Ichnaea: A Parallel Statistical Profiler

Matt Redmond

Massachusetts Institute of Technology

December 12, 2011

Introduction to Profiling

Introduction to Profiling

- *Deterministic Profiling*

Introduction to Profiling

- *Deterministic Profiling*
- Python Profiling tools

Introduction to Profiling

- *Deterministic Profiling*
- Python Profiling tools
- GCC instrumentation

Introduction to Profiling

- *Deterministic Profiling*
- Python Profiling tools
- GCC instrumentation
- *Statistical Profiling*

Introduction to Profiling

- *Deterministic Profiling*
- Python Profiling tools
- GCC instrumentation
- *Statistical Profiling*
- perf subsystem

Introduction to Profiling

- *Deterministic Profiling*
- Python Profiling tools
- GCC instrumentation
- *Statistical Profiling*
- perf subsystem
- *OProfile*

What is OProfile?

What is OProfile?

It's an *Event Driven Statistical Profiler*

What is OProfile?

It's an *Event Driven Statistical Profiler*
Choose a supported event and a sampling count.

What is OProfile?

It's an *Event Driven Statistical Profiler*
Choose a supported event and a sampling count.
CPU increments counter for each time that event occurs (simplistic model).

What is OProfile?

It's an *Event Driven Statistical Profiler*
Choose a supported event and a sampling count.
CPU increments counter for each time that event occurs (simplistic model).
When counter value is a multiple of count, look at the stack frame.

What is OProfile?

It's an *Event Driven Statistical Profiler*
Choose a supported event and a sampling count.
CPU increments counter for each time that event occurs (simplistic model).
When counter value is a multiple of count, look at the stack frame.
Write this data to a log file somewhere.

Why Statistical?

Why Statistical?

Low overhead.
Only reasonable choice for cluster-scale profiling.
Some interesting data unavailable through other methods.

Interesting Data

Interesting Data

- CPU_CLK_UNHALTED: Clock cycles when not halted

Interesting Data

- CPU_CLK_UNHALTED: Clock cycles when not halted
- INST_RETIRED_ANY_P: Number of instructions retired

Interesting Data

- CPU_CLK_UNHALTED: Clock cycles when not halted
- INST_RETIRED_ANY_P: Number of instructions retired
- LLC_MISSES: Last level cache misses
  (L1 Miss = 18 cycles, L2 Cache Miss = 600 cycles)

Interesting Data

- CPU_CLK_UNHALTED: Clock cycles when not halted
- INST_RETIRED_ANY_P: Number of instructions retired
- LLC_MISSES: Last level cache misses
  (L1 Miss = 18 cycles, L2 Cache Miss = 600 cycles)
- MUL / DIV: Number of multiplies / Number of divides

Interesting Data

- CPU_CLK_UNHALTED: Clock cycles when not halted
- INST_RETIRED_ANY_P: Number of instructions retired
- LLC_MISSES: Last level cache misses
  (L1 Miss = 18 cycles, L2 Cache Miss = 600 cycles)
- MUL / DIV: Number of multiplies / Number of divides
- BR_MISSP_EXEC: Branch mispredictions

Interesting Data

- CPU_CLK_UNHALTED: Clock cycles when not halted
- INST_RETIRED_ANY_P: Number of instructions retired
- LLC_MISSES: Last level cache misses
  (L1 Miss = 18 cycles, L2 Cache Miss = 600 cycles)
- MUL / DIV: Number of multiplies / Number of divides
- BR_MISSP_EXEC: Branch mispredictions
- CPI: Cycles per Instruction (how efficient is my program?)

Why Ichnaea?

Why Ichnaea?

No good open source parallel CPU profilers available.

Why Ichnaea?

No good open source parallel CPU profilers available.
Simple command line invocation ("feels *Unixy*")

Why Ichnaea?

No good open source parallel CPU profilers available.
Simple command line invocation ("feels *Unixy*")

```
ichnaea ~/hostfile.txt event_name /remote/path/to/mycommand ~/report.txt
```

High Level Overview

High Level Overview

(1) On each host in the hostfile, load the OProfile kernel module and select event_name to profile.

High Level Overview

(1) On each host in the hostfile, load the OProfile kernel module and select event_name to profile.

(2) Run the parallel program mycommand.

High Level Overview

(1) On each host in the hostfile, load the OProfile kernel module and select event_name to profile.

(2) Run the parallel program mycommand.

(3) Unload the kernel module

High Level Overview

(1) On each host in the hostfile, load the OProfile kernel module and select event_name to profile.

(2) Run the parallel program mycommand.

(3) Unload the kernel module

(4) Aggregate individual output files into one file.
(todo: implement aggregation as map-reduce)

Setup

Setup

Need sudo access on each machine

Setup

Need sudo access on each machine
Need passwordless SSH set up to each machine.

Setup

Need sudo access on each machine
Need passwordless SSH set up to each machine.
yum install oprofile (each machine)

Setup

Need sudo access on each machine
Need passwordless SSH set up to each machine.
yum install oprofile (each machine)
Copy the ichnaea folder to / (each machine)

Sample Program

```
#include <stdio.h>
long foo(long i, long j) {
    return 2*(i%7);
}

long bar(long i, long j) {
    double jDi = j / i;
    return (long) (i ^ j) * (jDi * jDi);
}
int main() {
    long i, j = 0;
    while(i< 1000000000) {
        j = foo(i, j);
        j = bar(i, j);
        i++;
    }
    printf("%l", j);
    return 0;
}
```

Sample Output

```
Counted BRANCH_RETIRED events (retired branches) with a unit mask of 0x0c (multiple flags) count 50053
samples  %        app name                 symbol name
83334    58.5445  dostuff                  bar
47824    33.5977  no-vmlinux               /no-vmlinux
2786      1.9572  libstdc++.so.6.0.13      /usr/lib/libstdc++.so.6.0.13
1738      1.2210  dostuff                  foo
1425      1.0011  dostuff                  main
569       0.3997  libc-2.11.2.so           free
555       0.3899  libpthread-2.11.2.so     pthread_mutex_lock
542       0.3808  libc-2.11.2.so           malloc
460       0.3232  libpthread-2.11.2.so     __pthread_mutex_unlock_usercnt
324       0.2276  libpthread-2.11.2.so     __pthread_enable_asynccancel
151       0.1061  oprofiled                /usr/bin/oprofiled
144       0.1012  libc-2.11.2.so           __strlen_sse2
92        0.0646  python2.6                /usr/bin/python2.6
87        0.0611  libc-2.11.2.so           __i686.get_pc_thunk.bx
84        0.0590  bash                     /bin/bash
83        0.0583  libpthread-2.11.2.so     __nanosleep_nocancel
78        0.0548  libc-2.11.2.so           _int_malloc
76        0.0534  libc-2.11.2.so           memcpy
73        0.0513  libpthread-2.11.2.so     __pthread_disable_asynccancel
```

Sample Output from Julia's perf.j Benchmark

```
Counted LLC_MISSES events count 10053
samples  %        app name                 symbol name
120      15.0943  no-vmlinux               /no-vmlinux
68        8.5535  julia                    jl_gc_collect
49        6.1635  libc-2.13.so             /lib/x86_64-linux-gnu/libc-2.13.so
30        3.7736  julia                    ptrhash_lookup_bp
20        2.5157  julia                    gc_markval_
11        1.3836  julia                    llvm::InstCombiner::DoOneIteration(llvm::Function&, ...
8         1.0063  julia                    allocobj
7         0.8805  julia                    llvm::ConstantUniqueMap<llvm::ExprMapKeyType, llvm:::...
7         0.8805  julia                    llvm::ScheduleDAGSDNodes::NewSUnit(llvm::SDNode*)
6         0.7547  julia                    TwoAddressInstructionPass::runOnMachineFunction(llvm:...
6         0.7547  julia                    llvm::SelectionDAGISel::SelectCodeCommon(llvm::SDNode*, ...
6         0.7547  julia                    void llvm::Calculate<llvm::Function, llvm::BasicBlock*>...
6         0.7547  libLAPACK.so             dgemm_beta_PENRYN
6         0.7547  libglib-2.0.so.0.2800.6  /lib/x86_64-linux-gnu/libglib-2.0.so.0.2800.6
5         0.6289  julia                    MachineCSE::ProcessBlock(llvm::MachineBasicBlock*)
5         0.6289  julia                    SelectionDAGLegalize::LegalizeOp(llvm::SDValue)
```

Sample Output from Julia's perf.j Benchmark

```
Counted INST_RETIRED_ANY_P events (number of instructions retired) count 50053
samples  %        app name                 symbol name
14558     7.6507  libc-2.13.so             /lib/x86_64-linux-gnu/libc-2.13.so
13825     7.2655  no-vmlinux               /no-vmlinux
12013     6.3132  julia                    llvm::MachineInstr::addRegisterDead(unsigned int, llvm::...
9122      4.7939  julia                    anon (tgid:16034 range:0x7f73b056a000-0x7f73b05ea000)
7099      3.7307  libLAPACK.so             dgemm_kernel_PENRYN
4272      2.2451  julia                    gc_markval_
3029      1.5918  julia                    llvm::ComputeMaskedBits(llvm::Value*, llvm::APInt const&...
2884      1.5156  julia                    ptrhash_lookup_bp
2591      1.3616  julia                    llvm::InstCombiner::DoOneIteration(llvm::Function&, ...
2418      1.2707  julia                    llvm::MachineInstr::findRegisterDefOperandIdx(unsigned int...
2063      1.0842  julia                    jl_method_table_assoc_exact.clone.1
1976      1.0384  julia                    RALinScan::assignRegOrStackSlotAtInterval(llvm::LiveInterval*)
1968      1.0342  julia                    llvm::SmallPtrSetImpl::insert_imp(void const*)
1892      0.9943  julia                    llvm::LiveVariables::HandlePhysRegKill(unsigned int, ...
1775      0.9328  julia                    llvm::InstCombineWorklist::AddInitialGroup(llvm::Instruction*
1549      0.8140  julia                    llvm::MachineOperand::isIdenticalTo(llvm::MachineOperand ...
1499      0.7878  julia                    llvm::SelectionDAGISel::SelectCodeCommon(llvm::SDNode*, ...
1400      0.7357  julia                    relocate
1387      0.7289  julia                    apply_cl
1324      0.6958  julia                    jl_subtype_le
```

Future Research

Future Research

- Benchmark Julia's hardware performance against other languages.

Future Research

- Benchmark Julia's hardware performance against other languages.
- Get Ichnaea running on virtual machines (XenOProfile).

Future Research

- Benchmark Julia's hardware performance against other languages.
- Get Ichnaea running on virtual machines (XenOProfile).
- Map-reduce aggregation of individual output files for large cluster.

Future Research

- Benchmark Julia's hardware performance against other languages.
- Get Ichnaea running on virtual machines (XenOProfile).
- Map-reduce aggregation of individual output files for large cluster.

Will be releasing source with the final paper (a few bugs to hammer out first).

Future Research

- Benchmark Julia's hardware performance against other languages.
- Get Ichnaea running on virtual machines (XenOProfile).
- Map-reduce aggregation of individual output files for large cluster.

Will be releasing source with the final paper (a few bugs to hammer out first).
Any questions?

References

- Google Wide Profiling: A Continuous Profiling Infrastructure for Data Centers
  http://research.google.com/pubs/archive/36575.pdf
- OProfile Documentation
  http://oprofile.sourceforge.net/docs/
- OProfile Overhead Data
  http://oprofile.sourceforge.net/performance/
- XenOProfile (attempts to port to EC2 will need this)
  http://xenoprof.sourceforge.net/xenoprof_2.0.txt
- Julia Project
  https://github.com/JuliaLang/julia