

**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**

**LEARN. NETWORK.  
EXPERIENCE OPEN SOURCE.**



# Selecting the Right Tools for Performance Analysis

William Cohen  
Performance Tools Engineer, Red Hat  
06.27.12

**SUMMIT**

JBoss  
WORLD

PRESENTED BY RED HAT



To measure is to know.  
-Lord Kelvin

**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**



If you can not measure it,  
you can not improve it.  
-Lord Kelvin

**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**



# The Importance of Correct Metrics

- Have standard econobox car
- Want it “faster”

**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**



# The Ugly Result of Using the Wrong Metric



<http://commons.wikimedia.org/wiki/File:Poser.JPG>



Except where otherwise noted, this work is licensed under  
<http://creativecommons.org/licenses/by-sa/3.0/>

**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**



# What are your important metrics?

- Response time for a query
- Maximum number of simultaneous requests
- Memory required per query
- Time spent in key sections of code



# Acceptable measurement cost?

- Zero cost
- Low cost
- Cost is no object

**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**





# When is it “Good Enough”?

- Fine as is
- Nice if a bit better
- Need to meet some specific performance metric
- Never “good enough”

**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**



# Want data that guide actions

- Some metrics show there is a problem
- Want metrics to show how or where to fix the problem

**SUMMIT**

JBoss  
WORLD

PRESENTED BY RED HAT



# Map your metrics to tool metrics

- Your metrics probably not the same as the tool's metrics
- Need to translate into available tool measurements
- Typical tool metrics:
  - Software events (for example page faults)
  - Latency
  - Code hotspots
  - Processor microarchitecture performance



# What region metrics measured?

- Single process
- User-space
- Kernel-space
- System-wide

**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**



# Many Linux Performance Tools

- Tools developed to address specific issues
- Provide different:
  - Measurements
  - Performance impact
  - Filtering
  - Data analysis

**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**



# Instrumentation Techniques

**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**



# Statistical Sampling

- Every Nth (+/-) sample record details
- Number of samples should be proportional to events at that location
- Reduces overhead
- Often used with hardware performance counters



# Probing

- Predefined points in software:
  - kernel tracepoints
  - user-space markers
- Use breakpoints instructions:
  - Kprobes
  - Uprobes
- Only encounter cost when hit
- Can be used on already running code





# Binary Rewrite

- Examine the code as it is loaded
- Take machine code apart
- Add instrumentation and generate new code
- Allow instrumentation of code without recompilation of source
- Cannot attach to already running apps
- Can significantly slow the startup and execution of code



# Shared Library Wrapper

- Inserts a wrapper function between caller and callee in shared library
- Lower overhead than binary rewrite
- Limit to function call boundaries
- Cannot attach to already running apps
- Implemented with LD\_PRELOAD=libwrapper.so



# Mapping Data Back to Source Code

- Most performance tools collect instruction addresses
- Want to map instruction address to source code
- Debuginfo (-g) provides mapping
- Debuginfo does not affect runtime performance
- Can strip out debuginfo to make executables smaller



# Mapping Data Back for Interpreters

- Instruction address maps to runtime
- Just-In-Time translation no debug info for address
- Some interpreters provide profilers:
  - Python (python -m cProfile.py ..)
  - Perl (perl -d:DProf ...)
  - Ruby (ruby-prof ...)
- Some interpreters have hooks for other tools:
  - Java jvmti and jvmpi interfaces



# Where are the processors spending time?

**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**



# Why?

**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**



# OProfile

- Useful for finding code hotspots
- Not useful for I/O or latency problems
- Uses performance monitoring hardware
- Performs sampling
- Collects system-wide data
- Requires root privileges
- Can record data for Java applications
- Supports a wide range of processors
- Available on most distributions

**SUMMIT**

JBoss  
WORLD

PRESENTED BY RED HAT



# Perf

- Provides access to:
  - performance monitoring hardware
  - Kernel software events
- Data metrics:
  - Count events (stats)
  - Trace of samples (record)
- Available to normal user
- Requires newer Linux kernels ( $\geq 2.6.32$ )

**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**





# What about watching events across processes?

**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**



# What about watching events in both user- and kernel-space?

**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**



# SystemTap

- Can instrument already running apps
- Probes system-wide (both user- and kernel-space)
- Very flexible filtering
- Requires special privileges

**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**



# Why is so much memory being used?

**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**



# Valgrind

- Originally a tool to look at memory issues:
  - Allocation, initialization, and freeing
- Actually a very flexible binary rewrite framework:
  - Memory diagnostics, memory allocation, cache simulator, threading
- Instruments shared libraries, but not kernel code
- Slows application (4x-100x slower)
- Run as normal user
- Cannot attach to already running code



# Valgrind (cachegrind) Example

```
valgrind --tool=cachegrind stap -p4 disktop.stp
```

...

```
==10672==
```

```
==10672== | refs:    4,556,135,689
```

```
==10672== |l misses:    2,192,463
```

```
==10672== LLi misses:    30,208
```

```
==10672== |l miss rate:    0.04%
```

```
==10672== LLi miss rate:    0.00%
```

...

**SUMMIT**

JBoss  
WORLD

PRESENTED BY RED HAT



# Valgrind (cachegrind) Example (cont.)

==10672==

==10672== D refs: 1,846,734,454 (1,161,695,711 rd + 685,038,743 wr)

==10672== D1 misses: 15,404,912 ( 13,300,378 rd + 2,104,534 wr)

==10672== LLd misses: 8,070,771 ( 6,160,076 rd + 1,910,695 wr)

==10672== D1 miss rate: 0.8% ( 1.1% + 0.3% )

==10672== LLd miss rate: 0.4% ( 0.5% + 0.2% )

==10672==

==10672== LL refs: 17,597,375 ( 15,492,841 rd + 2,104,534 wr)

==10672== LL misses: 8,100,979 ( 6,190,284 rd + 1,910,695 wr)

==10672== LL miss rate: 0.1% ( 0.1% + 0.2% )

**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**



# Kcachegrind GUI

cachegrind.out.10672 [../install/bin/stap -p4 testsuite/systemtap.examples/io/desktop.stp]

File View Go Settings Help

Open Back Forward Up % Relative Cycle Detection Relative to Parent Instruction Fetch

Flat Profile

Search: (No Grouping)

Self	Function	Location
22.24	__libdw_find_attr	dwarf_child.c
8.33	__libdw_form_val_len	libdw_form.c
8.18	dwarf_siblingof	string3.h, dwarf_siblingof.c
6.83	int_malloc	malloc.c
4.17	std::tr1::hash<std::string...	basic_string.h, functional_h...
3.20	int_free	malloc.c
3.06	std::tr1::Hashtable<std::...	basic_string.h, char_traits.h...
2.78	dwarf_tag	dwarf_tag.c
2.22	std::basic_string<char, st...	basic_string.h, basic_string....
1.92	malloc	dl-minimal.c, malloc.c
1.76	__libdw_findabbrev	dwarf_tag.c
1.65	Dwarf_Abbrev_Hash_find	dynamicsizehash.c
1.62	lookup	dynamicsizehash.c
1.25	dwflpp::global_alias_cachi...	dwflpp.cxx, basic_string.h, ...
1.15	malloc Consolidate	malloc.c
1.14	std::basic_string<char, st...	basic_string.h, atomicity.h
1.12	relocate.7402	relocate.c
1.07	lexer::scan(bool)	parse.cxx, util.h, basic_ios....
1.04	__libdw_formref	dwarf_formref.c
1.01	(unknown)	syscall-template.S, memcp...
0.99	std::tr1::__detail::Map_ba...	basic_string.h, char_traits.h...
0.94	std::string::append(char c...	basic_string.h, basic_string....

(unknown)

Types	Callers	All Callers	Callee Map	
Event Type	Incl.	Self	Short	For
Instruction Fetch	1.01	1.01	Ir	
L1 Instr. Fetch Miss	12.06	12.06	ILmr	
ILmr	5.92	5.92	ILmr	
Data Read Access	3.96	3.96	Dr	
L1 Data Read Miss	0.70	0.70	D1mr	
DLmr	0.02	0.02	DLmr	
Data Write Access	0.00	0.00	Dw	
L1 Data Write Miss	0.00	0.00	D1mw	
DLmw	0.00	0.00	DLmw	

Profile Part	Incl.	Self	Called	Comment
--------------	-------	------	--------	---------

Parts Calleees Call Graph All Calleees

cachegrind.out.10672 [1] - Total Instruction Fetch Cost: 4 556 135 689



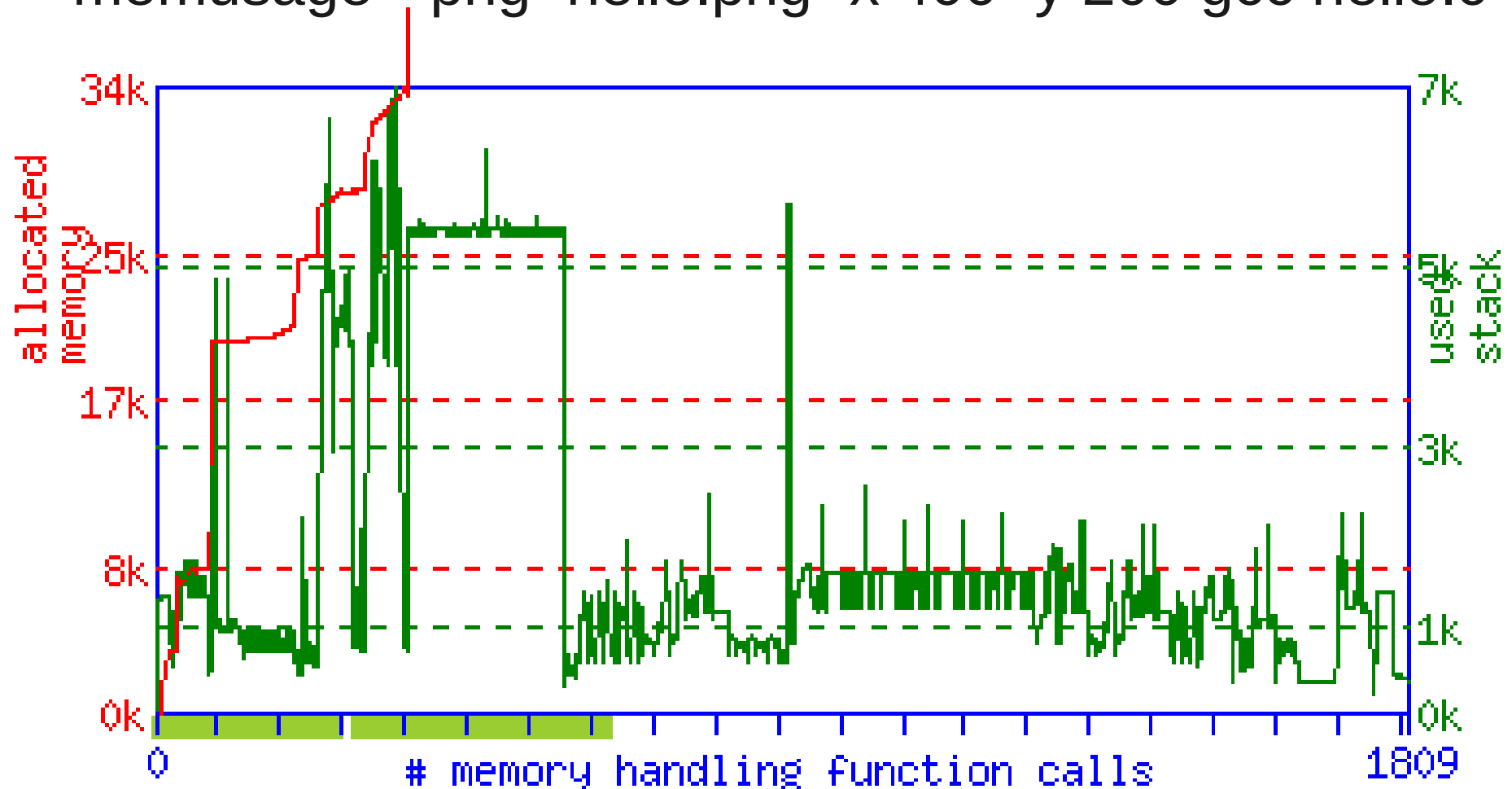
# Glibc-utils

- Analysis more limited than Valgrind:
  - Only analyzes memory allocation/free operations
  - No checks for use of uninitialized memory
- Faster than valgrind:
  - Intercepting calls for memory allocation/free
  - Not rewriting and instrumenting the binary code
- Tools in the glibc-utils rpm



# Glibc memusage demonstration

```
memusage --png=hello.png -x 400 -y 200 gcc hello.c
```



**SUMMIT**

JBoss  
WORLD

PRESENTED BY RED HAT



# Graphical User Interface?

- Eclipse IDE
- Plug-ins:
  - Systemtap
  - Oprofile
  - Valgrind
- Navigate data by point and click



# Further Information

- Developer Guide on <http://docs.redhat.com/docs/en-US/index.html>
- Oprofile - <http://oprofile.sourceforge.net/news/>
- Perf – perf help
- SystemTap - <http://sourceware.org/systemtap/>
- Valgrind - <http://valgrind.org/>
- glibc utils – memusage –help

**SUMMIT**

**JBoss  
WORLD**

**PRESENTED BY RED HAT**



# Stay connected through the Red Hat Customer Portal

Instrumenting the Linux Kernel with SystemTap

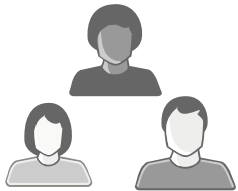


Review Tech brief

Performance Issues in Red Hat Enterprise Linux (Part 1)



Review Tech brief



Join Red Hat Enterprise Linux

Join Group

[access.redhat.com](https://access.redhat.com)



**SUMMIT**

**JBoss  
WORLD**

PRESENTED BY RED HAT



**LIKE US ON FACEBOOK**

[www.facebook.com/redhatinc](http://www.facebook.com/redhatinc)

**FOLLOW US ON TWITTER**

[www.twitter.com/redhatsummit](http://www.twitter.com/redhatsummit)

**TWEET ABOUT IT**

#redhat

**READ THE BLOG**

[summitblog.redhat.com](http://summitblog.redhat.com)

**GIVE US FEEDBACK**

[www.redhat.com/summit/survey](http://www.redhat.com/summit/survey)

**SUMMIT**

**JBoss  
WORLD**

PRESENTED BY RED HAT

