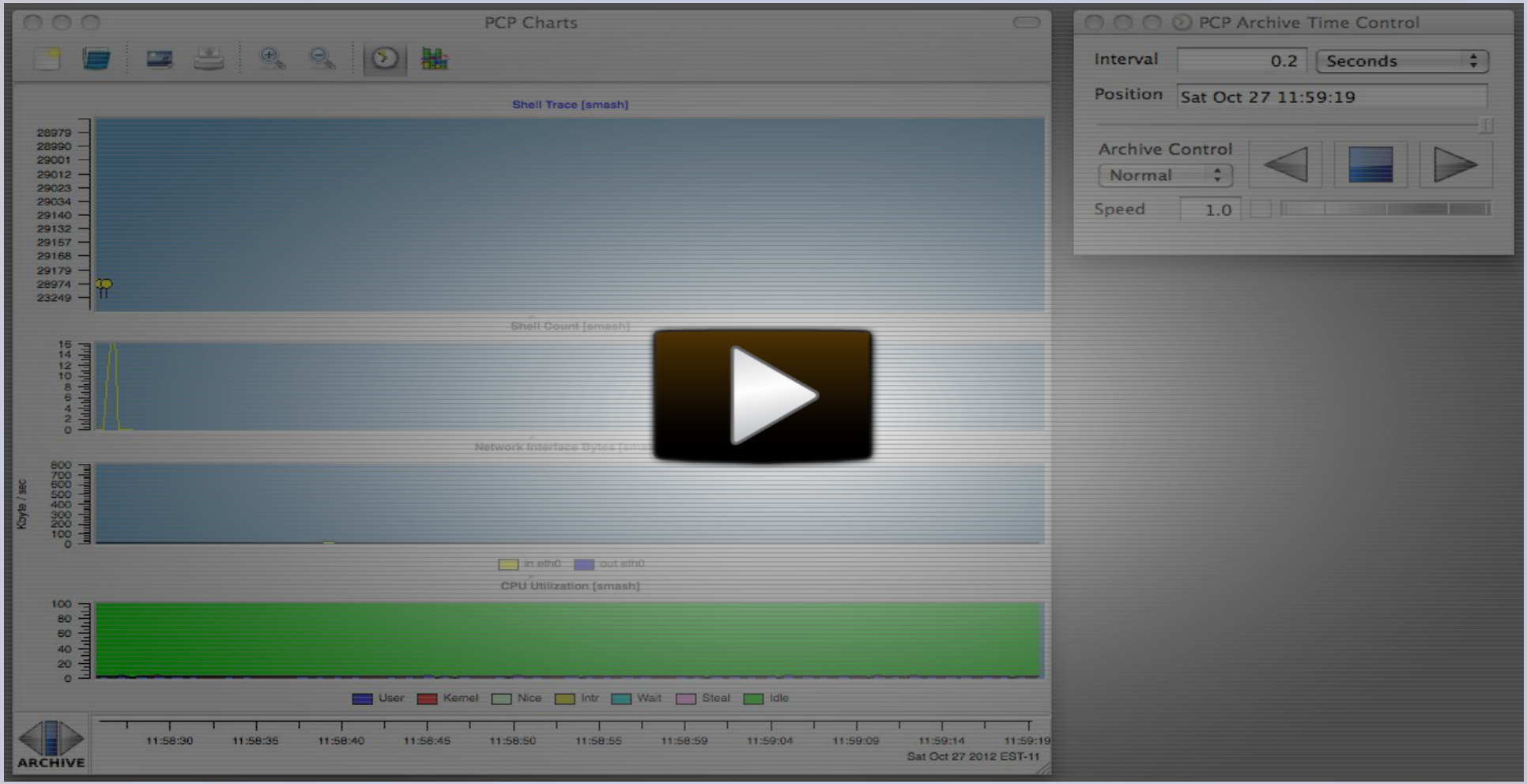
- **test-flood.sh:**

```
#!/bin/bash
. /etc/pcp.sh
pcp_trace on
sudo ping -q -f -i 0.00002 -c 30000 verge
```

- **test-wired.sh:**

```
#!/bin/bash
. /etc/pcp.sh
pcp_trace on
for i in {0..4} ; do
        ./cpuburn &
done
wait
```

# Quick example - visually




**http://screencast.com/t/Xa5wmEq7**