SREcon20 Americas

Capacity Planning and Performance Enhancement with Page Reference Sampling

Danny Chen
Trading Solutions SRE Team, Bloomberg, LP

TechAtBloomberg.com

© 2020 Bloomberg Finance L.P. All rights reserved.

Bloomberg
Engineering

Outline

- Page reference sampling - some personal history
- The Solaris page reference sampling interface (an open secret?)
- Brief review of paging architectures
- Utilities I’ve written and examples
- Summary - why we need this and what else do we need

© 2020 Bloomberg Finance L.P. All rights reserved.

USENIX Association

Proceedings of the Summer 1988 USENIX Conference

June 20-24, 1988
San Francisco, California USA

RUNTIME ENVIRONMENTS

Thursday (4:00-5:30) **Chair: Jay Lepreau**

A Memory Allocation Profiler for C and Lisp Programs 223
Benjamin Zorn & Paul Hilfinger, University of California, Berkeley

A RISC Approach to Runtime Exceptions 239
Mark Himelstein, MIPS Computer Systems, Inc.; Steven Correll, Key Computer Laboratories; Kevin Enderby, NeXT Inc.

CASPER the Friendly Daemon 251
Ronald E. Barkley, AT&T Information Systems; Danny Chen, AT&T Bell Laboratories

TechAtBloomberg.com

© 2020 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

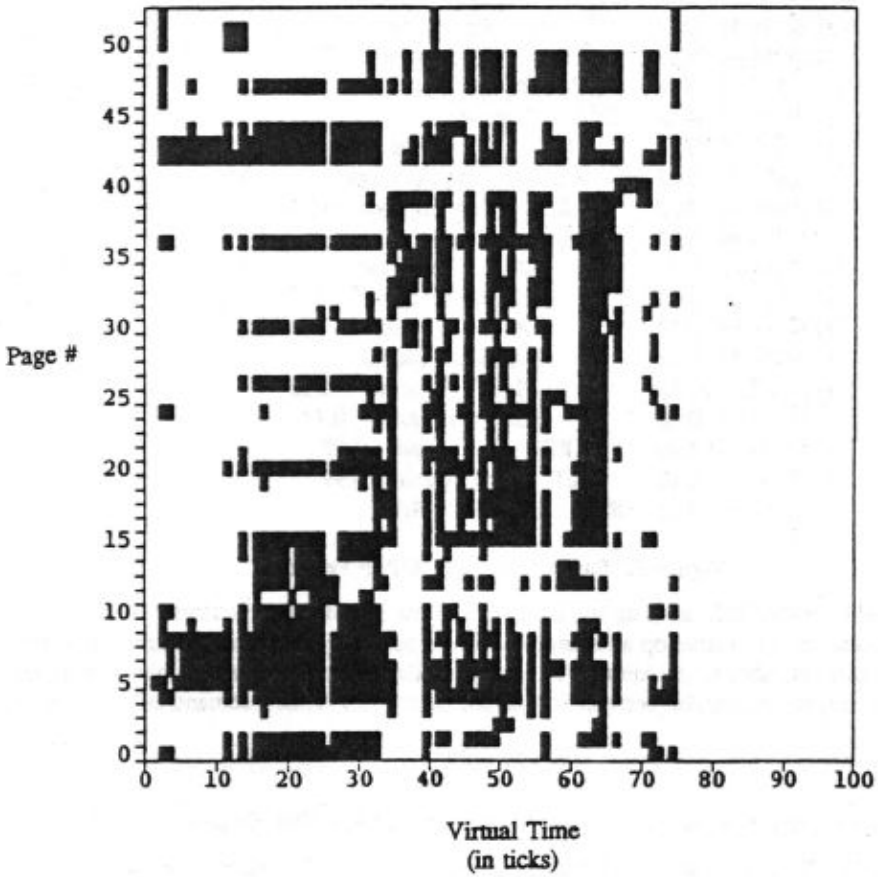
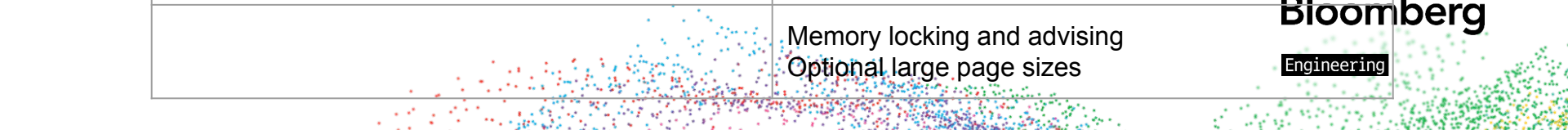


Figure 4. Memory Reference Chart

© 2020 Bloomberg Finance L.P. All rights reserved.

Software & Systems: Then vs Now

Then	Now
32 bit • Small address space programs	32 and 64 bit • Large, sparse address space programs
Uni-processor	Multi-processor, multi-threads
Speed in 1-10 MIPS	Speed in 10^5-10^6 MIPS
Small processor caches	Large, multi-level processor caches
Main memory in Megabytes	Main memory in GB and TB
Static linking	Dynamic linking and loading
Mainly “utilities” (short lived) Compiled	Lots of “servers” (long lived) Interpreted languages with GC
	Memory locking and advising Optional large page sizes



Bloomberg

Engineering

Available on Solaris via /proc/pid/pagedata

```
pagedata

Opening the page data file enables tracking of address space references
and modifications on a per page basis.
```

TechAtBloomberg.com

© 2020 Bloomberg Finance L.P. All rights reserved.



Bloomberg

Engineering

Available on Solaris via /proc/pid/pagedata

pagedata

Opening the page data file enables tracking of address space references and modifications on a per-page basis.

A read(2) of the page data file descriptor returns structured page data and atomically clears the page data maintained for the file by the system. That is to say, each read returns data collected since the last read; the first read returns data collected since the file was opened. When the call completes, the read buffer contains the following structure as its header and thereafter contains a number of section header structures and associated b… must be accessed by walking linearly through the buffer.

```
typedef struct prpageheader {
    timestruc_t pr_tstamp; /* real time stamp, time of read() */
    ulong_t pr_nmap;       /* number of address space mappings */
    ulong_t pr_npage;      /* total number of pages */
} prpageheader_t;
```

The header is followed by `pr_nmap` `prasmap` structures and associated data arrays. The `prasmap` structure contains the following elements:

```
typedef struct prasmap {
    uintptr_t pr_vaddr;        /* virtual address of mapping */
    ulong_t pr_npage;          /* number of pages in mapping */
    char pr_mapname[PRMAPSZ];  /* name in /proc/pid/object */
    offset_t pr_offset;        /* offset into mapped object, if any */
    int pr_mflags;             /* protection and attribute flags */
    int pr_pagesize;           /* pagesize for this mapping in bytes */
    int pr_shmid;              /* SysV shared memory identifier */
} prasmap_t;
```

Each section header is followed by `pr_npage` bytes, one byte for each page in the mapping, plus 0-7 null bytes at the end so that the nex… structure begins on an eight-byte aligned boundary. Each data byte may contain these flags:

`PG_REFERENCED`

page has been referenced.

`PG_MODIFIED`

page has been modified.

If the read buffer is not large enough to contain all of the page data, the read fails with `E2BIG` and the page data is not cleared. The required size of the read buffer can be determined through fstat(2). Application of lseek(2) to the page data file descriptor is ineffective; every read starts from the beginning of the file. Closing the page data file descriptor terminates the system overhead associated with collecting the data.

More than one page data file descriptor for the same process can be opened, up to a system-imposed limit per traced process. A read of one does not affect the data being collected by the system for the others. An open of the page data file will fail with `ENOMEM` if the system-imposed limit would be exceeded.


```
typedef struct prpageheader {
    timestruc_t pr_tstamp; /* real time stamp, time of read() */
    ulong_t pr_nmap;       /* number of address space mappings */
    ulong_t pr_npage;      /* total number of pages */
} prpageheader_t;
```

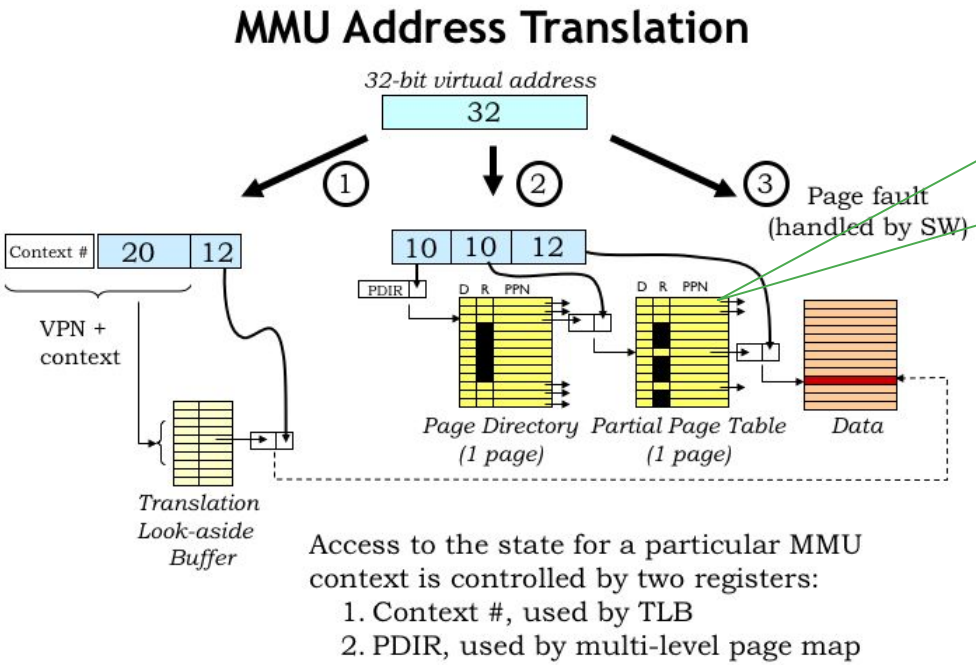
The header is followed by `pr_nmap` `prasmap` structures and associated data arrays.

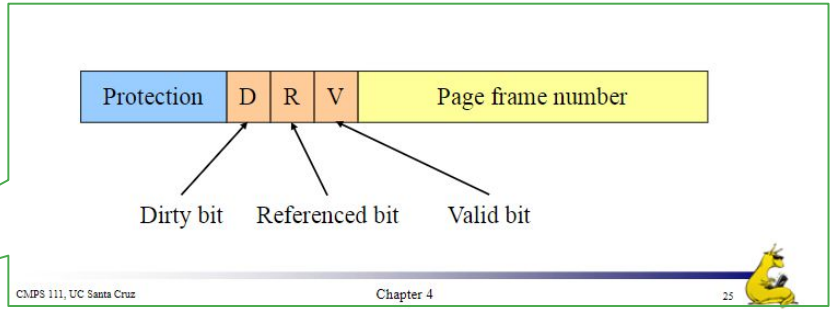
```
typedef struct prasmap {
    uintptr_t pr_vaddr;        /* virtual address of mapping */
    ulong_t pr_npage;          /* number of pages in mapping */
    char pr_mapname[PRMAPSZ];  /* name in /proc/pid/object */
    offset_t pr_offset;        /* offset into mapped object, if any */
    int pr_mflags;             /* protection and attribute flags */
    int pr_pagesize;           /* pagesize for this mapping in bytes */
    int pr_shmid;              /* SysV shared memory identifier */
} prasmap_t;
```


Bloomberg

Engineering

How is this done?





MMU translation graphic from https://computationstructures.org/lectures/virtual/virtual.html#2
Page table entry graphic from: https://users.soe.ucsc.edu/~scott/courses/Fall05/111/Slides/chapter4.pdf

Counting page references with wssts

Usage: wssts -c <command string> [-i data_interval] [-I metadata_interval]

Working Set Size Time Series (wssts)

- -c command string: monitor processes that match this command string
- -I metadata_interval: the interval used to look for processes that match the command string
- -i data_interval: interval to do page reference polling



Bloomberg

Engineering

Example: looking at crwrkr with wssts

```
ps -opid,rss,rssprivate,comm,args | grep
```

```
[dchen294@nylxdev3 talk (master)]$ cat crwrkr.rss
39587  1861544   710136 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 14 regular 1
40634  1427424   410736 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 3 regular 1
39623  1516624   503840 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 15 regular 1
 3260  1761584   623080 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 30 regular 1
39267  1616752   517160 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 25 regular 1
40949  1412592   404464 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 6 regular 1
39378  1505904   497840 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 21 regular 1
57005  1786048   634944 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 8 regular 1
39390  1768040   617768 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 20 regular 1
39558  1581760   513632 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 18 regular 1
41197  1542976   471048 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 7 regular 1
39674  1588776   512064 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 9 regular 1
40618  1404232   400136 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 1 regular 1
40181  1672640   549480 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 4 regular 1
39637  1617768   507696 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 12 regular 1
60486  1710192   571520 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 23 regular 1
16474  1684672   542376 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 24 regular 1
39324  1580048   508144 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 26 regular 1
39543  1608296   509360 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 17 regular 1
39753  1758440   625496 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 2 regular 1
39482  1595624   506288 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 16 regular 1
39337  1898984   751656 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 28 regular 1
39659  1613192   519312 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 11 regular 1
39707  1832872   685912 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 10 regular 1
39577  1806152   655072 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 19 regular 1
39246  1609976   515640 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 27 regular 1
39248  1646464   528416 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 29 regular 1
42349  1770784   628536 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 5 regular 1
39607  1539608   506240 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 13 regular 1
39292  1637928   538424 /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 22 regular 1
[dchen294@nylxdev3 talk (master)]$
```

```
[dchen294@nylxdev3 talk (master)]$ cat crwrkr.wss
20:47:17 wssts:[60486] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 23 regular 1: wss = 50232, rocnt = 21673, wrcnt = 28559
20:47:18 wssts:[16474] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 24 regular 1: wss = 68591, rocnt = 22183, wrcnt = 46408
20:47:19 wssts:[3260] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 30 regular 1: wss = 79362, rocnt = 23626, wrcnt = 55736
20:47:19 wssts:[41197] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 7 regular 1: wss = 544, rocnt = 28, wrcnt = 516
20:47:19 wssts:[40181] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 4 regular 1: wss = 564, rocnt = 48, wrcnt = 516
20:47:19 wssts:[39246] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 27 regular 1: wss = 564, rocnt = 48, wrcnt = 516
20:47:20 wssts:[39248] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 29 regular 1: wss = 564, rocnt = 48, wrcnt = 516
20:47:20 wssts:[39267] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 25 regular 1: wss = 564, rocnt = 48, wrcnt = 516
20:47:21 wssts:[42349] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 5 regular 1: wss = 60528, rocnt = 21600, wrcnt = 38928
20:47:21 wssts:[39292] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 22 regular 1: wss = 564, rocnt = 48, wrcnt = 516
20:47:22 wssts:[39324] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 26 regular 1: wss = 564, rocnt = 48, wrcnt = 516
20:47:22 wssts:[39337] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 28 regular 1: wss = 94976, rocnt = 24244, wrcnt = 70732
20:47:23 wssts:[39378] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 21 regular 1: wss = 564, rocnt = 48, wrcnt = 516
20:47:24 wssts:[39390] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 20 regular 1: wss = 564, rocnt = 48, wrcnt = 516
20:47:25 wssts:[39482] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 16 regular 1: wss = 564, rocnt = 48, wrcnt = 516
20:47:25 wssts:[39543] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 17 regular 1: wss = 564, rocnt = 48, wrcnt = 516
20:47:26 wssts:[39558] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 18 regular 1: wss = 564, rocnt = 48, wrcnt = 516
20:47:26 wssts:[39577] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 19 regular 1: wss = 1068, rocnt = 48, wrcnt = 1020
20:47:26 wssts:[39587] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 14 regular 1: wss = 92614, rocnt = 24147, wrcnt = 68467
20:47:26 wssts:[40618] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 1 regular 1: wss = 544, rocnt = 28, wrcnt = 516
20:47:27 wssts:[57005] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 8 regular 1: wss = 60516, rocnt = 21701, wrcnt = 38815
20:47:27 wssts:[39607] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 13 regular 1: wss = 564, rocnt = 48, wrcnt = 516
20:47:27 wssts:[40634] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 3 regular 1: wss = 42, rocnt = 37, wrcnt = 5
20:47:28 wssts:[39623] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 15 regular 1: wss = 564, rocnt = 48, wrcnt = 516
20:47:28 wssts:[39637] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 12 regular 1: wss = 564, rocnt = 48, wrcnt = 516
20:47:28 wssts:[39659] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 11 regular 1: wss = 564, rocnt = 48, wrcnt = 516
20:47:29 wssts:[39674] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 9 regular 1: wss = 564, rocnt = 48, wrcnt = 516
20:47:29 wssts:[39707] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 10 regular 1: wss = 51485, rocnt = 20503, wrcnt = 30982
20:47:29 wssts:[39753] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 2 regular 1: wss = 564, rocnt = 48, wrcnt = 516
20:47:30 wssts:[40949] /bb/bin/bigsuite/current/bin/crwrkr.tsk crwrkr 6 regular 1: wss = 35, rocnt = 30, wrcnt = 5
[dchen294@nylxdev3 talk (master)]$
```



Bloomberg

Engineering

Example: looking at crwrkr with wssts (cont'd)

pid	rss	rssprivate	wss (K)	rocnt (K)	wrcnt (K)
3260	1761584	623080	634896	189008	445888
16474	1684672	542376	548728	177464	371264
39246	1609976	515640	4512	384	4128
39248	1646464	528416	4512	384	4128
39267	1616752	517160	4512	384	4128
39292	1637928	538424	4512	384	4128
39324	1580048	508144	4512	384	4128
39337	1898984	751656	759808	193952	565856
39378	1505904	497840	4512	384	4128
39390	1768040	617768	4512	384	4128
39482	1595624	506288	4512	384	4128
39543	1608296	509360	4512	384	4128
39558	1581760	513632	4512	384	4128
39577	1806152	655072	8544	384	8160
39587	1861544	710136	740912	193176	547736
39607	1539608	506240	4512	384	4128
39623	1516624	503840	4512	384	4128
39637	1617768	507696	4512	384	4128
39659	1613192	519312	4512	384	4128
39674	1588776	512064	4512	384	4128
39707	1832872	685912	411880	164024	247856
39753	1758440	625496	4512	384	4128
40181	1672640	549480	4512	384	4128
40618	1404232	400136	4352	224	4128
40634	1427424	410736	336	296	40
40949	1412592	404464	280	240	40
41197	1542976	471048	4352	224	4128
42349	1770784	628536	484224	172800	311432
57005	1786048	634944	484128	173608	310520
60486	1710192	571520	401856	173384	228472

- RSS doesn't factor in (or factor out) shared pages
- In the absence of page stealing, RSS is monotonically increasing
- RSSprivate factors out shared pages
- Dirty pages have to be written before they are stolen

In this particular case, it looks like there MAY be a large memory startup cost. E.g. initialization code and data.

* all memory values normalized to kilobytes

Capturing Page References (prsample)

Usage: prsample [-o outputfile] -p pid | command args

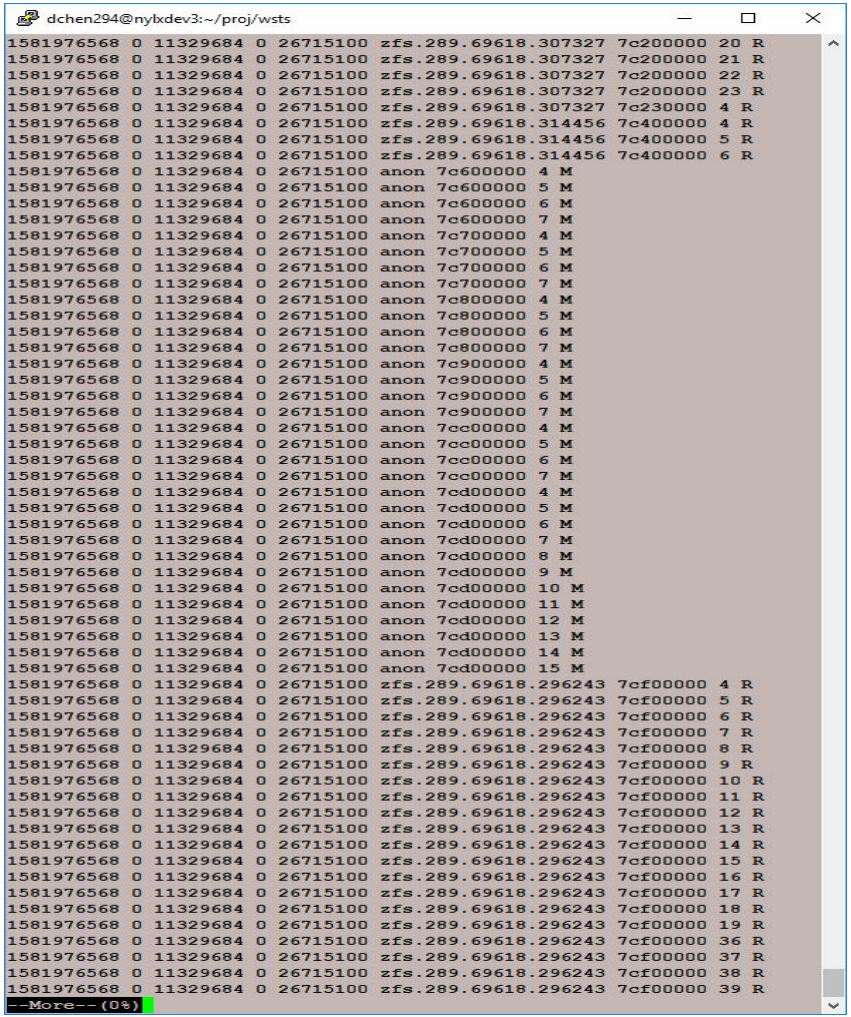


Bloomberg

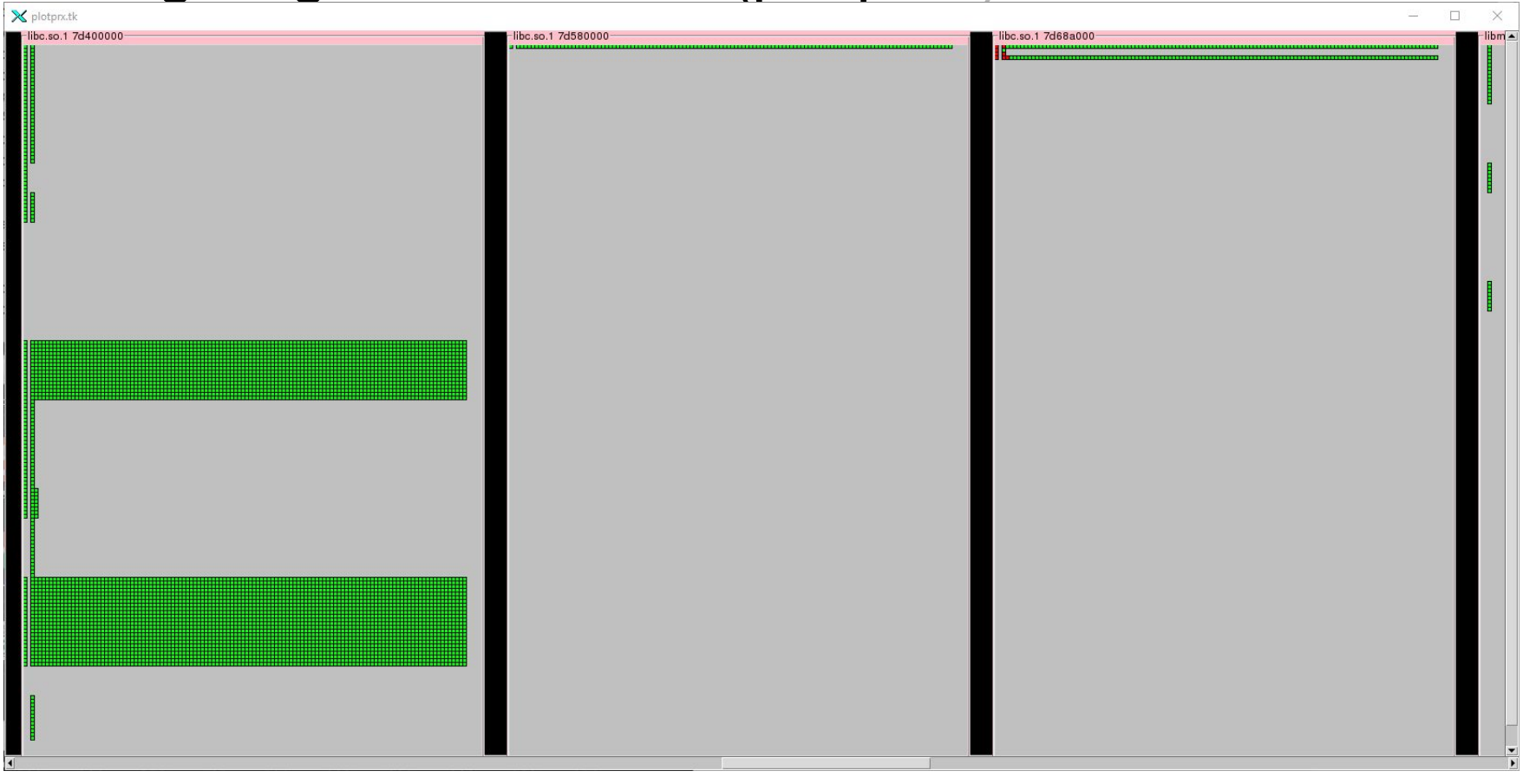
Engineering

Example: prsample

- Wall time (seconds)
- User time (second, nanoseconds)
- System time (seconds, nanoseconds)
- Mapping name
- Base virtual address
- Page offset
- R (read) or M (modified)



Plotting Page References (plotpr.tk)



Counting Page References Over Time

- Knowing which pages are referenced the most (from the sampling perspective) can provide insights into which pages should be kept in memory
  - Leverage OS memory locking facilities
  - External utilities can sometimes help with locking
  - On systems that support pre-paging, knowing which pages to pre-load can help with start up performance
- Caveat: program behavior is often workload dependent



Bloomberg

Engineering

Predicting Page Fragmentation

```
[dchen294@nylxdev3 wsts (master)]$ ./pagefrag.py crwrkr.refcnts 8
'/bb/sys/package/c/crwrkr/0.000000.1053959-20200301T025606-1053959/bin/crwrkr.tsk',
'38234000': 6/1
'/bb/sys/package/c/crwrkr/0.000000.1053959-20200301T025606-1053959/bin/crwrkr.tsk',
'38240000': 4/4763
'/lib/ld.so.1', 'fb9a0000': 0/31
'/lib/libc.so.1', 'fa2c0000': 0/183
'/lib/libc.so.1', 'fa448000': 2/5
'anon', '3a7a2000': 3/4
'anon', '3a800000': 2/95741
'anon', 'a2400000': 0/511
'anon', 'a2800000': 4/155
'anon', 'a2c00000': 0/511
'anon', 'c0c00000': 0/15
'anon', 'c7c00000': 0/2559
'anon', 'cbc00000': 4/515
'anon', 'cc800000': 0/3583
'anon', 'd3000000': 0/1023
```



Bloomberg

Engineering

What about Linux?

- Can estimate wss (see http://www.brendangregg.com/wss.html)
- /proc/pid/smaps - # of pages referenced in a segment
- /proc/pid/clear_refs - clears the reference bits
  - A separate operation from smaps
  - Not coordinated with the page replacement system



Bloomberg

Engineering

In Closing...

- A different view into process behavior
  - Roughly what the page replacement system sees
  - Memory capacity planning
- Selecting page sizes
  - Pro: Larger page sizes -> smaller page tables -> faster address translation -> faster code execution
  - Con: Larger memory footprint
  - Solaris 11 will grow pages in mappings automatically
    - Would be good to be able to disable so that more granular page reference data can be collected if needed.
- Selecting memory locking strategies and other “advice” to the OS



Bloomberg

Engineering

In Closing...

- Memory layout for programs/processes
  - Feedback for memory layout strategies to reduce memory footprint
  - Text vs data
    - Something like valgrind for data “profiling”?
  - Inline vs no-inline?
  - What about interpreted languages and JIT compilers?
    - Interpreters have visibility into data access patterns



Bloomberg

Engineering

References

- Solaris /proc: https://docs.oracle.com/cd/E19253-01/816-5174/proc-4/index.html
- Linux /proc: http://man7.org/linux/man-pages/man5/proc.5.html
- Brendan Gregg's wss estimation: http://www.brendangregg.com/wss.html
- MMU diagram: https://computationstructures.org/lectures/virtual/virtual.html#2
- Page table entry diagram: https://users.soe.ucsc.edu/~scott/courses/Fall05/111/Slides/chapter4.pdf



Bloomberg

Engineering