OpenPOWER™

15 Tools in 30 minutes

Paul Clarke, Software Architect / Developer
IBM

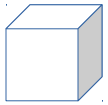
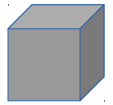
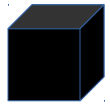
OpenPOWER

Agenda

Advance Toolchain	
XL C, C++, and Fortran compilers	
FDPR	
SDK for Linux on Power	
SDK Migration Advisor	ma
SDK Source Code Advisor	sca
SDK CPI Breakdown & Drill-down	cpi
curt	
Power Functional Simulator	
Performance Simulator (sim_ppc)	
pipestat	
LPCPU	
pveclib	
SPHDE	

OpenPOWER

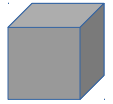
Legend

requirement for best value	white box	gray box	black box
source code changes	✔		
source code access	✔	✔	
recompile	✔	✔	
object access	✔	✔	
relink	✔	✔	

OpenPOWER

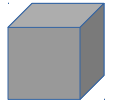
Advance Toolchain



- IBM Advance Toolchain for Linux on Power
- **Latest open source compilers, runtime, and tools**
- **Enabled and optimized for the latest POWER processor**
  - Quarterly updates, security fixes
- Supported
- Open source
- http://developer.ibm.com/linuxonpower/advance-toolchain

OpenPOWER

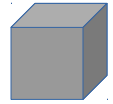
Advance Toolchain



- Advance Toolchain 11.0-3
  - GCC 7.3.1
  - glibc 2.26 (libc, libm, libpthread, ...)
  - binutils 2.29
  - Boost 1.64
  - libdfp, PAFlib, gdb, Python, golang, GFortran, OProfile, Valgrind, itrace, TBB, Userspace RCU, libhugetlbfs, zlib, OpenSSL, TCMalloc, SPHDE, ...

OpenPOWER

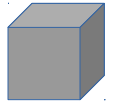
Advance Toolchain



component	Red Hat Enterprise Linux 7.4	Red Hat Developer Toolset 7	Advance Toolchain 11.0-3
GCC	4.8.5	7.2.1	7.3.1
GNU libc (glibc)	2.17	-	2.26
binutils	2.25	2.28	2.29
Boost	1.53	-	1.64
runtime dependency?	-	no	**yes**

OpenPOWER

XL C, C++, and Fortran compilers



- IBM's flagship compiler suite
- Proprietary
- Two editions:
  - Licensed, supported
  - Community Edition (**free, full-function**, unsupported)
- Used on AIX, z/OS, SPEC publishes
- New “clang” frontend for better source-code compatibility with GCC and LLVM and new language standards
- https://www.ibm.com/developerworks/downloads/r/xlcpluslinux/

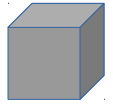
OpenPOWER

XL C, C++, and Fortran compilers

- Automatic parallelization
- GPU acceleration
- OpenMP 3.1 and partial OpenMP 4.5
- High-performance libraries (MASS, BLAS)
- XML, HTML reports for further optimization insights

OpenPOWER

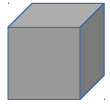
FDPR



- Feedback-Directed Program Restructuring
- IBM Research
- Post-link binary optimization tool
- Simple 3-step process: instrument, profile, optimize
- Gray box: requires preservation of relocation information at link-time
- Free download

OpenPOWER

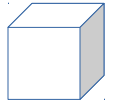
FDPR



```
$ gcc -o load -Wl,--emit-relocs load.c

$ /opt/ibm/fdprpro/bin/fdpr_instr_prof_opt load
/home/pc/summit/load-2.1pc/load.instr
FDPR profiling: /home/pc/summit/load-2.1pc/load.instr  …
Total run time for 10000 iterations was: 72.699327 seconds

$ ./load
Total run time for 10000 iterations was: 8.755640 seconds

$ ./load.fdpr
Total run time for 10000 iterations was: 8.162184 seconds
```

OpenPOWER

FDPR

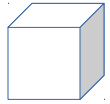


- FDPR journal: suggests source changes for performance improvement (white box)

```
$ gcc -o load -Wl,--emit-relocs -g load.c

$ /opt/ibm/fdprpro/bin/fdpr_instr_prof_jour load
/home/pc/summit/load-2.1pc/load.instr
FDPR profiling: /home/pc/summit/load-2.1pc/load.instr  …
Total run time for 10000 iterations was: 72.699327 seconds
$ ls -1tr | tail -1
load_jour.xml
```

OpenPOWER

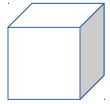
FDPR



```
$ cat load_jour.xml
[...]
                <operation name="Inline function">
                        <problem>High call overhead of a hot small function</problem>
                        <solution>Compiler: inline callee into caller - replace call to callee with its body</solution>
                        <site>
                                <ip>10000958</ip>
                                <dir>/home/pc/summit/load-2.1pc</dir>
                                <file>load.c</file>
                                <fn>main</fn>
                                <line>70</line>
                                <xcount>265801472</xcount>
                        </site>
                        <param name="callee">
                                <site>
                                        <ip>10000734</ip>
                                        <dir>/home/pc/summit/load-2.1pc</dir>
                                        <file>load.c</file>
                                        <fn>sum_add</fn>
                                        <line>18</line>
                                        <xcount>265801472</xcount>
                                </site>
                        </param>
                </operation>
```

OpenPOWER

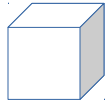
IBM SDK for Linux on Power



- Eclipse-based Integrated Development Environment (IDE)
- With IBM plug-ins for Linux on Power
- Runs on Linux x86 (or Linux VM on Windows or Mac) or Linux on Power
  - run IDE on Linux on Power, display back to laptop via remote X display, ssh X-Windows tunneling, or VNC
  - run entirely on x86, using cross-compilation and POWER emulation
  - run IDE on laptop, build and run on remote Power system
  - http://ibm.biz/ibmsdklop
    https://developer.ibm.com/linuxonpower/sdk
- Free download
- Tutorial: https://developer.ibm.com/linuxonpower/tutorials/sdk_linux_on_power/
- Videos: http://ibm.biz/linuxonpowervideos

OpenPOWER

IBM SDK for Linux on Power

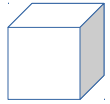


Basic operation:

1. Launch SDK
2. Create new project, import code
3. Run Migration Advisor
4. Build (note Build Advisor output)
5. Run Source Code Advisor
6. Run perf or OProfile
7. Run CPI Breakdown & Drill-down
8. Run Power Performance Advisor

OpenPOWER

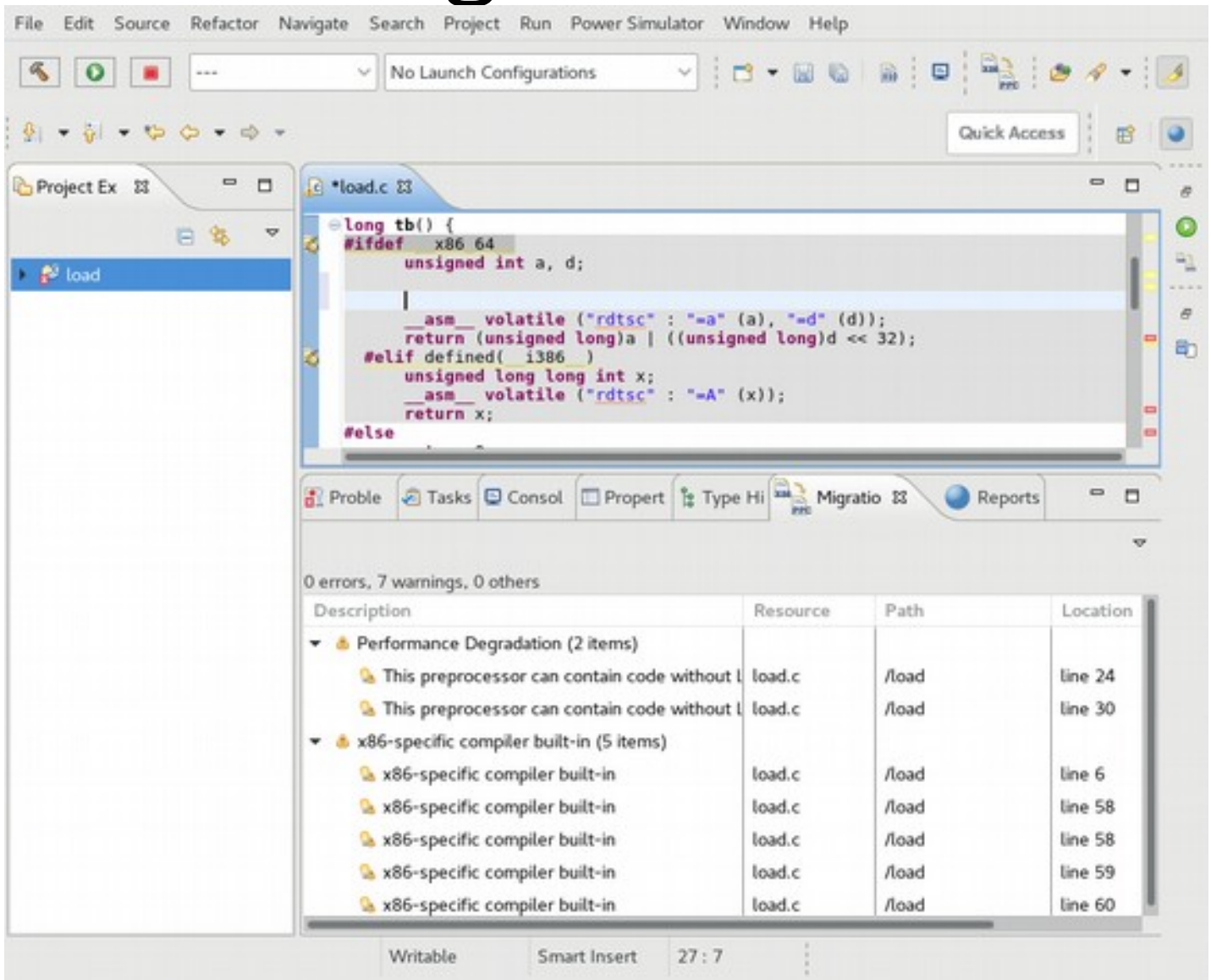
SDK: Migration Advisor



- SDK plug-in code-scanner
- Focus: Linux on x86 to Linux on Power
- Reports on portability issues and concerns
  - `#ifdef x86`
  - x86 syscalls, APIs, built-ins, assembly, pthread `_np`
  - `long double, Float128`
  - non-portable Hardware Transactional Memory
  - `char` type default signedness
  - 32/64 bit
  - `sync` built-ins
  - endian issues
- Simple: “Run Migration Advisor” from project's context menu
- Easy: Quick Fixes
- Efficient: Automated Quick Fix (!)

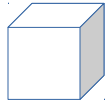
OpenPOWER

SDK: Migration Advisor



OpenPOWER

SDK: Migration Advisor



*load.c

```
#define MIGRATION_ADVISOR_BAIT
#ifdef MIGRATION_ADVISOR_BAIT
    {
        /* amazingly complex way to set sum=0 */
        __m128i a = _mm_set1_epi32(2);
        a = _mm_sub_epi32(a,a);
        sum
    }
    sum += t
#endif
```

x86-specific compiler built-in

1 quick fix available:

Built-in quick fix

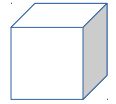
Press 'F2' for focus

Proble Task

```
_ADVISOR_BAIT

ngly complex way to set sum=0 */
a = _mm_set1_epi32(2);

#ifdef __PPC__
        a = (__vector signed char) vec_sub((__vector int) a, (__vector int) a)
#else
        a = _mm_sub_epi32(a,a);
#endif
        sum = _mm_cvtsi128_si32(a);
    }
    sum += tb();
```

OpenPOWER

SDK: Migration Advisor



Advanced configuration

Apply basic fixes automatically.

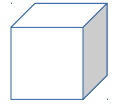
Migration Advisor

```
1 C/C++ files detected.
11 checkers enabled.
Analyzing load.c.
   5 x86-specific compiler built-in found.
      Fixed a possible issue at line 6.
      Fixed a possible issue at line 46.
      Fixed a possible issue at line 52.
      Fixed a possible issue at line 58.
Operation completed.
```

Stop Migration | Save Log

OpenPOWER

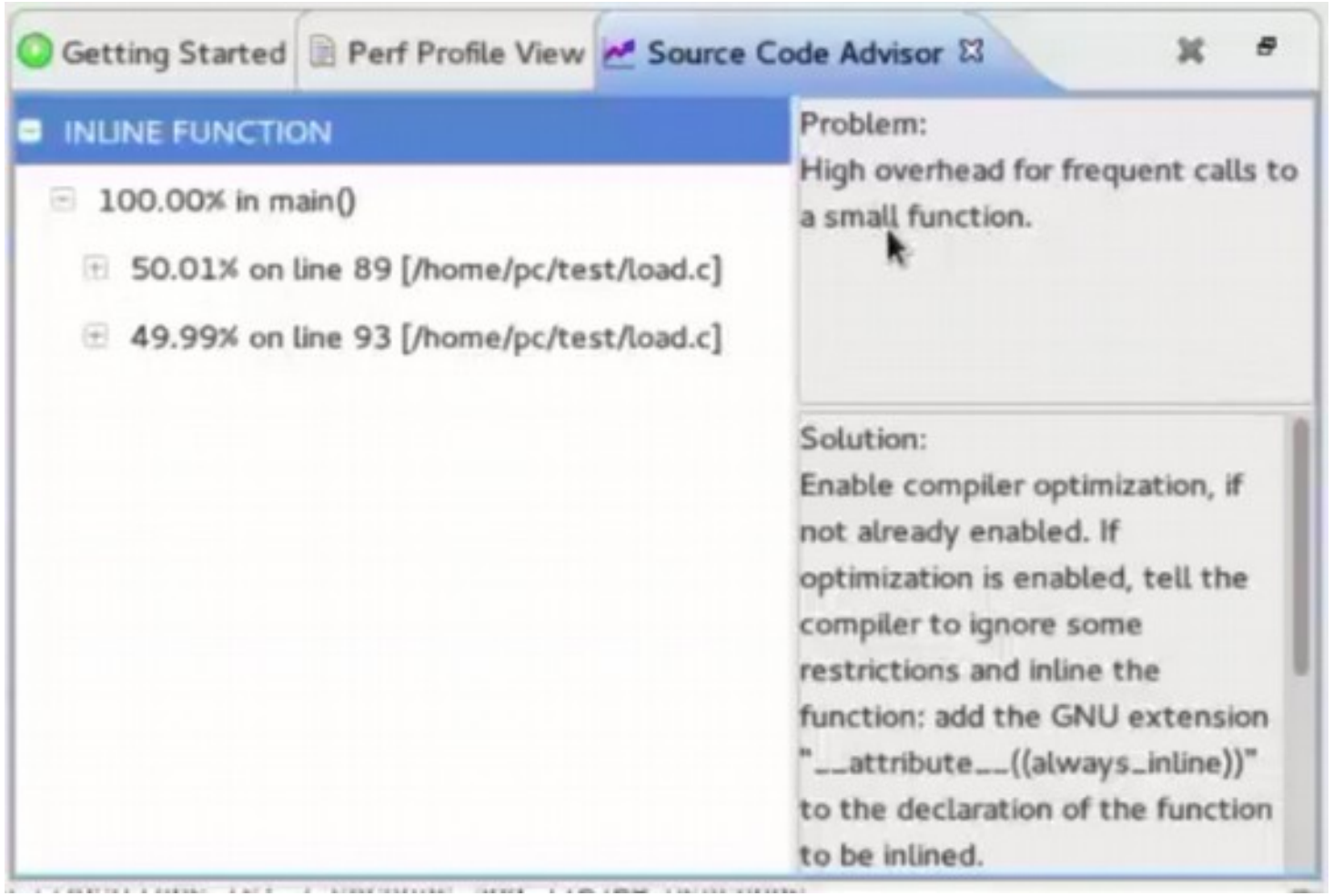
SDK: Source Code Advisor



- Run-time performance analysis and report suggested source code changes for improvement
- Leverages FDPR's journal
- Quick Fixes

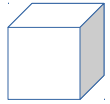
OpenPOWER

SDK: Source Code Advisor



OpenPOWER

SDK: Source Code Advisor



load.c load.c

```
87              }
88              if (direction == 1) {
89                  sum=sum_add(sum, array[j]);
90              }
91
92              if (direction == 0) {
93                  sum=sum_sub(sum, array[j]);
94              }
        }
```

High overhead for frequent calls to a small function.

1 quick fix available:

Inline Function Quick Fix (Function: "sum_sub")

Press 'F2' for focus

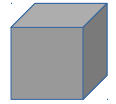
Problems Tasks Cons

load.c

```
ine NUM_ITER        10000
ine UPPER_LIMIT     1000000
ine LOWER_LIMIT     100

int sum_add(int sum, int value) {
    return (sum + value);
}

__attribute__((always_inline))
inline int sum_sub(int sum, int value) {
    return (sum - value);
}
```

OpenPOWER

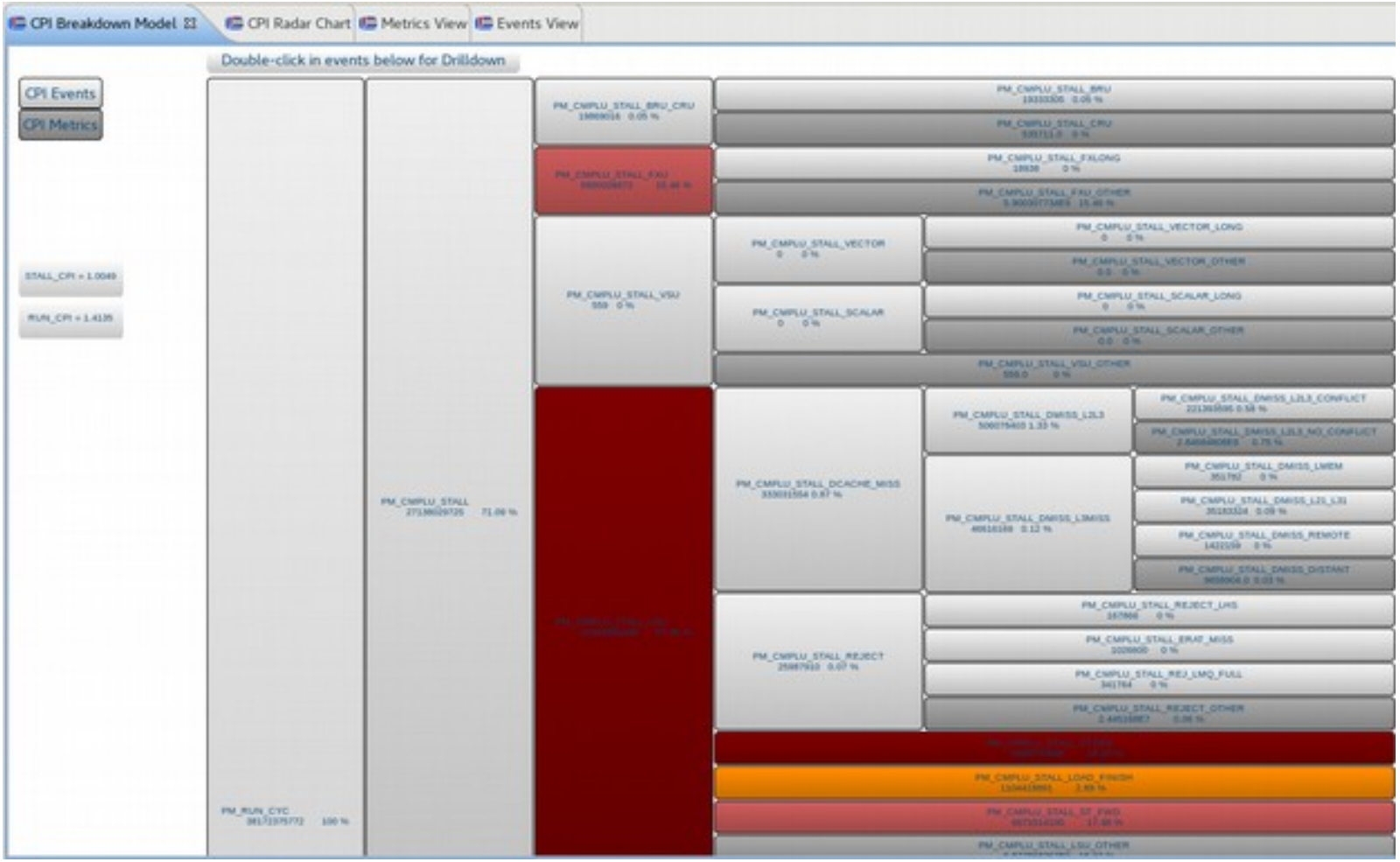
SDK: CPI Breakdown & Drill-down



- Cycles-per-Instruction: a measure of processor efficiency
- On average, how many cycles does it take to execute each instruction? Lower is better!
- CPI Breakdown will breakdown CPI measurement into a hierarchy of processor-specific event classes
- Note: does not currently support POWER9

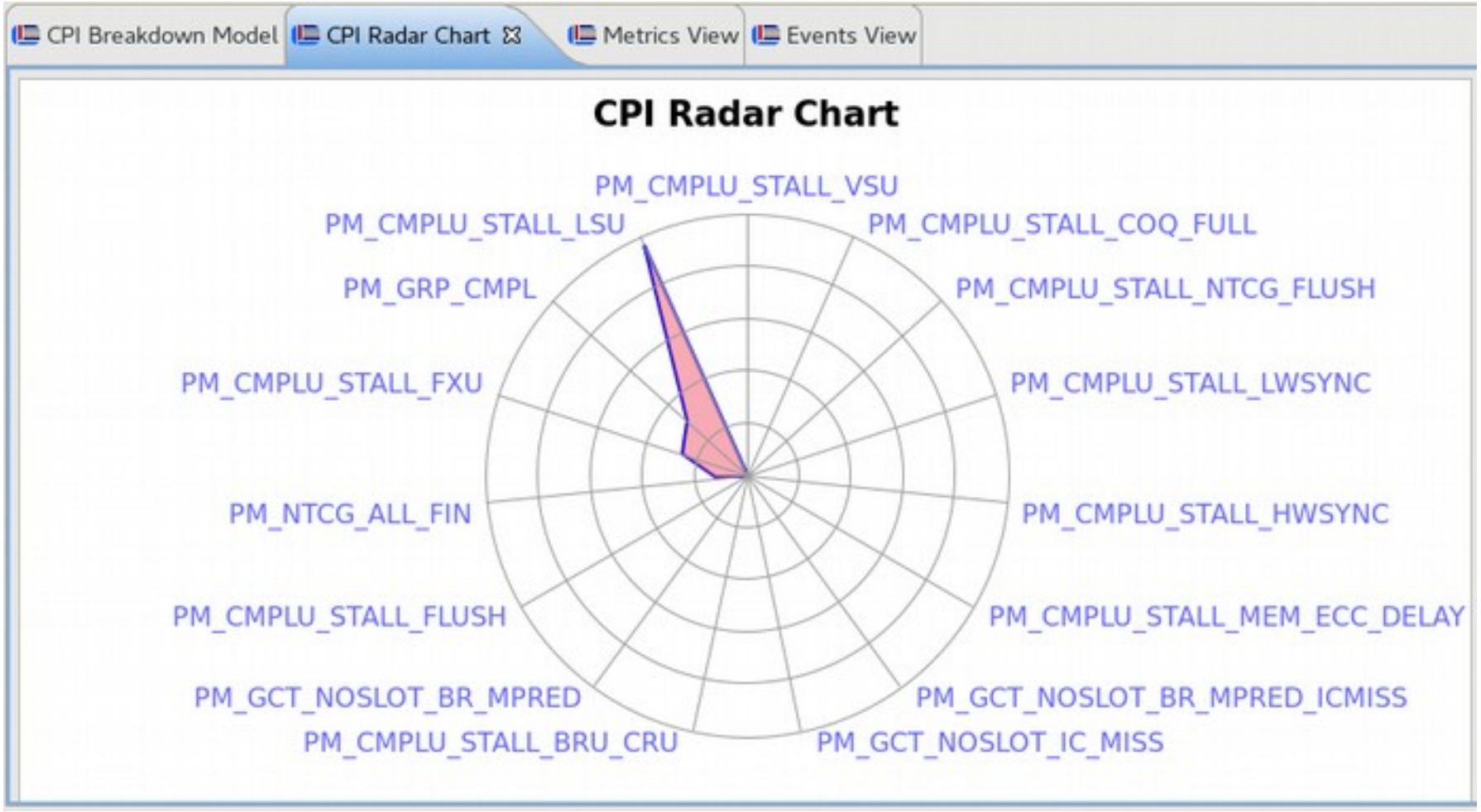
OpenPOWER

SDK: CPI Breakdown & Drill-down

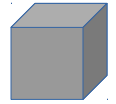


OpenPOWER

SDK: CPI Breakdown & Drill-down



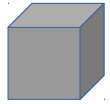
SDK: CPI Breakdown & Drill-down

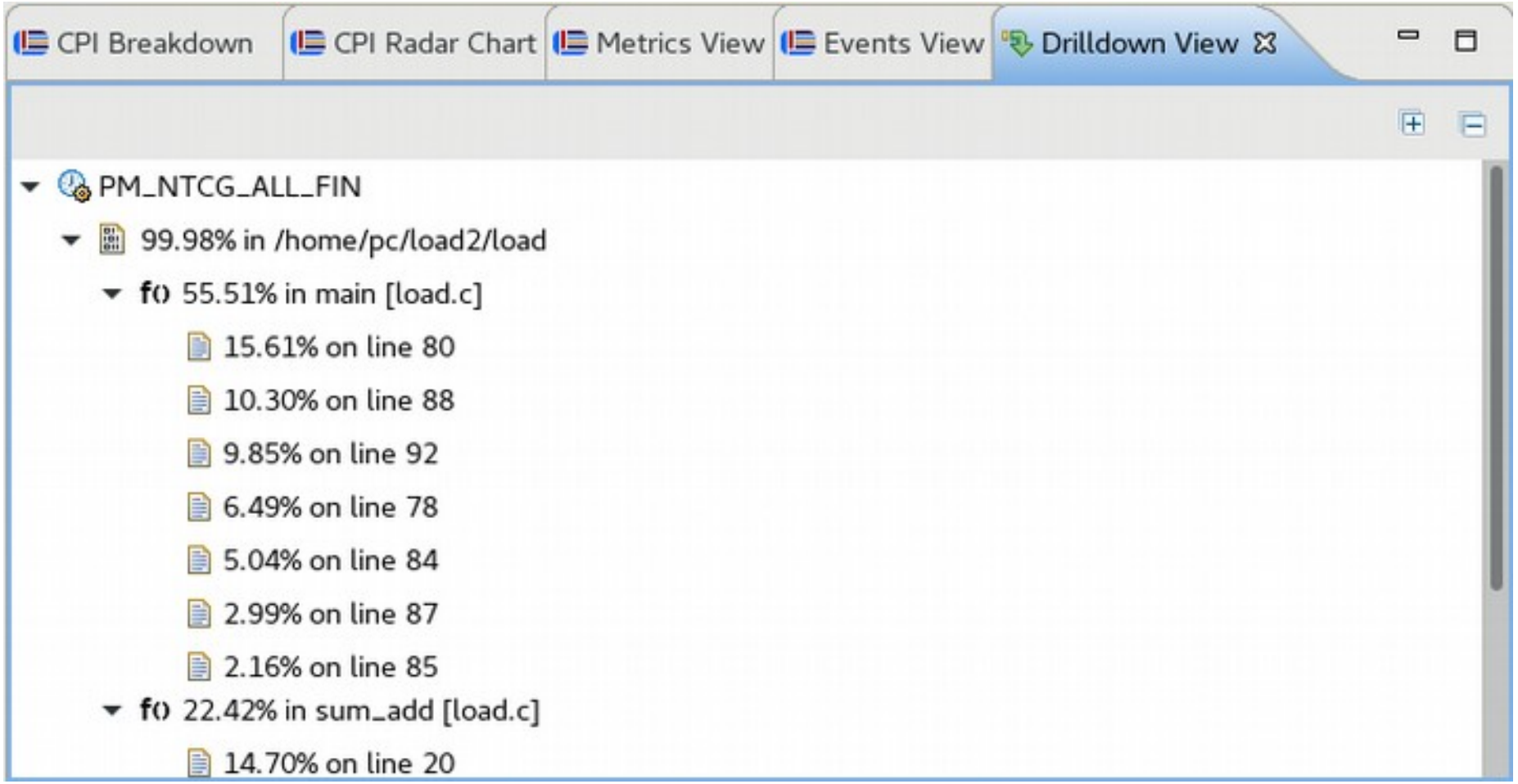


- CPI Drill-down: profile based on specific hardware events
- Simple: double-click on events in breakdown view
  - A new profiling run is automatically launched

OpenPOWER

SDK: CPI Breakdown & Drill-down





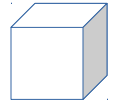
OpenPOWER

Command-line tools from the SDK

- GUIs are powerful
- But, GUIs do not fit every customer, every situation
  - Difficult to automate
  - Expensive to develop and “get right”
  - Customer preference
- Command line tools for SDK features are now available
- All open source
- https://github.com/open-power-sdk

OpenPOWER

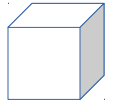
ma (migration advisor)



- Code scanner
- Focus: Linux on x86 to Linux on Power
- Reports on portability issues and concerns
  - list checkers here
- No “Quick Fixes”
- Open source (Python)
- https://github.com/open-power-sdk/migration-advisor

OpenPOWER

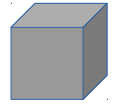
ma (migration advisor)



```
$ ma run ma/
[…]
================
Migration Report
================
Problem type: Non Portable Pthread
Problem description: Reports occurrences of non-portable Pthreads API
   File: ma/pthread.c
      Line: 3
      Problem: pthread_id_np_t tid

      Line: 4
      Problem: pthread_getthreadid_np()

Problem type: Performance degradation
Problem description: This preprocessor can contain code without Power optimization
   File: ma/performance.c
      Line: 3
      Problem: #ifdef _x86_
```

OpenPOWER

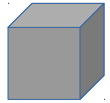
sca (source code advisor)



- Run-time performance analysis and report suggested source code changes for improvement
- Leverages FDPR's journal
- No Quick Fixes
- Open source (Python)
- https://github.com/open-power-sdk/source-code-advisor

OpenPOWER

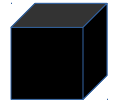
sca (source code advisor)



```
$ sca load
[…]
[Problem: FIX LOAD-HIT-STORE]
[Description: A data store operation followed closely by a load from the same address causes the load to take extra time to complete. ]
[Solution:
        1) Remove unnecessary "volatile" from variables.
        2) Use local variables instead of pointer references.
           Pseudo-example:
           loop { *p = foo(*p); }
           - Change to:
           p = *p
           loop { p = foo(p); }
           *p = p;
[Reference: /home/pc/load2/load.c:20 | Function: sum_add | Instruction Pointer: 10000774]
[Reference: /home/pc/load2/load.c:24 | Function: sum_sub | Instruction Pointer: 100007bc]
-----------------------------------------------------
[Problem: INLINE FUNCTION]
[Description: High overhead for frequent calls to a small function.]
[Solution:
        Enable compiler optimization, if not already enabled.
        If optimization is enabled, tell the compiler to ignore some restrictions and inline the function: add the GNU extension
"__attribute__((always_inline))" to the declaration of the function to be inlined.
        Example: void __attribute__((always_inline)) foo(void);
        Note: Validate the results, as inlining can cause performance regressions in some scenarios.
[Reference: /home/pc/load2/load.c:88 | Function: main | Instruction Pointer: 100009d0]
[Reference: /home/pc/load2/load.c:92 | Function: main | Instruction Pointer: 10000a0c]
```

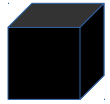
OpenPOWER

cpi (CPI breakdown)



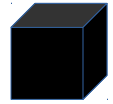
- CPI Breakdown will breakdown CPI measurement into a hierarchy of processor-specific event classes
- 2-step process: record, display
- Open source (Python)
- https://github.com/open-power-sdk/cpi-breakdown

cpi (CPI breakdown)



```
$ cpi record ./load
    Recording CPI Events: 20/20 iterations (elapsed time: 168 seconds)
$ cpi display --file ./load_20180301_123123.cpi
[edited for brevity]
RUN_CPI: 1.398 (100.00 %)
      STALL_CPI: 0.989 (70.78 %)
                  BRU_CRU_STALL_CPI: 0.001 (0.07 %)
                  FXU_STALL_CPI: 0.218 (15.63 %)
                  VSU_STALL_CPI: 0.000 (0.00 %)
                  LSU_STALL_CPI: 0.610 (43.63 %)
                  NTCG_FLUSH_CPI: 0.000 (0.00 %)
                  NO_NTF_STALL_CPI: 0.000 (0.00 %)
                  OTHER_STALL_CPI: 0.160 (11.45 %)
      NTCG_ALL_FIN_CPI: 0.099 (7.09 %)
      THREAD_BLOCK_STALL_CPI: 0.019 (1.35 %)
                  LWSYNC_STALL_CPI: 0.000 (0.00 %)
                  HWSYNC_STALL_CPI: 0.000 (0.00 %)
                  MEM_ECC_DELAY_STALL_CPI: 0.000 (0.01 %)
                  FLUSH_STALL_CPI: 0.019 (1.34 %)
                  COQ_FULL_STALL_CPI: 0.000 (0.00 %)
                  OTHER_BLOCK_STALL_CPI: 0 (0.0 %)
      GCT_EMPTY_CPI: 0.012 (0.84 %)
                  GCT_EMPTY_IC_MISS_CPI: 0.000 (0.03 %)
                  GCT_EMPTY_BR_MPRED_CPI: 0.009 (0.67 %)
                  GCT_EMPTY_BR_MPRED_IC_MISS_CPI: 0.000 (0.00 %)
                  GCT_EMPTY_DISP_HELD_CPI: 0.001 (0.06 %)
                  GCT_EMPTY_OTHER_CPI: 0.001 (0.07 %)
      COMPLETION_CPI: 0.257 (18.40 %)
      OTHER_CPI: 0.022 (1.54 %)
```

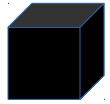
cpi (CPI breakdown)



```
$ cpi display --top-events 5 --top-metrics 5 --file ./load_20180301_123123.cpi

=================
Metrics Hot Spots
=================
    RUN_CPI : 1.398
    STALL_CPI : 0.989
    LSU_STALL_CPI : 0.610
    LSU_STALL_ST_FWD_CPI : 0.406
    COMPLETION_CPI : 0.257

================
Events Hot Spots
================
    PM_RUN_CYC : 37751459119
    PM_RUN_INST_CMPL : 27005146322
    PM_CMPLU_STALL : 26719546039
    PM_CMPLU_STALL_LSU : 16471141823
    PM_CMPLU_STALL_ST_FWD : 10959720878
```

OpenPOWER

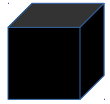
curt



- System-wide utilization report
- Inspired by AIX “curt” command, but otherwise unrelated
- Full-system trace (“perf”) with specific events enabled
- Reports system-wide, per-process, per-task, per-CPU:
  - user time
  - kernel time
  - idle time
  - syscall time
  - HCALL time
  - HCALL time per syscall
  - interrupt time
  - task migrations

OpenPOWER

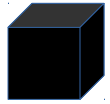
curt



```
$ perf script -s ./curt.py        # [heavily edited for brevity, clarity]
--  PID:
  17096:
     -- [    task] command              cpu         user          sys           hv         busy         idle
        [   17096] curl                   0     0.622258     0.455966     0.533538     0.000000    70.900980
        [   17096] curl                   7     4.957260     1.281960     0.638216     0.000000   184.465554
        [   17096] curl                 ALL     5.579518     1.737926     1.171754     0.000000   255.366534

        [   17096] curl                 cpu      runtime        sleep         wait      blocked       iowait  unaccounted |  util%  moves
        [   17096] curl                   0     2.215550     0.000000     0.000000     0.000000     0.000000     0.084866
        [   17096] curl                   7     7.285182     0.000000     0.000000     0.000000     0.000000     0.000000
        [   17096] curl                 ALL     9.500732     0.000000     0.000000     0.000000     0.000000     0.084866 |   3.0%      1

     -- name                   count      elapsed      pending      average      minimum      maximum
        mmap                      72     0.382168     0.000000     0.005308     0.002336     0.027584
        close                     38     0.028054     0.000000     0.000738     0.000510     0.003484
        mprotect                  37     0.205024     0.000000     0.005541     0.001896     0.008204
        open                      36     0.181678     0.000000     0.005047     0.003246     0.016274
        fstat                     35     0.029962     0.000000     0.000856     0.000596     0.003878
        read                      33     0.109874     0.000000     0.003330     0.000718     0.030738

     -- (hvc)name                 count      elapsed      pending      average      minimum      maximum
        (  8)H_ENTER                249     0.471600     0.000000     0.001894     0.001136     0.006916
        (  4)H_REMOVE               152     0.260904     0.000000     0.001716     0.001120     0.002862
        (292)H_BULK_REMOVE           39     0.102006     0.000000     0.002616     0.001366     0.007068

     -- (irq)  count      elapsed      pending      average      minimum      maximum
        ( 20)      1     0.051650     0.000000     0.051650     0.051650     0.051650

     -- [    task] command              cpu         user          sys           hv         busy         idle
[     ALL]                        ALL     6.164324     2.401406     1.456634     0.000000   512.169142
        [     ALL] curl         runtime        sleep         wait      blocked       iowait  unaccounted |  util%  moves
        [     ALL] curl     10.985284     0.000000     0.000000     0.000000     0.000000   202.917200 |   1.0%      3
```

OpenPOWER

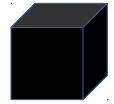
curt



```
ALL:
 -- [    task] cpu         user          sys           hv         busy         idle
    [     ALL] ALL    81.803774   191.490346     1.541632     0.461220  2629.644696

 -- [    task] cpu      runtime        sleep         wait      blocked       iowait  unaccounted |  util%  moves
    [     ALL] ALL   266.336282     0.000000     0.000000     0.000000     0.000000  1620.639964 |   9.0%      5

 -- name                 count      elapsed      pending      average      minimum      maximum
    write               134926   189.079072     0.000000     0.001401     0.001194     0.159456
    mmap                    83     0.417228     0.000000     0.005026     0.001994     0.027584
    close                   51     0.037880     0.000000     0.000742     0.000482     0.003484
    open                    45     0.214196     0.000000     0.004759     0.002254     0.016274
    read                    45     0.175362     0.054250     0.003896     0.000718     0.030738

 -- (hvc)name                  count      elapsed      pending      average      minimum      maximum
    (  8)H_ENTER                329     0.678906     0.000000     0.002063     0.001112     0.032556
    (  4)H_REMOVE               190     0.314528     0.000000     0.001655     0.001094     0.002862
    (292)H_BULK_REMOVE           48     0.128092     0.000000     0.002668     0.001366     0.007068
    (768)H_RANDOM                 7     0.006144     0.000000     0.000877     0.000604     0.001808
    ( 24)H_PROTECT                6     0.007320     0.000000     0.001220     0.000770     0.002390

 -- (irq)  count      elapsed      pending      average      minimum      maximum
    ( 20)      1     0.051650     0.000000     0.051650     0.051650     0.051650

Total Trace Time: 264.750018 ms
```

OpenPOWER

Power Functional Simulator



- Full-system Power hardware simulator
- Access a Power System without access to a Power System!
- Great for basic Linux on Power application development, porting
- Single core, single thread
- POWER8: https://www-304.ibm.com/webapp/set2/sas/f/pwrfs/home.html
- POWER9: https://www-304.ibm.com/webapp/set2/sas/f/pwrfs/pwr9/home.html
- Ease-of-use wrappers at https://github.com/open-power-sdk/power-simulator

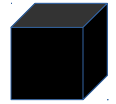
Power Functional Simulator

- [**x86-laptop**]$ **mambo -s power9** # (edited for brevity...)

```
-----------------------------------------
You are starting the IBM POWER9 Functional Simulator
When the boot process is complete, use the following credentials to access it via ssh:

     ssh root@172.19.98.109
     password: mambo

Licensed Materials - Property of IBM.
(C) Copyright IBM Corporation 2001, 2017
All Rights Reserved.
Using initial run script /opt/ibm/systemsim-p9/run/p9/linux/boot-linux-le-skiboot.tcl
Starting mambo with command: /opt/ibm/systemsim-p9/bin/systemsim-p9 -W -f /opt/ibm/systemsim-p9/run/p9/linux/boot-linux-le-skiboot.tcl
Found skiboot skiboot.lid in current directory
Found kernel vmlinux in current directory
Found disk image disk.img in current directory
Booting with skiboot ./skiboot.lid...…
Booting with kernel ./vmlinux...…
root disk ./disk.img
INFO: 0: (0): !!!!!! Simulator now in TURBO mode !!!!!!
OPAL v5.7-107-g8fb78ae starting…
[...]
Linux version 4.13.0-rc4+ (pc@moose1.pok.stglabs.ibm.com) (gcc version 4.8.5 20150623 (Red Hat 4.8.5-11) (GCC)) #2 SMP Fri Aug 18
17:01:57 EDT 2017
[...]
Debian GNU/Linux 9 mambo ppc64le 172.19.98.109
mambo login:
```

OpenPOWER

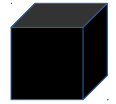
Performance Simulator (sim_ppc)



- Cycle-accurate POWER processor simulator
- Transforms instruction traces into processor cycle reports
- Cycle reports can be viewed using included viewers (ScrollPipeViewer and jviewer)
- Use Valgrind itrace to record instruction trace
  - Not part of standard Valgrind
  - Included in Advance Toolchain's Valgrind

OpenPOWER

Performance Simulator (sim_ppc)



Basic usage scenario:

1. Record instruction trace (.vgi file)

```
$ valgrind --tool=itrace --binary-outfile=tracefile.vgi --num-K-
insns-to-collect=100 --demangle=no /bin/ls
```

2. Convert .vgi to .qt

```
$ vgi2qt -f tracefile.vgi -o tracefile.qt
```

3. Run sim_ppc cycle-accurate timer (.pipe file)

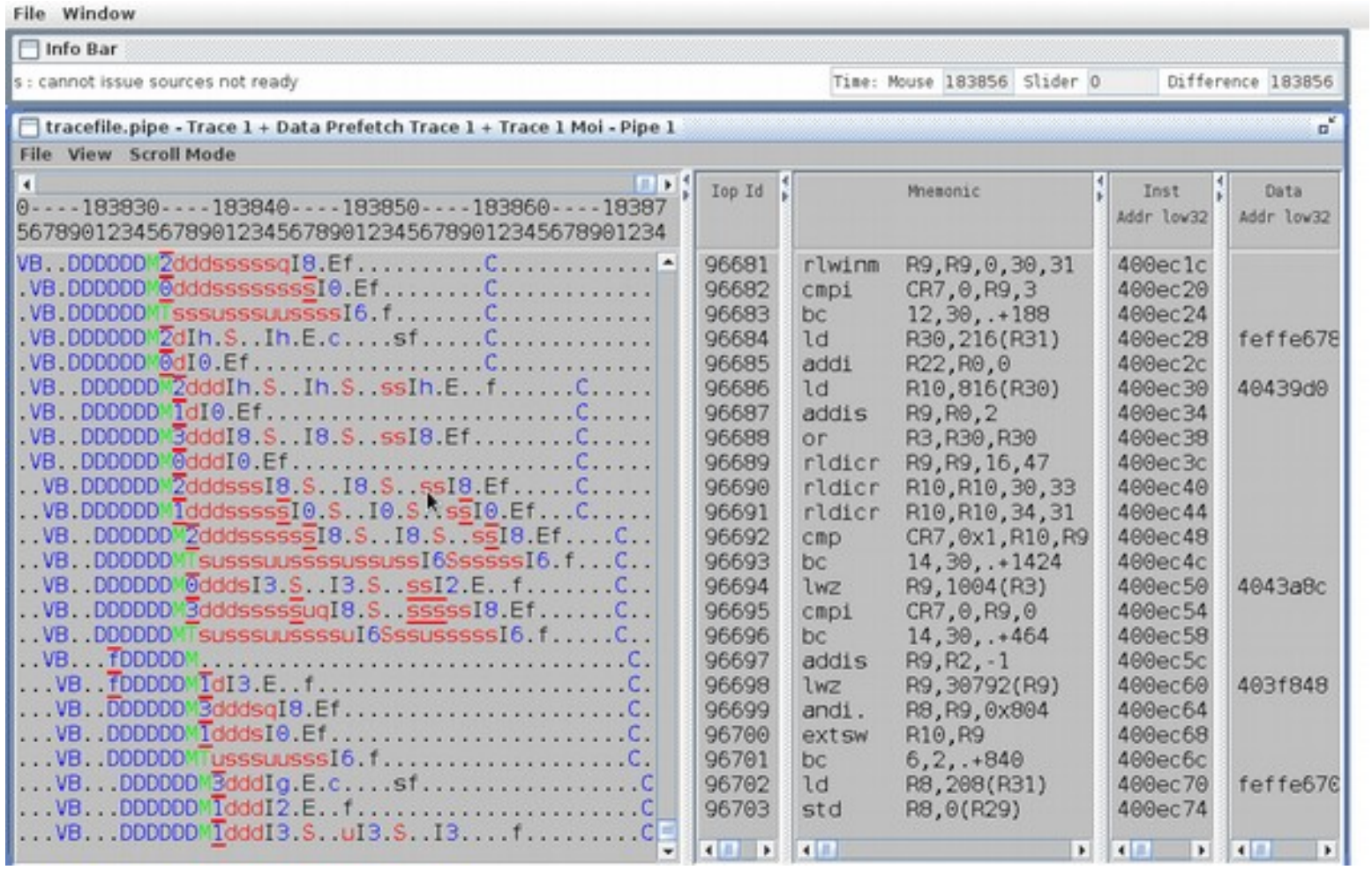
```
$ /opt/ibm/sim_ppc/sim_p8/bin/run_timer tracefile.qt 100000 10000
1 tracefile -scroll_pipe 1 -scroll_begin 1 -scroll_end 100000
```

4. Run viewer

```
$ /opt/ibm/sim_ppc/bin/scrollpv tracefile.pipe
```

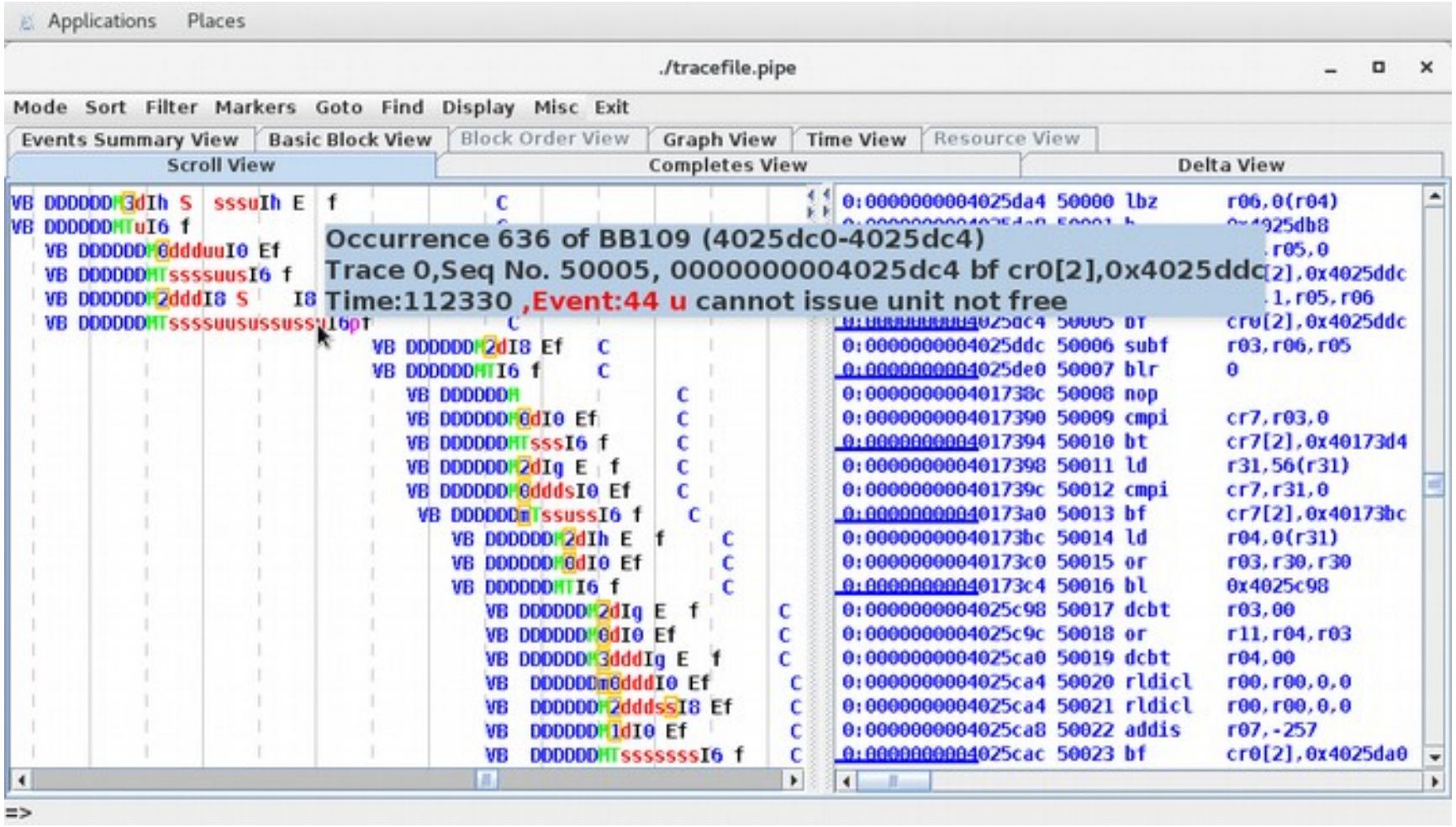
OpenPOWER

Performance Simulator (scrollpv)



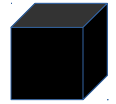
OpenPOWER

Performance Simulator (jviewer)



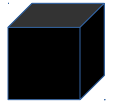
OpenPOWER

pipestat



- Automated analysis of processor instruction cycle reports (from Performance Simulator)
  - Most executed loops
  - Most executed misaligned short loops
  - Most executed blocks with long latency instructions
  - Most executed blocks with redundant loads
  - Most executed incorrectly hinted branches
  - Most executed mispredicted branches
  - Most frequently mispredicted branches
  - Most executed code paths that have a store followed soon after by a load of the same address
  - Most load-hit-store related events on a particular instruction address
  - Most executed instructions where the result is used a small number of instructions later but takes a large number of cycles before the dependent instruction starts

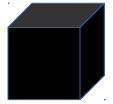
pipestat



```
$ pipestat tracefile.qt        # [heavily edited for brevity, clarity]
HOT execution count blocks:
  0x000004025c98-0x000004025cac N:678 6 inst trace inst 297
  0x000004025dc8-0x000004025dd8 N:784 5 inst trace inst 923
  0x00000400df48-0x00000400df7c N:248 14 inst trace inst 454

HOT misaligned short loops:
  0x00000400ea58-0x00000400ea74 N:170 8 inst short misalign32

Loop size summary data:
Instructions    loops      total iter   min iter   max iter   avg iter      total inst  % of trace
         6          1             402        402        402     402.00            2412        2.93
         8          1             170        170        170     170.00            1360        1.65

HOT Loop constructs (5 total)
                arch                       inst/                    BackEdge:          Edge:
Hdr blk IA       inst static  dynamic iter  iter nodes taken untaken bc_tk bc_nt fallth bc_tk bc_nt   br fallth   BL  BLR
000000000400eb40  112      6     2412  402   6.0     1  0.00    0.00   402     0      0     0     0    0      0    0    0
0000000004025db8  306     12     9830 1234   8.0     5  0.62    2.16     0     0    648   770  2666    0      0    0    0
000000000400ea58  419      8     1360  170   8.0     1  0.00    0.00   170     0      0     0     0    0      0    0    0

HOT long latency instruction blocks:
  0x00000400ea7c-0x00000400ea90 N:48 6 inst badness 240
  0x00000400e7d0-0x00000400e7ec N:55 8 inst badness 206
  0x0000040173f0-0x000004017408 N:235 7 inst badness 168
```

OpenPOWER

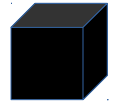
pipestat



```
HOT redundant loads: intra+stack: 261 (0.32%) inter+stack: 948 (1.15%) intra: 0 (0.00%) inter: 0 (0.00%)
  0x00000400e0d4-0x00000400e0f4 N:200 redundant loads 600
  0x0000040173f0-0x000004017408 N:235 redundant loads 235
  0x00000400e2ec-0x00000400e2f8 N:235 redundant loads 230

HOT bad branch hints:
0x000004025dc4 hint likely not taken but was taken 36.47% (450/1234)
0x000004025db0 hint likely not taken but was taken 15.84% (122/770)
0x00000400e124 hint likely taken but was not taken 100.00% (87/87)

HOT branches with high linear branch entropy and executed frequently
  0x00000400e824 N:110      LBE 0.00000 0.6909/0.0000
  0x00000400e338 N:48       LBE 0.00000 0.5417/0.0000
  0x00000400e90c N:200      LBE 0.14444 0.3600/0.1444

HOT load hit store separated by less than 100 instructions:
                                      of all exec:     Pathlength        Std. redund other   AGEN
Store IA       Load IA    Count   %    count min count   Avg max   Dev. loads store registers      Store Values
00000400df64 00000400e0dc   200 100.0%   200 13     87 34.17  68 358.1    200 1/1/1 st: G1 ld: G1
00000400ebac 00000400e1b0    48 100.0%    48 37     48 37.00  37   0.0      0 22/22/22 st: G1 ld: G1 BASE REG CHANGED: 48
00000400eba8 00000400e1ac    48 100.0%    48 37     48 37.00  37   0.0      0 23/23/23 st: G1 ld: G1 BASE REG CHANGED: 48
Total LHS events: 3282 15.1% of loads 40.0% of stores
```

OpenPOWER

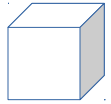
pipestat



```
Inst annotated disassembly
00000400df48 > mflr   r0            N:248 HOT-blk Use:7 HOT-LongLat LBAD 5 from:0x400e858,0x400e928
00000400df4c M std    r30,-16(r1)   N:248 HOT-blk HOT-LHS DA ref to intra 0x0400e0e8 22i 35.08%
00000400df50 M std    r31,-8(r1)    N:248 HOT-blk HOT-LHS DA ref to intra 0x0400e0ec 22i 35.08%
00000400df54   mr     r30,r3,r3     N:248 HOT-blk Use:11
00000400df58 M std    r28,-32(r1)   N:248 HOT-blk HOT-LHS DA ref to intra 0x0400e0e0 17i 35.08%
00000400df5c M std    r29,-24(r1)   N:248 HOT-blk HOT-LHS DA ref to intra 0x0400e0e4 17i 35.08%
00000400df60   mr     r31,r11,r11   N:248 HOT-blk Use:9
00000400df64   std    r0,16(r1)     N:248 HOT-blk HOT-LHS DA ref to intra 0x0400e0dc 13i 35.08%
00000400df68   stdu   r1,-64(r1)    N:248 HOT-blk Use:63
00000400df6c   ld     r10,8(r3)     N:248 HOT-blk Use:2
00000400df70   lbz    r9,4(r3)      N:248 HOT-blk Use:2
00000400df74   cmpdi  cr7,r10,0     N:248 HOT-blk Use:2
00000400df78   clrldi r8,r9,60      N:248 HOT-blk Use:7.2
00000400df7c C beq-   cr7,0x400e120 N:248 HOT-blk Hint:unlikely HOT BAD HINT HOT-LBE to:0x400e120 br prob:35.08% (avg seq tk 1.2 ntk
2.3 LBE 0.079 0.169/0.079)

Opcode Mix
Opcode Name         Mask   Match            Count
ld              fc000003 e8000000          11654
std             fc000003 f8000000           6560
cmpi            fc200000 2c200000           5255
bc              ffc00003 40c00000           4000

Total loads: 21672 Total stores: 8203
GPR: loads: 21672 stores: 8203
```

OpenPOWER

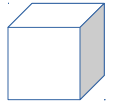
pveclib



- Header files that contain well crafted implementations of useful vector and large number operations the Power ISA Vector Facilities
- Provide higher order functions not provided directly by the PowerISA
- Open source
- https://github.com/open-power-sdk/pveclib

OpenPOWER

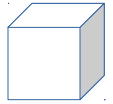
pveclib



- udiv_qrnnd
- fxu_bcdadd, fxu_bcd_sub
- vec_BCD2DFP, vec_DFP2BCD
- vec_bcdadd, vec_bcdsub, vec_bcdmul, vec_bcddiv
- vec_shift_leftdo
- vec_isalpha, vec_isalnum, vec_isdigit
- vec_toupper, vec_tolower
- vec_absdub
- vec_revq, vec_revd, vec_revw, vec_revh
- vec_clzq, vec_popcntq
- vec_sldq, vec_srqi, vec_srq, vec_slqi, vec_slq
- vec_pasted
- vec_mulouw, vec_muleuw, vec_mulosw, vec_mulesw
- vec_adduqm, …
- ...

OpenPOWER

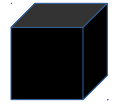
SPHDE



- Shared Persistent Heap Data Environment
  - Shared Address Space (SAS)
  - Shared Persistent Heap
  - Cross-platform
  - Lockless
- Shared memory using a shared heap allows passing valid pointers between processes, easing implementation of multiprocessing
- Lockless multiprocess logger
- Lockless producer-consumer queue (fast IPC)
- Fast timestamps
- Open source
- https://github.com/sphde/sphde

OpenPOWER

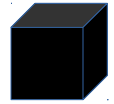
LPCPU



- Linux Performance Customer Profiler Utility
- Collects a large, customizable set of system information and performance data for offline analysis
- Two-step process:
  - Collect data (produces a compressed tar file)
  - Postprocess data (collates data, produces interactive HTML)
- Open source
- http://ibm.co/download-lpcpu

OpenPOWER

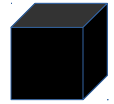
LPCPU



- `iostat`
- `mpstat`
- `vmstat`
- `perf or OProfile`
- `meminfo`
- `top`
- `sar`
- `/proc/interrupts`
- `tcpdump`
- kernel trace
- hardware performance counters
- `netstat`
- `...`

OpenPOWER

LPCPU

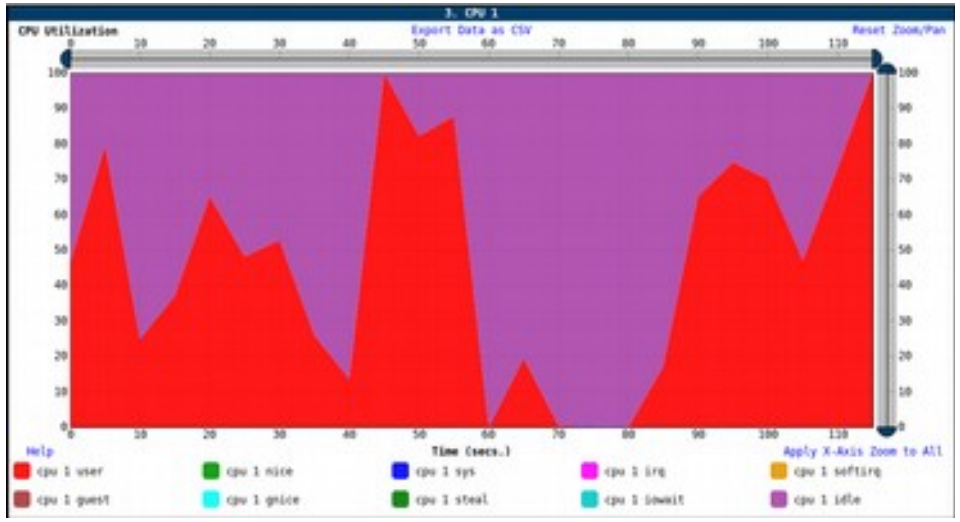


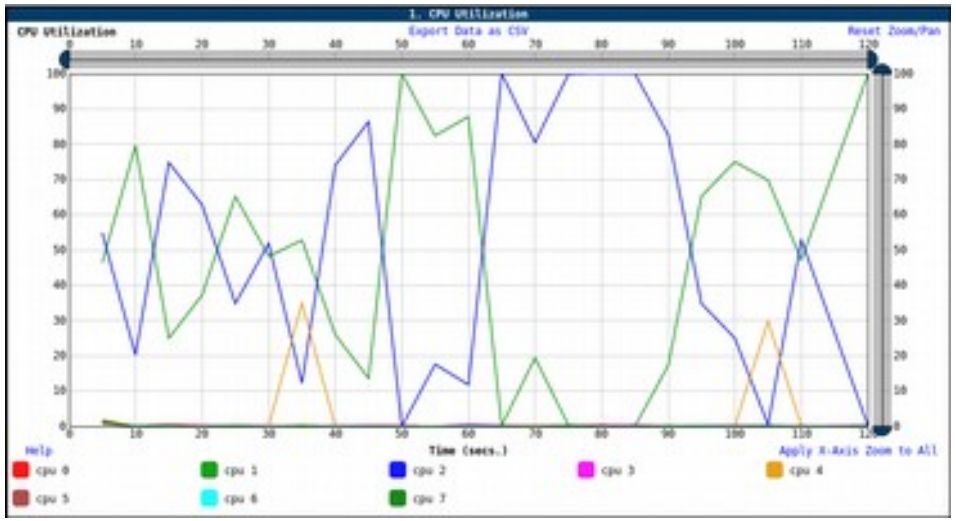
```
# tar -jxf lpcpu.tar.bz2
# cd lpcpu
# ./lpcpu.sh duration=15 extra_profilers="perf tcpdump"
[…]
Packaging data...data collected is in
/tmp/lpcpu_data.hostname.default.2018-02-26_1000.tar.bz2
[optionally, copy elsewhere…]
# tar -xf ./lpcpu_data.hostname.default.2018-02-
26_1000.tar.bz2
# cd lpcpu_data.hostname.default.2018-02-26_1000
# ./postprocess.sh
```

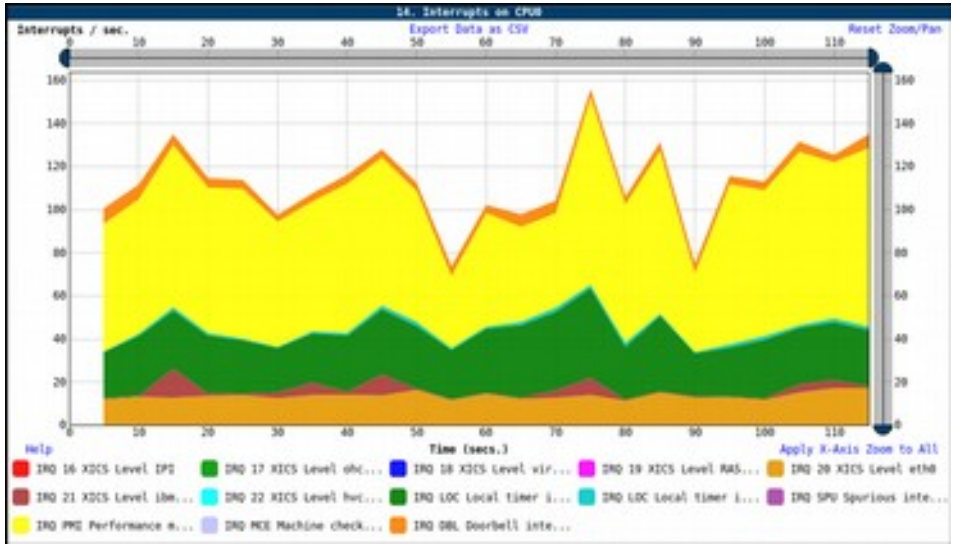
- Then point a browser at the summary.html file

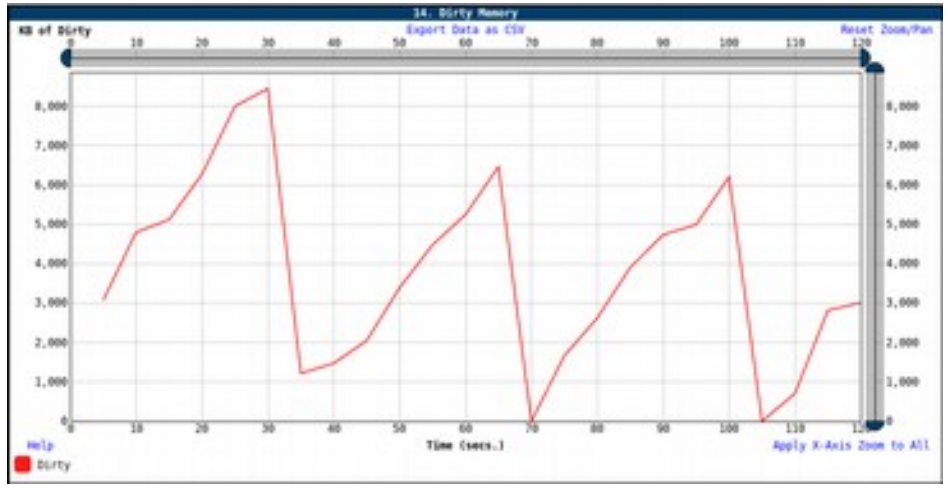
OpenPOWER

LPCPU









OpenPOWER

Support

- Formal Support (Advance Toolchain, XL compilers)
- Linux on Power Developer Portal
  - https://developer.ibm.com/linuxonpower
  - (replaces developerWorks Linux on Power Community)
  - dW Answers (https://developer.ibm.com/answers/smartspace/linuxonpower/index.htm)
- StackOverflow, ServerFault, StackExchange, SuperUser, ...
- IRC channels (FreeNode #ubuntu-powerpc, #fedora-ppc, ...)

OpenPOWER

Thank You!