

# <u>Agenda</u>

- Overview:
  - Where does my program get its memory from?
- Types of expensive memory accesses
- How to find out where they're happening?
- How to resolve them?

# **Background Basics**

## **System Layout**

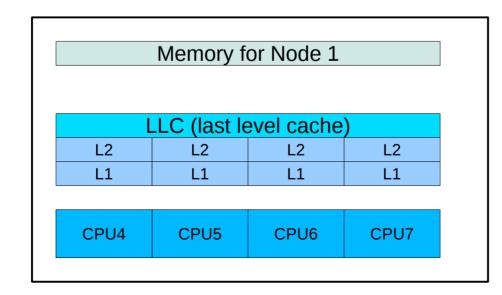
LLC (last level cache)

L2 L2 L2 L2
L1 L1 L1 L1

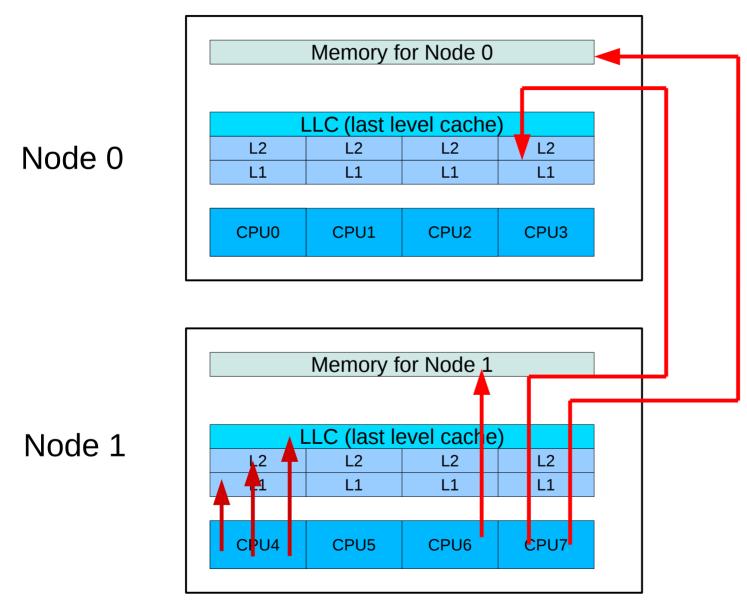
CPU0 CPU1 CPU2 CPU3

Node 0

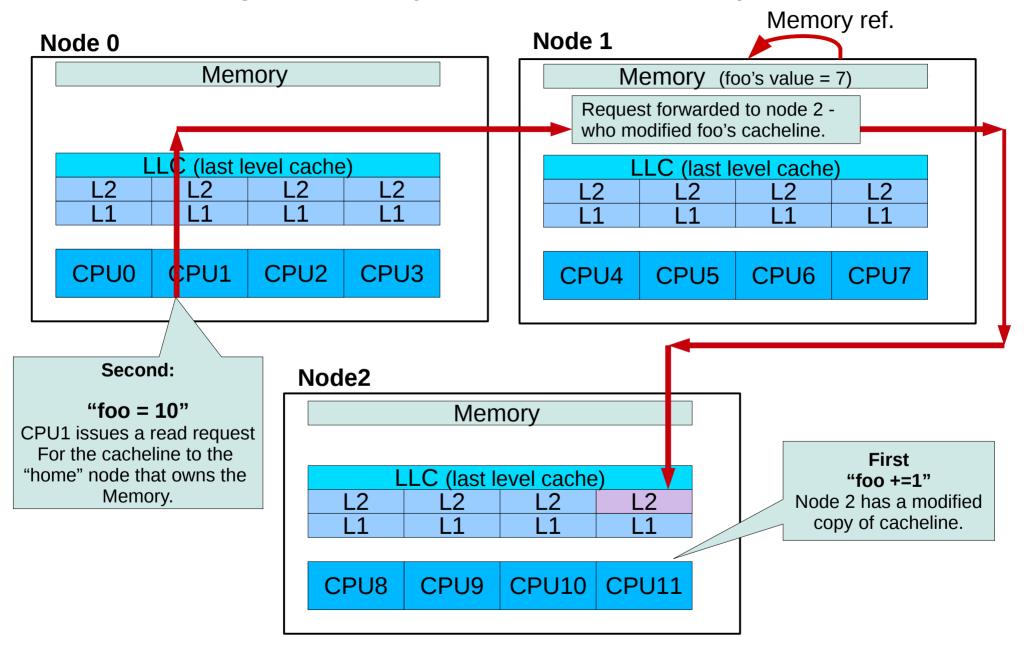
Node 1



# Background Basics Resolving a memory access



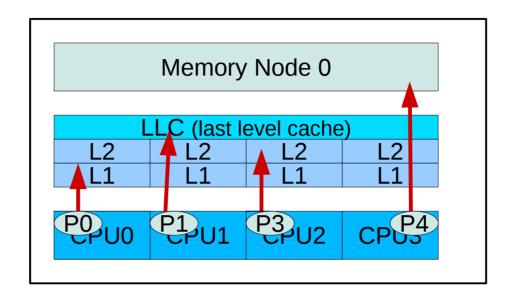
## Resolving a memory access – more expensive case.



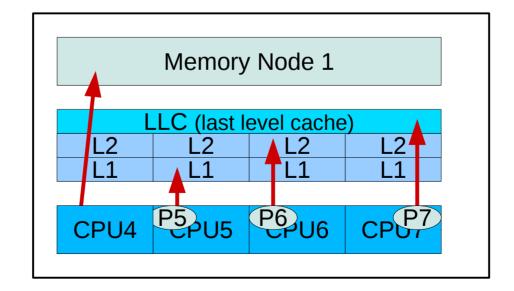
## In the ideal world:

All processes and memory are isolated to their own NUMA nodes.

Node 0



Node 1

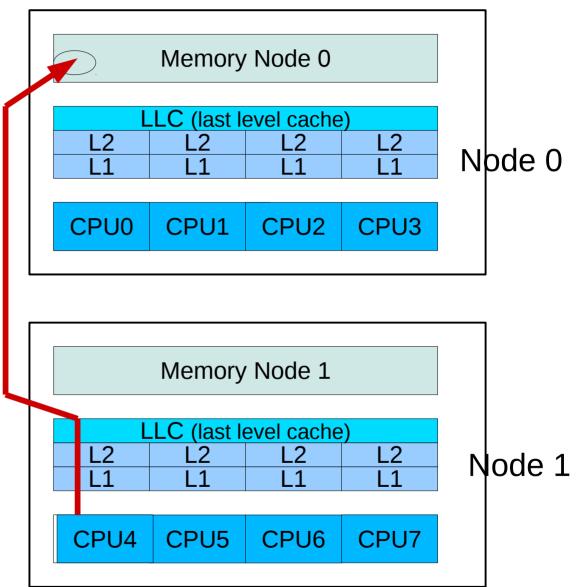


# In the "slightly less than" ideal world

"Sole user" of remote memory.

Not too bad if:

- 1. It fits in local node 1 cache
- 2. It stays in local node 1 cache
- 3. Your node is the only node accessing that memory.



# False Sharing - Where it can hurt the most

Multiple NUMA Memory Node 0 nodes accessing same memory LLC (last level cache) cacheline. Socket 0 POUO CPU1 CPU2 CPU3 Memory Node 1 LLC (last level cache) Socket 1 CPU5 CPU6 CPU7

Jiri Olsa, Joe Mario

# Basic triage steps

What does my system layout look like?

Istopo

Where is my program's memory located?

numastat

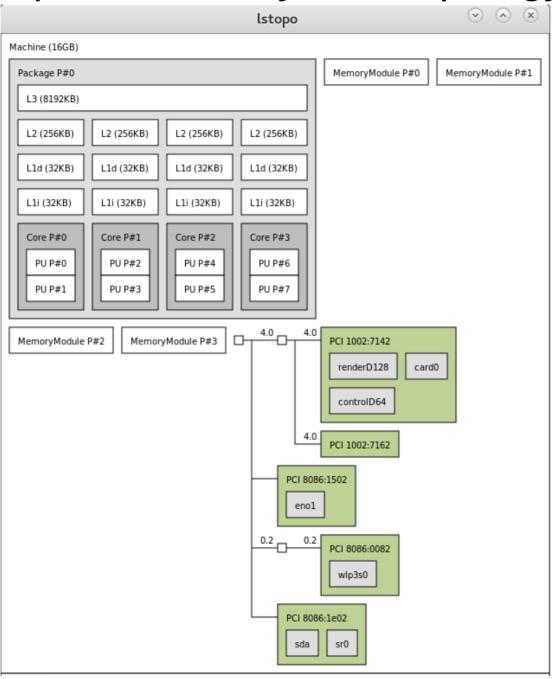
Where are my program's threads executing?

- *ps* -*T* -*o pid*,*tid*,*psr*,*comm* <*pid*>
- Run "top", then enter "f", then select "Last use cpu" field.
- trace-cmd

Where is the memory my program is accessing?

- perf mem
- numatop [Intel]

## Istopo – to see system topology



# Numastat Where is my program's memory?

#### Example:

Look at two unpinned instances of SPECjbb2005.

# numastat -c java

Per-no	ode proce	ess memo	ory usag	ge (in	MBs)
PID		Node 0	Node 1	Total	
31855	(java)	3160	6206	9366	
31856	(java)	4891	4481	9372	
Total		8051	10687	18738	

The memory for each pid is scattered across both numa nodes.

# Where is my program's memory? (continued)

Invoke it again, but with numactl pinning:

# numactl -m 0 -N 0 java <...>

The memory for each pid is confined to a numa node.

# Unanswered questions

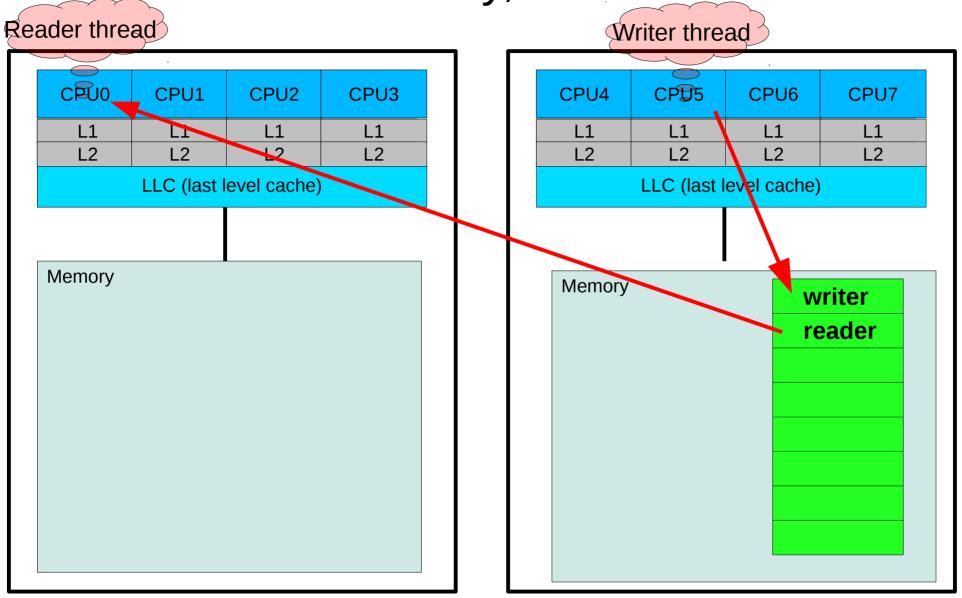
- numastat shows program's memory location, but not threads.
- The key question: Where are my threads executing <u>and are they</u> <u>contending for the same memory/cachelines</u>?
- If your program spans multiple numa nodes:
  - Are my threads accessing memory on remote nodes?
  - If so, how often?
  - Are they in contention for memory locations with other threads (E.G. false sharing)?
  - With multi-threads or shared memory, performance can take a bit hit.

## Look at a simple example false sharing example:

### Two flavors of a basic data structure

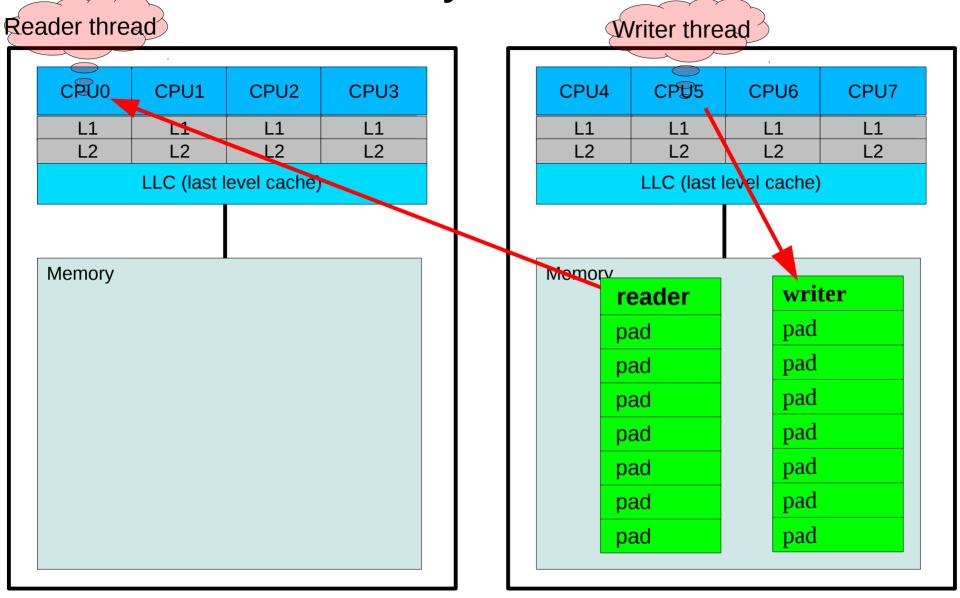
```
struct false sharing_buf {
                                         // Reader & writer
   long writer;
                                          // fields together
   long reader;
} buf ;
struct uncontended buf {
                                         // Writer fields
   long writer;
                                         // separated from
   long pad[7];
                                         // writer field
   long reader1;
   long pad2[7];
} buf;
```

# In memory, first struct:



Jiri Olsa, Joe Mario

# In memory, second struct:



Jiri Olsa, Joe Mario

## Run it through a simple loop:

- Two threads running in parallel.
- Assume buf struct aligned on 64-byte boundary.
- loop-cnt = 500,000,000

```
/* Writer thread on node 0 */
    for (i = 0; i < loop-cnt; ++i) {
        buf.writer += 1;
        asm volatile("rep; nop")
     }</pre>
```

```
/* Reader thread on node 1 */
    for (i = 0; i < loop-cnt; ++i) {
      var = buf.reader;
      asm volatile("rep; nop")
    }</pre>
```

#### **Question**:

How fast can the reader thread complete the loop?

#### Answer:

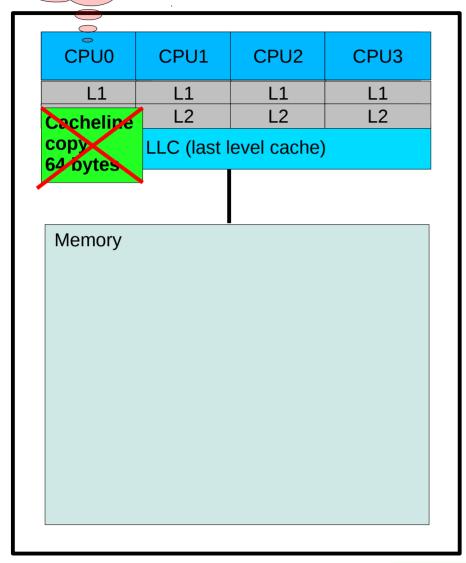
When "buf.writer" is in own cacheline, the reader thread finishes loop **2-4X** faster on 2 node system,

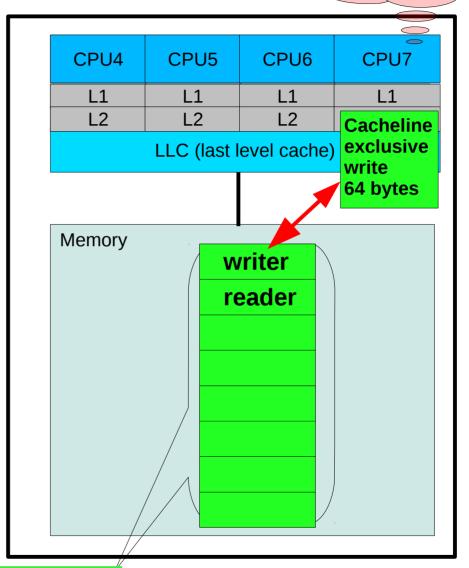
**And up to 20X** faster with multiple readers on a 4 node system.

# Simple false sharing

#### Reader Thread





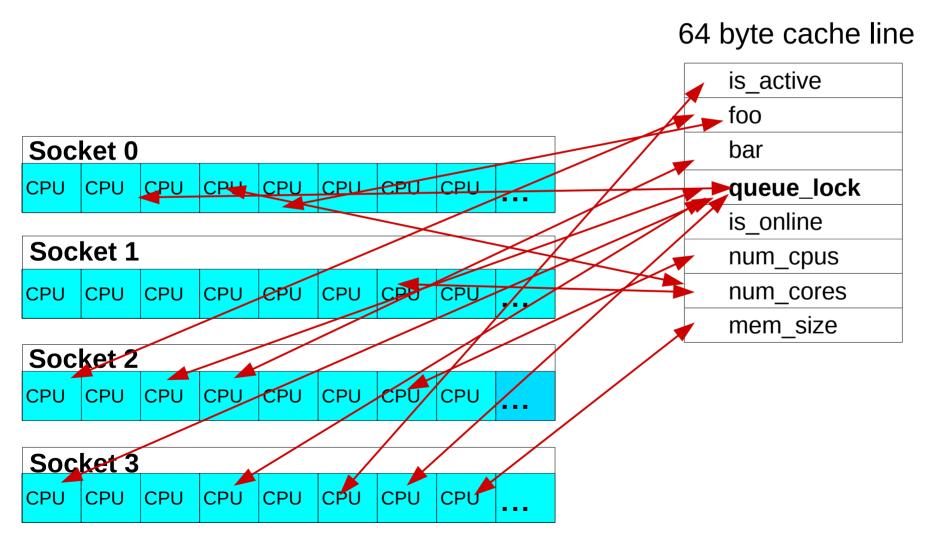


64-byte cache line

# Looking a little closer:

- Every time buf.writer is modified:
  - The reader thread's cacheline copy is disguarded.
  - Must go back for an updated cacheline copy.
  - Or get back in line if other threads are contending for the cacheline.
- With lots of threads and/or large systems:
  - It takes increasingly longer for any one of them to access the cacheline.
  - Often lots longer

# As your application gets larger... Lots of contention.



Jiri Olsa, Joe Mario

# CPU cacheline false sharing

- Multiple threads accessing/modifying same cacheline.
- Multiple processes to same cacheline in shared memory.
- Sharing cachelines across numa nodes costly.
- As are atomic memory operations, e.g. locked instructions, to same cachelines
- Magnified on larger systems (8 and 16 numa nodes)

# How to detect and find this?

New addition to the Linux perf tool: **perf c2c** 

"c2c" stands for "cache to cache"

Developed at Red Hat

Recently merged upstream into 4.9-rc2

Available in RHEL 7.4.

Use on Intel IVB or newer cpus

# At a high level, "perf c2c" provides:

- 1) All the readers and writers to the contended cachelines.
- 2) The cacheline's virtual addr.
- 3) The offsets into the cachelines for those accesses.
- 4) The pid, tid, instruction addr, function name, image filename.
- 5) The source file and line numbers.

# At a high level, "perf c2c" provides:

- 1) The node & cpu numbers where the accesses are occurring.
- 2) The average load latency for the loads.
- 3) Ability to see when hot variables are sharing a cacheline.
- 4) Ability to see unaligned hot data structs spilling into multiple cachelines.

## PERF C2C

record/report command

```
perf c2c record ...
perf c2c report ...
```

- sample INTEL memory events
- load/store memory
  - virtual address
  - type
  - latency (cycles)

perf c2c record [options] -- [record options] <command>

perf c2c record [options] -- [record options] <command>

- c2c record options:
  - -u,-k,-l <latency>

```
perf c2c record [options] -- [record options] <command>
```

- c2c record options:
  - -u,-k,-l < latency>
- standard record options:

perf c2c record [options] -- [record options] <command>

- c2c record options:
  - -u,-k,-l < latency>
- standard record options:

· command:

sleep ..., perf bench sched ...

- system wide, latency, user space, callchains:
  - perf c2c record -u -l 50 -- -a -g sleep 10
- specific process, user space, callchains:
  - perf c2c record -u -- -p 1234 -g
- system wide, kernel space, latency/frequency:
  - perf c2c record -k -l 100 -- -F 80000 -a
- system wide, system wide, workload:
  - perf c2c record -- -a perf bench sched pipe

### PERF C2C REPORT

- sorts and displays data
- TUI/stdio versions

```
perf c2c report --stats
perf c2c report
perf c2c report [--stdio] > out.txt
```

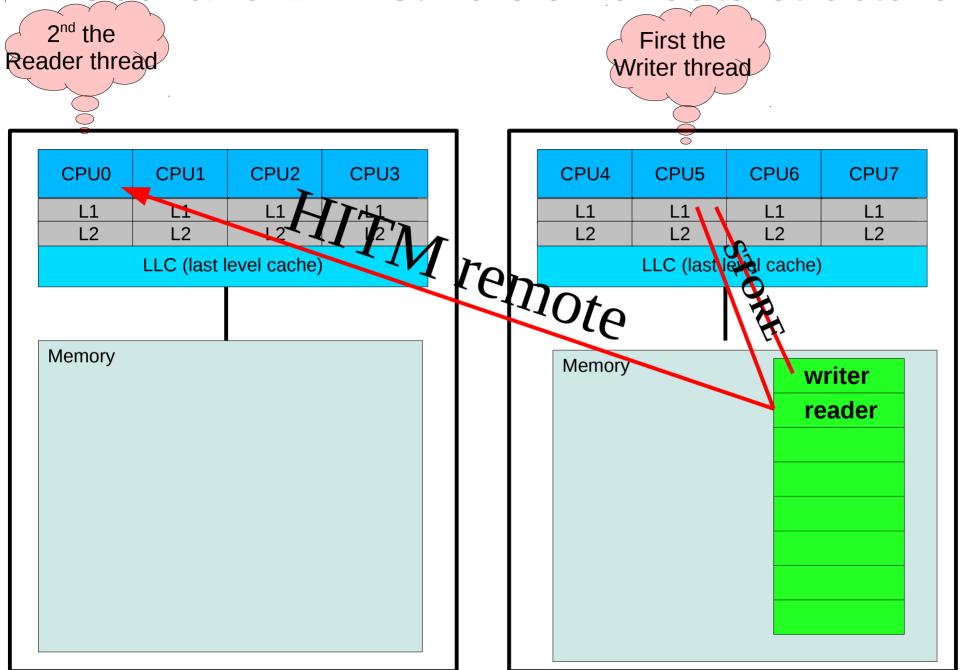
## REPORT - STATS

Trace Event Informatio					
Total records Locked Load/Store Operations	: : :	64860643 5015391			
Load Operations Loads - uncacheable	Lo	ad Local HI	TM	:	6758
Loads - IO Loads - Miss	Lo	ad Remote H	ITM	:	78
Loads - no mapping	:	64230			
Load Fill Buffer Hit	:	1486059			
Load L1D hit	:	1716191			
Load L2D hit	:	515375			
Load LLC hit	:	297056			
Load Local HITM	:	67581			
Load Remote HITM	:	788			
Load Remote HIT	:	338			
Load Local DRAM	:	10			
Load Remote DRAM	:	432			
Load MESI State Exclusive	:	388			
Load MESI State Shared	:	54			
Load LLC Misses	:	1568			
LLC Misses to Local DRAM	:	0.6%			
LLC Misses to Remote DRAM	:	27.6%			
LLC Misses to Remote cache (HIT)	) :	21.6%			
LLC Misses to Remote cache (HITM	(P	50.3%			
Store Operations	:	60346159			
Store - uncacheable	:	0			
Store - no mapping	:	39174			
Store L1D Hit	:	58855217			
Store L1D Miss	:	1451768			
No Page Map Rejects	:	1699			
Unable to parse data source	:	0			

Jiri Olsa, Joe Mario

32

## Scenario 1: First version of data structure



## PERF C2C REPORT

display cachelines that contains:

HITMs STOREs

• 2 tables:

cachelines list single cacheline details

## REPORT – SHARED CACHELINES

	Shared Data Cache																	
#		Total			C Load Hit		C+-	us Dafaus		Load 1	D	LLC	Total	Co:			110 1	
# # Index	Cacheline		Tot Hitm	Total	C Load Hit Lcl	m Rmt	Total	re kerere LlHit	LlMiss	Load I	Dram Rmt		Loads	Co:	re Load Hi L1	L2	LLC Loa	ıa нıt Rmt
#							10001		DIMIDD			Lu IIIDD	Бойць					
#																		
0	0xffff880477000140	12183	3.06%	363	363	0	1001	961	40	0	0	0	11182	1908	5885	2572	454	C
1	0xffff880277400140	11033	2.66%	316	316	0	1037	990	47	0	0	0	9996	1335	5509	2367	469	(
2	0xffff880477000ec0	12142	2.07%	246	246	0	1019	994	25	0	0	0	11123	1748	6049	2611	469	(
3	0xffff880277400ec0	10776	2.07%	245	245	0	993	965	28	0	0	0	9783	1354	5529	2238	417	(
4	0xffff8801aa0dc780	272	0.26%	31	31	0	128	104	24	0	0	0	144	63	7	43	0	(
5	0xffff880277c19740	192	0.24%	29	29	0	93	87	6	0	0	0	99	25	21	24	0	(
6	0xffff88046fa91a00	189	0.18%	21	21	0	108	88	20	0	0	0	81	36	10	14	0	(
7	0xffff88046fa92000	147	0.18%	21	21	0	88	72	16	0	0	0	59	22	5	11	0	(
8	0xffff88047fad9700	153	0.17%	20	20	0	109	109	0	0	0	0	44	2	7	15	0	(
9	0xffff88047fa59740	144	0.16%	19	19	0	72	70	2	0	0	0	72	22	11	20	0	(
10	0xffff88047fc59740	160	0.16%	19	19	0	93	85	8	0	0	0	67	19	12	17	0	(
11	0xffff8801aa0dd600	168	0.13%	16	16	0	101	92	9	0	0	0	67	26	11	14	0	(
12	0xffff8801aa0de800	157	0.13%	16	16	0	86	76	10	0	0	0	71	28	9	18	0	(
13	0xffff88046fc08980	211	0.13%	16	16	0	100	83	17	0	0	0	111	51	10	34	0	(
14	0xffff8801aa0dc600	199	0.13%	15	15	0	125	106	19	0	0	0	74	40	7	12	0	(
15	0xffff8801aa0de780	230	0.13%	15	15	0	116	91	25	0	0	0	114	61	12	26	0	(
16	0xffff880277bd9700	166	0.13%	15	15	0	132	129	3	0	0	0	34	0	4	15	0	(
17	0xffff880277c59700	147	0.13%	15	15	0	111	108	3	0	0	0	36	0	10	11	0	(
18	0xffff88046fa92180	227	0.13%	15	15	0	113	93	20	0	0	0	114	56	11	32	0	(
19	0xffff88047fad9740	177	0.13%	15	15	0	91	84	7	0	0	0	86	32	17	22	0	(
20	0xffff880277bd9780	164	0.12%	14	14	0	50	37	13	0	0	0	114	82	6	12	0	(
21	0xffff88046fa90180	187	0.12%	14	14	0	112	89	23	0	0	0	75	29	8	24	0	(
22	0xffff88046fa91e00	235	0.12%	14	14	0	145	142	3	0	0	0	90	30	27	19	0	(
23	0xffff88046fc09480	157	0.12%	14	14	0	61	61	0	0	0	0	96	48	16	18	0	(
24	0xffff88046fc09c80	177	0.12%	14	14	0	64	64	0	0	0	0	113	46	29	24	0	(
25	0xffff88047fbd9780	142	0.12%	14	14	0	50	44	6	0	0	0	92	57	8	13	0	(
26	0vffff8801aa0dd780	137	0 11%	13	13	0	86	71	15	0	٥	٥	5.1	22	5	11	0	

## REPORT – SHARED CACHELINES

 =====					
Shared	Data	Cache	Line	Table	

	Shared Data C																	
#										٦								
#		Tota	L Tot	LI	C Load Hi	tm	Stc	re Refere	nce	Load Dr	am	LLC	Total	Cc	re Load Hi	it	LLC Loa	d Hit -
# Inde:	c Cachel	ne record	Hitm	Total	Lcl	Rmt	Total	LlHit	LlMiss	Lcl	Rmt	Ld Miss	Loads	FB	L1	L2	Llc	Rr
#													• • • • • • •	• • • • • •				
#	0	1010	2 060	262	262		1001	0.61	4.0				11100	1000	F.0.0F	2572	45.4	
	0 xffff880477000 1 0xffff880277400			363 316	363 316	0	1001 1037	961 990	40 47	0	0	0	11182 9996	1908 1335	5885 5509	2572 2367	454 469	
	0xffff880477000			246	246	0	1019	994	25	0	0	0	11123	1748	6049	2611	469	
	3 0xffff880277400			245	245	0	993	965	28	0	0	0	9783	1354	5529	2238	417	
	0xffff8801aa0dc	780 27	0.26%	31	31	0	128	104	24	0	0	0	144	63	7	43	0	
!	0xffff880277c19			29	29	0	93	87	6	0	0	0	99	25	21	24	0	
	0xffff88046fa91			21	21	0	108	88	20	0	0	0	81	36	10	14	0	
	7 0xffff88046fa92			21	21	0	88	72	16	0	0	0	59	22	5	11	0	
	3 0xffff88047fad9 9 0xffff88047fa59			20 19	20 19	0	109 72	109 70	0 2	0	0	0	44 72	2 22	11	15 20	0	
10				19	19	0	93	70 85	8	0	0	0	67	19	12	17	0	
#					Tota	1	Tot		- LLC	Load H	itm			- Stor	e Refe	erenc	e	
#	Index	C	acheli	ine r	ecord	S	Hitm	To	tal	Lcl		Rmt	To	otal	L1Hi	it :	L1Miss	
#	• • • • • • • • • • • • • • • • • • • •	• • • • • •	• • • • •	• • •	• • • • •	• ••	• • • • •	• • • •	• • •	• • • • • •	• •	• • • • •	••••	• • •	• • • • •		• • • • •	
	0 0xf	fff8804	770001	L40	1218	3	3.06%		363	363		0	1	001	96	51	40	
	1 0xf	Eff8802	774001	L40	1103	3	2.66%		316	316		0	1	.037	99	90	47	
	2 0xf	fff8804	77000€	ec0	1214	2	2.07%		246	246		0	1	.019	99	94	25	
	3 0xf	fff8802	277400€	ec0	1077	6	2.07%		245	245		0		993	96	65	28	
	4 0xf	fff8801	aa0dc7	780	27	2	0.26%		31	31		0		128	10	04	24	

## REPORT – SHARED CACHELINES

		=======		===														
		Total	Tot	LL	C Load Hitm		Store	Refere	nce	Load Dram		LLC	Total	Core	e Load H	it	LLC Loa	d Hit
Index	Cacheline	records	Hitm	Total	Lcl	Rmt	Total	L1Hit	LlMiss	Lcl	Rmt	Ld Miss	Loads	FB	L1	L2	Llc	Rmt
• • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • •	• • • • • • • •	• • • • • • •		• • • • •			• • • • • • •		• • • •	• • • • • • •		• • • • • • •	• • • • • • •	• • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • •
0	0xffff880477000140	12183	3.06%	363	363	0	1001	961	40	0	0	0	11182	1908	5885	2572	454	(
	0xffff880277400140	11033	2.66%	316	316	0	1037	990	47	0	0	0	9996	1335	5509	2367	469	·
	0xffff880477000ec0	12142	2.07%	246	246	0	1019	994	25	0	0	0	11123	1748	6049	2611	469	
3	0xffff880277400ec0	10776	2.07%	245	245	0	993	965	28	0	0	0	9783	1354	5529	2238	417	
4	0xffff8801aa0dc780	272	0.26%	31	31	0	128	104	24	0	0	0	144	63	7	43	0	(
	0xffff880277c19740	192	0.24%	29	29	0	93	87	6	0	0	0	99	25	21	24	0	(
	0xffff88046fa91a00	189	0.18%	21	21	0	108	88	20	0	0	0	81	36	10	14	0	
	0xffff88046fa92000	147	0.18%	21	21	0	88	72	16	0	0	0	59	22	5	11	0	
	0xffff88047fad9700	153	0.17%	20	20	0	109	109	0	0	-0	0	44	2	7	15	0	
	0xffff88047fa59740 0xffff88047fc59740	144 160	0.16%	19 19	19 19	0	72 93	70 85	2	0	0	0	72 67	22 19	11 12	20 17	0	
10 11	0xffff8801aa0dd600	160	0.16% 0.13%	16	16	0	93 101	92		0	0	0	67	26	11	14	0	,
	0xffff8801a					- 0												,
	0xffff88046	Load	Dram		LI	¹C	Total		C	ore Load	Hit			LLC L	oad	Hit	-   0	
	0xffff8801a	Lcl		Rmt	Ld Mis	s	Loads		FB	L1	L	$L_2$	2	Llc		Rm	t 0	
	0xffff8801a 0xffff88027					_					_		_				0	
	0xffff88027	• • • • •	• • • •	• • • •	• • • • • •	•	• • • • • • •	• •	• • • • •	• • • • • •	•	• • • • •	• •	• • • • •	• •	• • • • •	•   0	,
	0xffff88046																0	
	0xffff88047	_		_		_					_						_   0	
	0xffff88027	0		0		0	11182		1908	5885	5	2572	2	454		(	0   0	(
21	0xffff88046	0		Λ		Λ	9996		1335	5509	)	2367	7	469		(	) 0	(
	0xffff88046	-		U		U										•	0	
	0xffff88046	0		0		0	11123		1748	6049	)	2611	_	469		(	)   0	
	0xffff88046	^		^		^	0702		1254	EEOC	,	2220	,	117			0	(
	0xffff88047 0xffff8801a	0		0		0	9783		1354	5529	,	2238	)	417		(	0 0	
26	UXIIII00Ula	0		0		0	144		63	7	7	43	}	0		(	)	'

	nared Cach	e Line Di	stributio												
# # #	НІ	TM	Store		Data address				- cycles -		cpu		Shared		
# Num	Rmt	Lcl	Ll Hit	L1 Miss	Offset	Pid	Code address	rmt hitm	Icl hitm	load	cnt	Symbol	0bject	Source:Line	Node
#	•••••	•••••		•••••			•••••					•••••	••••	•••••	••••
0	1310	1630	1583	999	0x602100										
	0.76%	0.00%	0.00%	0.00%	0x0	108517	0x400b7a	597	0	1158	1	[.] lock_thd	a.out	false_sharing_example.c:134	0
	0.15%	0.00%	66.08%	0.00%	0x0	108517	0x400b61	2780	0	1569	1	[.] lock_thd	a.out	false_sharing_example.c:132	0
	0.00%	0.00%	33.92%	100.00%	0x0	108517	0x400b85	0	0	0	1	[.] lock thd	a.out	false sharing example.c:134	0
	32.67%	29.02%	0.00%	0.00%	0x8	108517	0x400bab	1100	498	1021	12	[.] reader_thd	a.out	false_sharing_example.c:145	0 1 2 3
	31.68%	41.23%	0.00%	0.00%	0x10	108517	0x400bab	1075	542	1074	11	[.] reader thd	a.out	false sharing example.c:145	0 1 2 3
	34.73%	29.75%	0.00%	0.00%	0x18	108517	0x400bab	1008	429	1077	12	[.] reader_thd	a.out	false_sharing_example.c:145	0 1 2 3
1	1240	1515	3088	178	0x602140										
	0.16%	0.00%	97.51%	0.00%	0x0	108517	0x400b55	1379	0	1441	1	[.] lock thd a.ou	t fals	e sharing example.c:131 0	
	0.00%	0.00%	2.49%	100.00%	0x0	108517	0x400b73	0	0	0				e_sharing_example.c:133 0	
	32.66%	30.43%	0.00%	0.00%	0x8	108517	0x400ba0	994	383	998	12				1 2 3
	31.29%	38.02%	0.00%	0.00%	0x10	108517	0x400ba0	1046	423	976	11				1 2 3
	35.89%	31.55%	0.00%	0.00%	0x18	108517	0x400ba0	999	384	1028	12				1 2 3
2	 1	2	2		0xffff887bfd04b100										
	100.00%		0.00%	0.00%	0x0	108517	0xfffffffff810c7f0f	2541	124	474	9	<pre>[k] update_cfs_share</pre>	s	[kernel.vmlinux] compiler.	
	0.00%	0.00%	50.00%	0.00%	0x0	108517	0xffffffff810cbd10	0	0	767	3	[k] task_tick_fair		[kernel.vmlinux] atomic64_	
	0.00%	0.00%	50.00%	0.00%	0x0	0	0xfffffffff810c4bb4	0	0	0	1	[k] update_blocked_a	verages	[kernel.vmlinux] atomic64_	64.h:45 0

\_\_\_\_\_

			istributio ======		====											
Num	HI Rmt	TM	Store	Refs L1 Miss	Data a	ddress Offset	Pid	Code address		cycles lcl hitm	load	cpu cnt	Symbo	Shared		Node
• • •	· · · · · ·							·····					Symbo			····
0	1310	1630	1583	999	0x	602100										
	0.76%	0.00%		0.00%		0x0	108517	0x400b7a	597	0	1158		[.] lock_thd		false_sharing_example.c:134	
	0.15%	0.00%		0.00%		$0 \times 0$ $0 \times 0$	108517 108517	0x400b61 0x400b85	2780 0	0	1569 0		[.] lock_thd [.] lock thd		<pre>false_sharing_example.c:132 false sharing example.c:134</pre>	0
	32.67%	29.02%	0.00%	0.00%		0x8	108517	0x400bab	1100	498	1021	12	[.] reader_thd	a.out	false_sharing_example.c:145	0 1 2
	31.68% 34.73%	41.23% 29.75%	0.00%	0.00%		0x10 0x18	108517 108517	0x400bab 0x400bab	1075 1008	542 429	1074 1077		<pre>[.] reader_thd [.] reader_thd</pre>		<pre>false_sharing_example.c:145 false_sharing_example.c:145</pre>	0 1 2 0 1 2
1	1240	1515	3088	178	0x	602140										
	0.16%	0.00%		0.00%		0x0	108517	0x400b55	1379	0	1441	1	[.] lock thd a.	out fals	e sharing example.c:131 0	
	0.00%	0.00%	2.49%	100.00%		0x0	108517	0x400b73	0	0	0	1	[.] lock_thd a.	out fals	e_sharing_example.c:133 0	
	32.66% 31.29%	30.43% 38.02%		0.00%		0x8 0x10	108517 108517	0x400ba0 0x400ba0	994 1046	383 423	998 976					1 2 3 1 2 3
	35.89%	31.55%		0.00%		0x18	108517	0x400ba0	999	384	1028					1 2 3
#									·							
#		_		HITM			Store	Refs		Dat	a add	ress				13
#	Nu	ım	Rm	ıt	Lcl	L	l Hit	L1 Miss			Of	fset	Pid		Code address	1:45 1:45
#	• • • •	• •	• • • • •		• • • • •	• •	• • • •	• • • • • •	• • • •	• • • • •	• • • • •	• • • •	• • • • • •	••••	••••••	
		0	131	.0	1630		1583	999			0x60	2100				
			0.76	 ;ह	 0.00%		 ).00%	0.00%				0x0	108517		0x400b7a	
			0.15	િ	0.00%		5.08%	0.00%				0x0	108517		0x400b61	
			0.00	읭	0.00%	33	3.92%	100.00%				0x0	108517		0x400b85	
			32.67		29.02%		0.00%	0.00%				0x8	108517		0x400bab	
			31.68		41.23%		0.00%	0.00%				0x10	108517		0x400bab	
			$\mathbf{J} + 0$	0 0	TI • 2 3 0	,	J • U U U	0.000				OVIO	10001		CATUUDAD	

	nared Cach	e Line Di		n Pareto															
Num		TM Lcl	Store		Data address Offset	Pid	Code address		- cycles - lcl hitm	load	cpu cnt	Sy	Sh ymbol Ob	nared oject		So	ource:Line	Node	
0	1310	1630	1583	999	0x602100														
	0.76% 0.15% 0.00% 32.67% 31.68% 34.73%	0.00% 0.00% 0.00% 29.02% 41.23% 29.75%	0.00% 66.08% 33.92% 0.00% 0.00%	0.00% 0.00% 100.00% 0.00% 0.00%	0x0 0x0 0x0 0x8 0x10 0x18	108517 108517 108517 108517 108517	0x400b7a 0x400b61 0x400b85 0x400bab 0x400bab 0x400bab	597 2780 0 1100 1075 1008	0 0 0 498 542 429	1158 1569 0 1021 1074 1077	1 1 12 11	[.] lock_thd	a. a. a.	out foot foot foot foot	alse_sha alse_sha alse_sha alse_sha	ring_examp ring_examp ring_examp ring_examp	ple.c:134 ple.c:132 ple.c:134 ple.c:145 ple.c:145	0 0 0 0 1 2 0 1 2	3
1	1240	1515	3088	178	0x602140														
	0.16% 0.00% 32.66% 31.29% 35.89%	0.00% 0.00% 30.43% 38.02% 31.55%	97.51% 2.49% 0.00% 0.00% 0.00%	0.00% 100.00% 0.00% 0.00% 0.00%	0x0 0x0 0x8 0x10 0x18	108517 108517 108517 108517 108517	0x400b55 0x400b73 0x400ba0 0x400ba0 0x400ba0	1379 0 994 1046 999	0 0 383 423 384	1441 0 998 976 1028	1 12 11	[.] lock thd [.] lock thd [.] reader thd [.] reader thd [.] reader thd	a.out a.out a.out	false_ false_ false_	sharing_ sharing_ sharing_	example.c example.c example.c example.c	:133 0 :144 0 1 :144 0 1	. 2 3 . 2 3 . 2 3	
2																			
	100.0		cpu			Symbo	Shared Object				Sc	ource:Lir	ne N	Node	<b>!</b>		compiler.h	4.h:45	0 1 0 1 0
		• • •	• • • •	• • •	• • • • • • • • •	• • • • •	• •••••	• • •	• • • •	• • • •	• • • •	•••••	• •						
							2 011+	falc	a aha					`					
			1	. [•]	] lock_thd		a.out	Tals	se_sna	ring_	examp	ole.c:134	1 0	)					
			1 1	· [•]	·		a.out		_		-	ole.c:134 ole.c:132							
			1 1 1		lock_thd			fals fals	e_sha se_sha	ring_ ring_	examp examp	ole.c:132 ole.c:134	2 0 1 0	)					
			1 1 1 12	[•]	lock_thd	hd	a.out	fals fals	e_sha se_sha	ring_ ring_	examp examp	le.c:132	2 0 1 0	)	2	3			
			1 1 1 12 11	[•]	lock_thd lock_thd reader_t		a.out	fals fals fals	se_sha se_sha se_sha	ring_ ring_ ring_	examp examp	ole.c:132 ole.c:134	2 0 1 0 5 0	) ) ) 1		3			

Jiri Olsa, Joe Mario

40

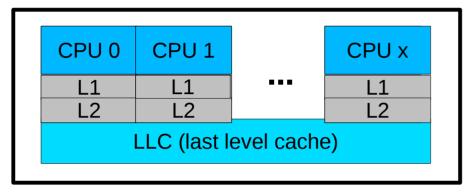
	hared Cach	e Line Di	stributio												
# # # Nur #	HI		Store		Data address Offset	Pid	Code address			load	cpu cnt	Symbol Ob	nared ject	Source:Line	
*	1310	1630	1583	999	0x602100										
	0.76% 0.15% 0.00% 32.67% 31.68% 34.73%	0.00% 0.00% 0.00% 29.02% 41.23% 29.75%	0.00% 66.08% 33.92% 0.00% 0.00%	0.00% 0.00% 100.00% 0.00% 0.00%	0x0 0x0 0x0 0x8 0x10	108517 108517 108517 108517 108517 108517	0x400b7a 0x400b61 0x400b85 0x400bab 0x400bab 0x400bab	2780 0 1100 1075	0 0 0 498 542 429	1158 1569 0 1021 1074 1077	1 1 12 11	[.] lock_thd a. [.] lock_thd a. [.] reader_thd a. [.] reader_thd a.	out false_sharing_exa out false_sharing_exa out false_sharing_exa out false_sharing_exa out false_sharing_exa	mple.c:132 mple.c:134 mple.c:145 mple.c:145	0 0 0 0 1 2 3 0 1 2 3 0 1 2 3
	1240	1515	3088	178	0x602140										
	0.16% 0.00% 32.66% 31.29% 35.89%	0.00% 0.00% 30.43% 38.02% 31.55%	97.51% 2.49% 0.00% 0.00% 0.00%	0.00% 100.00% 0.00% 0.00% 0.00%	0x0 0x0 0x8 0x10 0x18	108517 108517 108517 108517 108517	0x400b55 0x400b73 0x400ba0 0x400ba0 0x400ba0	0 994 1046	0 0 383 423 384	1441 0 998 976 1028	1 12 11		false_sharing_example.	c:133 0 c:144 0 c:144 0	1 2 3 1 2 3 1 2 3
	1	2	2	0	0xffff887bfd04b100										
	100.00% 0.00% 0.00%	100.00% 0.00% 0.00%	0.00% 50.00% 50.00%	0.00% 0.00% 0.00%	0x0 0x0 0x0		hitm lc	ycles l hitm	 1	load		<pre>[k] update_cfs_shares [k] task_tick_fair [k] update_blocked_aver</pre>	[kernel.vmlinux] [kernel.vmlinux] rages [kernel.vmlinux]	atomic64_	64.h:45 0 1 2
							597 2780	0		1158 1569					
							0	Ö		0					
							1100	498	}	1021					
							1075	542		1074					
							1008	429		1077					

\_\_\_\_\_

- perf multi threaded report
- new perf feature by Namhyung Kim
- multiple threads reading/parsing data file
- It is faster... of course ;-)

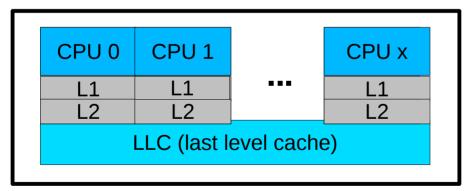
### PERF MULTI THREADED REPORT

#### Node 0



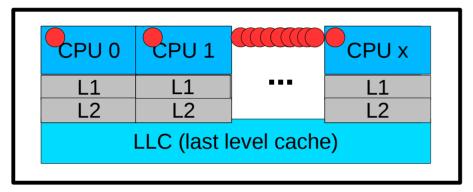
 multi threaded report creates thread on every CPU in system

#### Node 1



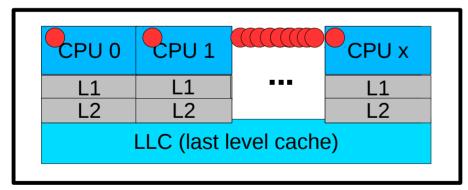
### PERF MULTI THREADED REPORT

#### Node 0



 multi threaded report creates thread on every CPU in system

#### Node 1



#### -----

#### Shared Data Cache Line Table

=======	=======	=======	 ========

#																		
#		Total	Tot	LL	C Load Hi	tm	Sto	re Refere	nce	Load	Dram	LLC	Total	Co	re Load H	it	LLC Lo	ad Hit
# Index	Cacheline	records	Hitm	Total	Lcl	Rmt	Total	L1Hit	LlMiss	Lcl	Rmt	Ld Miss	Loads	FB	L1	L2	Llc	Rmt
#																		
#																		
0	0x2c0d880	5721043	77.20%	44115	43701	414	3686291	3607172	79119	0	145	584	2034752	741904	994452	213480	40631	25
1	0x2c0d7c0	189627	6.90%	3942	3870	72	8430	8307	123	0	29	150	181197	31943	97518	45751	1965	49
2	0x2c0d800	114597	6.73%	3844	3802	42	6932	4511	2421	0	0	47	107665	88022	6075	9639	80	5
3	0x2c2a280	40187	2.80%	1601	1570	31	3494	387	3107	0	0	65	36693	3584	6910	24123	441	34
4	0x2c0d8c0	43943	0.07%	40	0	40	0	0	0	0	4	148	43943	1106	11805	27916	2968	104

#### \_\_\_\_\_

Shared Data Cache Line Table

46

#										1								
#		Total	Tot	LL	C Load Hi	tm	Sto	re Refere	nce	Load	Dram	LLC	Total	Co	re Load H	it	LLC Lo	ad Hit
# Index	Cacheline	records	Hitm	Total	Lcl	Rmt	Total	L1Hit	LlMiss	Lcl	Rmt	Ld Miss	Loads	FB	L1	L2	Llc	Rmt
#																		
#																		
0	0x2c0d880	5721043	77.20%	44115	43701	414	3686291	3607172	79119	0	145	584	2034752	741904	994452	213480	40631	25
1	0x2c0d7c0	189627	6.90%	3942	3870	72	8430	8307	123	0	29	150	181197	31943	97518	45751	1965	49
2	0x2c0d800	114597	6.73%	3844	3802	42	6932	4511	2421	0	0	47	107665	88022	6075	9639	80	5
3	0x2c2a280	40187	2.80%	1601	1570	31	3494	387	3107	0	0	65	36693	3584	6910	24123	441	34
4	0x2c0d8c0	43943	0.07%	40	0	40	0	0	0	0	4	148	43943	1106	11805	27916	2968	104

L1Miss	t L1  t Hit	Total	Rmt	Lcl	Total	Hitm	records
• • • • • •	• • • • • •	• • • • • •	• • • • • •	• • • • • •	• • • • • •	• • • • • •	• • • • • •
79119	3607172	3686291	414	43701	44115	77.20%	5721043
123	8307	8430	72	3870	3942	6.90%	189627
2421	4511	6932	42	3802	3844	6.73%	114597
3107	387	3494	31	1570	1601	2.80%	40187
0	0	0	40	0	40	0.07%	43943

Shared	Cache	Line	${\tt Distribution}$	Pareto

					====										
#	HI	TM	Store	Refs	Data address				- cycles -		cpu		Shared		
# Num	Rmt	Lcl	L1 Hit	L1 Miss	Offset	Pid	Code address	rmt hitm	lcl hitm	load	cnt	Symbol	Object	Source:Line	Node
# ····· #	• • • • • • • • • • • • • • • • • • • •	•••••	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	•••••	• • • • • • • • • • • • • • • • • • • •	•••••	•••••	• • • • • • • • • • • • • • • • • • • •	•••••	•••••	•••••	•••••	• • • • • • • • • • • • • • • • • • • •	• • • •
0	414	43701	3607172	79119	0x2c0d880										
	12.80%	13.05%	0.00%	0.00%	0x18	14493	0x4ad298	296	151	364	24	[.] maps find by time	perf.old	map.c:866	0 1
	22.22%	26.02%	27.44%	0.43%	0x20	14493	0x7fa88a71e302	1000	476	612	24	[.]pthread_rwlock_rdlock	libpthread-2.24.so	.:0	0 1
	11.35%	7.94%	10.56%	0.71%	0x20	14493	0x7fa88a71effb	777	466	588	24	[.] pthread rwlock unlock	libpthread-2.24.so	.:0	0 1
	2.66%	3.08%	17.78%	0.00%	0x20	14493	0x7fa88a71e35a	784	468	444	24	[.]pthread_rwlock_rdlock	libpthread-2.24.so	.:0	0 1
	3.14%	2.55%	20.26%	0.13%	0x20	14493	0x7fa88a72238f	588	402	438		[.]lll_lock_wait	libpthread-2.24.so	.:0	0 1
	2.90%	2.22%	17.36%	0.00%	0x20	14493	0x7fa88a71f0b5	390	432	471		[.]pthread_rwlock_unlock	libpthread-2.24.so		0 1
	0.00%	0.00%	0.00%	0.00%	0x20	14493	0x7fa88a722393	0	5833	6490		[.]lll_lock_wait	libpthread-2.24.so	.:0	0 1
	0.00%	0.00%	0.00%	0.00%	0x20	14493	0x7fa88a71e307	0	0	5276		[.]pthread_rwlock_rdlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.00%	0.00%	0.00%	0x20	14493	0x7fa88a71e35e	0	0	4976	11	[.]pthread_rwlock_rdlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.00%	0.00%	0.00%	0x20	14493	0x7fa88a71efff	0	0	5150		[.]pthread_rwlock_unlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.00%	0.00%	0.00%	0x20	14493	0x7fa88a71f0b8	0	0	4946		[.]pthread_rwlock_unlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.00%	0.01%	0.38%	0x20	14493	0x7fa88a722412	0	0	0	24	[.]lll_unlock_wake	libpthread-2.24.so	.:0	0 1
	0.00%	0.31%	0.00%	0.00%	0x24	14493	0x7fa88a71e337	0	152	344		[.]pthread_rwlock_rdlock	libpthread-2.24.so	.:0	0 1
	0.24%	0.29%	0.00%	0.00%	0x24	14493	0x7fa88a71f022	733	149	391	24	[.]pthread_rwlock_unlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.16%	0.00%	0.00%	0x24	14493	0x7fa88a71efe1	0	158	480		[.]pthread_rwlock_unlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.00%	3.37%	47.01%	0x24	14493	0x7fa88a71e340	0	0	0	24	[.]pthread_rwlock_rdlock	libpthread-2.24.so		0 1
	0.00%	0.00%	3.22%	51.34%	0x24	14493	0x7fa88a71f02a	0	0	0		[.]pthread_rwlock_unlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.01%	0.00%	0.00%	0x30	14493	0x7fa88a71e470	0	174	280		[.]pthread_rwlock_rdlock	libpthread-2.24.so		0 1
	0.00%	0.00%	0.00%	0.00%	0x30	14493	0x7fa88a71f041	0	153	322		[.]pthread_rwlock_unlock	libpthread-2.24.so		0 1
	0.00%	0.21%	0.00%	0.00%	0x34	14493	0x7fa88a71e327	0	170	340	24	[.]pthread_rwlock_rdlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.02%	0.00%	0.00%	0x34	14493	0x7fa88a71f033	0	152	335		[.]pthread_rwlock_unlock	libpthread-2.24.so		0 1
	0.48%	2.59%	0.00%	0.00%	0x38	14493	0x7fa88a71e31f	266	148	340	24	[.]pthread_rwlock_rdlock	libpthread-2.24.so	.:0	0 1
	0.72%	2.14%	0.00%	0.00%	0x38	14493	0x7fa88a71f017	317	146	394	24	[.]pthread_rwlock_unlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.43%	0.00%	0.00%	0x38	14493	0x7fa88a71efd8	0	162	473		[.]pthread_rwlock_unlock	libpthread-2.24.so		0 1
	41.06%	34.16%	0.00%	0.00%	0x3c	14493	0x7fa88a71e2d1	310	146	398		[.]pthread_rwlock_rdlock	libpthread-2.24.so		0 1
	2.42%	4.35%	0.00%	0.00%	0x3c	14493	0x7fa88a71efc0	416	166	481		[.]pthread_rwlock_unlock	libpthread-2.24.so		0 1
	0.00%	0.26%	0.00%	0.00%	0x3c	14493	0x7fa88a71e356	0	194	353		[.]pthread_rwlock_rdlock	libpthread-2.24.so		0 1
	0.00%	0.20%	0.00%	0.00%	0x3c	14493	0x7fa88a71f0b2	0	176	404		[.]pthread_rwlock_unlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.00%	0.00%	0.00%	0x3c	14493	0x7fa88a71e35a	0	4983	0	1	[.]pthread_rwlock_rdlock	libpthread-2.24.so	.:0	0

Shared	Cache	Line	Distribution	Pareto

Num	HIT Rmt		Store L1 Hit	L1 Miss	Data addı Ofi	fset Pid	Code address			load	cpu cnt	Symbol	-	ct Source:Lir	ne No
0	414	43701	3607172	79119	0x2c0	 1880									
	12.80%	13.05%	0.00%	0.00%		 0x18 14493	0x4ad298	296	151	364	24 [.	maps find by time	perf.old	map.c:866	0
	22.22%	26.02%	27.44%	0.43%	(	0x20 14493	0x7fa88a71e302	1000	476	612	24 [.	]pthread_rwlock_rdlock	libpthread-2.24.	so .:0	0
	11.35%	7.94% 3.08%	10.56% 17.78%	0.71%		0x20 14493 0x20 14493	0x7fa88a71effb 0x7fa88a71e35a	777 784	466 468	588 444		<pre>]pthread_rwlock_unlock ]pthread_rwlock_rdlock</pre>	libpthread-2.24. libpthread-2.24.		C
	3.14%	2.55%	20.26%	0.13%		0x20 14493	0x7fa88a72238f	588	402	438	24 [.]	] lll lock wait	libpthread-2.24.	so .:0	(
	2.90% 0.00%	2.22% 0.00%	17.36% 0.00%	0.00%		0x20 14493 0x20 14493	0x7fa88a71f0b5 0x7fa88a722393	390 0	432 5833	471 6490	19 [.	]pthread_rwlock_unlock ]lll_lock_wait	libpthread-2.24. libpthread-2.24.		
	0.00%	0.00%	0.00%	0.002		14493	0v7fa88a71a307	0	0	5276	14 [	nthread rwlock rdlock	libnthroad_2 24	eo •0	
	0.00%	0.00% 0.00%	0.00%	°: #											
	0.00%	0.00%	0.00%	o: #		Н	ITM	9	Store	Refs		Data addre	SS		
	0.00% 0.24%	0.31% 0.29%	0.00%	0: #	Num	Rmt			Hit	L1 Miss		Offs		id	
	0.00% 0.00%	0.16% 0.00%	0.00% 3.37%	0: #	• • • •		• • • • • •	• • • •			• •			• •	
	0.00%	0.00% 0.01%	3.22% 0.00%	51. #											
	0.00%	0.00%	0.00%	0.											
	0.00%	0.21% 0.02%	0.00%	0.											
	0.48%	2.59%	0.00%	0.	0	414	43701	3607	/1/2	79119		0x2c0d8	80		
	0.72% 0.00%	2.14%	0.00%	0.											
	41.06%	34.16% 4.35%	0.00%	0.		12.80%	13.05%	0.	.00%	0.00%		0x	18 144	93	
	2.42% 0.00%	0.26%	0.00%	0.		22.22%			.44%	0.43%		0x			
	0.00%	0.20% 0.00%	0.00%	0.											
	0.000	0.000	0.000	Ľ		11.35%			.56%	0.71%		0x			
						2.66%	3.08%	17.	.78%	0.00%		0x	20 144	93	
						3.14%	2.55%	20.	.26%	0.13%		0x	20 144	93	
						2.90%	2.22%	17.	.36%	0.00%		0x	20 144	93	
						0.00%	0.00%	0.	.00%	0.00%		0x	20 144	93	
						0.00%	0.00%		.00%	0.00%		0x			
						0.00%	0.00%	0.	.00%	0.00%		0x	20 144	93	
						0.00%	0.00%	0.	.00%	0.00%		0x	20 144	93	
						0.00%	0.00%	0.	.00%	0.00%		0x	20 144	93	
						0.00%	0.00%		.01%	0.38%		0x			
						0.00%			.00%	0.00%		0x			

					===												
m.	HIT Rmt	M Lcl	Store L1 Hit		Data address Offset	Pid	Code address		cycles	load	cpu cnt		Symbol		Shared Object	Source:Lin	e No
•	•••••	•••••	•••••		•••••	•••••	•••••	•••••	•••••	•••••		•••••	• • • • • • •		• • • • • •	• • • • • • • • • • • • • • • • • • • •	
0	414	43701	3607172	79119	0x2c0d880												
	12.80%	13.05%	0.00%	0.00%	0x18	14493	0x4ad298 0x7fa88a71e302	296	151 476	364	24			perf.old	2 24 ==	map.c:866	
	22.22% 11.35%	26.02% 7.94%	27.44% 10.56%	0.43%	0x20 0x20	14493 14493	0x/fa88a/le302 0x7fa88a7leffb	1000 777	476	612 588	24	[.]pthread_rwlock_rd [.] pthread rwlock un		libpthread- libpthread-		.:0	
	2.66%	3.08%	17.78%	0.00%	0x20	14493	0x7fa88a71e35a	784	468	444	24	[.]pthread_rwlock_rd		libpthread-			
	3.14%	2.55%	20.26%	0.13%	0x20	14493	0x7fa88a72238f	588	402	438	24	[.]lll_lock_wait		libpthread-			
	2.90% 0.00%	2.22%	17.36% 0.00%	0.00%	0x20 0x20	14493 14493	0x7fa88a71f0b5 0x7fa88a722393	390 0	432 5833	471 6490	24 19	[.]pthread_rwlock_un		libpthread- libpthread-			
	0.00%	0.00%	0.00%	0.00%	0x20	14493	0x7fa88a71e307	0	0	5276	14			libpthread-			
		cpu								Share	d						
		cnt					Symbol			Obje	c+	Source: I	T.ine	Node			
										ے ر من	しし	DOUTCE.1	птис	Node			
		0110					. <u> </u>										
		••••										• • • • • • • • •					
	• • • •	••••	• • •	• • • • •	• • • • • • • • • • • • • • • • • • • •	• • • •		• • • • •	• • • • •	• • • • •	• •	• • • • • • • • • • • •	• • • •	• • • •			
	••••	• • • •	[.]	mans	find by	+ i me		nerf.	old	••••	••	map.c:866	••••	0 1	1		
	••••	24	[.]	maps	find by	_	÷	perf.		-2.24.	• •	map.c:866	••••	0 1	1		
		24 24	[•]	pt	hread_rwlo	ock_r	edlock	libpt	hread	l-2.24.	so	.:0	• • • •		1 1		
		24 24 24	[·] [·]	pt pt	hread_rwlo	ock_r	edlock	libpt libpt	hread hread	-2.24.	so so	.:0 .:0	••••		1 1 1		
		24 24 24 24	[·] [·] [·]	pt pt	hread_rwlo hread_rwlo hread_rwlo	ock_r ock_u ock_r	edlock	libpt libpt libpt	hread hread hread	l-2.24. l-2.24.	so so so	.:0 .:0 .:0	••••		1 1 1 1		
		24 24 24 24 24	[·] [·] [·]	pt pt pt 11	hread_rwlo hread_rwlo hread_rwlo l_lock_wa	ock_r ock_u ock_r it	edlock inlock cdlock	libpt libpt libpt libpt	hread hread hread hread	l-2.24. l-2.24. l-2.24.	so so so	.:0 .:0 .:0	••••		1 1 1 1		
		24 24 24 24 24 24 24	[·] [·] [·] [·]	pt pt pt 11 pt	hread_rwlo hread_rwlo hread_rwlo l_lock_wai hread_rwlo	ock_r ock_r ock_r it ock_r	edlock inlock cdlock	libpt libpt libpt libpt libpt	hread hread hread hread hread	l-2.24. l-2.24. l-2.24. l-2.24.	so so so so	.:0 .:0 .:0 .:0	••••		1 1 1 1 1		
		24 24 24 24 24 24 24	[·] [·] [·] [·]	pt pt pt 11 pt	hread_rwlo hread_rwlo hread_rwlo l_lock_wa: hread_rwlo l_lock_wa:	ock_r ock_r ock_r it ock_r	edlock inlock inlock	libpt libpt libpt libpt libpt libpt	hread hread hread hread hread hread	-2.24.  -2.24.  -2.24.  -2.24.	so so so so so	.:0 .:0 .:0 .:0 .:0	••••		1 1 1 1 1 1		
		24 24 24 24 24 24 24 19	[·] [·] [·] [·]	pt pt pt 11 pt pt	hread_rwlo hread_rwlo hread_rwlo l_lock_wa: hread_rwlo l_lock_wa: hread_rwlo	ock_r ock_u ock_r it ock_u it	edlock inlock inlock inlock	libpt libpt libpt libpt libpt libpt libpt	hread hread hread hread hread hread	l-2.24. l-2.24. l-2.24. l-2.24. l-2.24.	so so so so so so	.:0 .:0 .:0 .:0 .:0	••••		1 1 1 1 1 1 1		
		24 24 24 24 24 24 24	[·] [·] [·] [·] [·] [·]	pt pt pt 11 pt pt pt	hread_rwlo hread_rwlo hread_rwlo l_lock_wa: hread_rwlo l_lock_wa: hread_rwlo hread_rwlo	ock_r ock_r ock_r it ock_r it ock_r	edlock edlock edlock edlock edlock	libpt libpt libpt libpt libpt libpt libpt libpt	hread hread hread hread hread hread hread	1-2.24. 1-2.24. 1-2.24. 1-2.24. 1-2.24. 1-2.24.	so so so so so so	.:0 .:0 .:0 .:0 .:0 .:0	••••		1 1 1 1 1 1 1		
		24 24 24 24 24 24 29 14 11	[·] [·] [·] [·]	pt pt pt 11 pt pt pt	hread_rwlo hread_rwlo hread_rwlo l_lock_wa: hread_rwlo hread_rwlo hread_rwlo hread_rwlo	ock_rock_rock_rit ock_rit ock_rock_rock_rock_rock_rock_rock_rock_r	edlock enlock enlock enlock enlock enlock enlock enlock	libpt libpt libpt libpt libpt libpt libpt libpt	hread hread hread hread hread hread hread hread	1-2.24. 1-2.24. 1-2.24. 1-2.24. 1-2.24. 1-2.24. 1-2.24.	so so so so so so so so	.:0 .:0 .:0 .:0 .:0 .:0 .:0	••••		1 1 1 1 1 1 1 1		
		24 24 24 24 24 24 24 19	[·] [·] [·] [·]	ptptptptptpt	hread_rwlo hread_rwlo hread_rwlo l_lock_wa: hread_rwlo l_lock_wa: hread_rwlo hread_rwlo	ock_rock_rock_rit ock_rit ock_rock_rock_rock_rock_rock_rock_rock_r	edlock enlock enlock enlock enlock enlock enlock enlock	libpt libpt libpt libpt libpt libpt libpt libpt libpt libpt	hread hread hread hread hread hread hread hread hread	1-2.24. 1-2.24. 1-2.24. 1-2.24. 1-2.24. 1-2.24.	so so so so so so so so so so so so so s	.:0 .:0 .:0 .:0 .:0 .:0	••••		1 1 1 1 1 1 1 1 1		

```
856 struct map *maps find by time(struct maps *maps, u64 ip, u64 timestamp)
857 {
858
            struct rb node **p;
859
            struct rb node *parent = NULL;
860
            struct map *m;
861
            struct map *best = NULL;
862
            pthread rwlock rdlock(&maps->lock);
863
864
865
            p = &maps->entries.rb node;
            while (*p != NULL) {
866
867
                    parent = *p;
868
                    m = rb entry(parent, struct map, rb node);
                                                                                     map.c
869
                    if (ip < m->start)
```

```
struct maps {
    struct rb_root entries;
    pthread_rwlock_t lock;
};
map.h
```

```
856 struct map *maps find by time(struct maps *maps, u64 ip, u64 timestamp)
857 {
858
            struct rb node **p;
859
            struct rb node *parent = NULL;
860
            struct map *m;
861
            struct map *best = NULL;
862
863
            pthread rwlock rdlock(&maps->lock);
864
865
            p = &maps->entries.rb node;
            while (*p != NULL) {
866
867
                    parent = *p;
868
                    m = rb entry(parent, struct map, rb node);
                                                                                      map.c
                    if (ip < m->start)
869
```

```
struct maps {
    struct rb_root entries;
    pthread_rwlock_t lock;
};

struct maps {
    struct rb_root entries;
    char u[64];
    pthread_rwlock_t lock;
};
```

- pad members
- USE GCC\_\_attribute\_\_((\_\_aligned\_\_(SMP\_CACHE\_BYTES)))

results:

BASE - 4000 sec

FIX – 2189 sec (45% faster)

#### Shared Data Cache Line Table

53

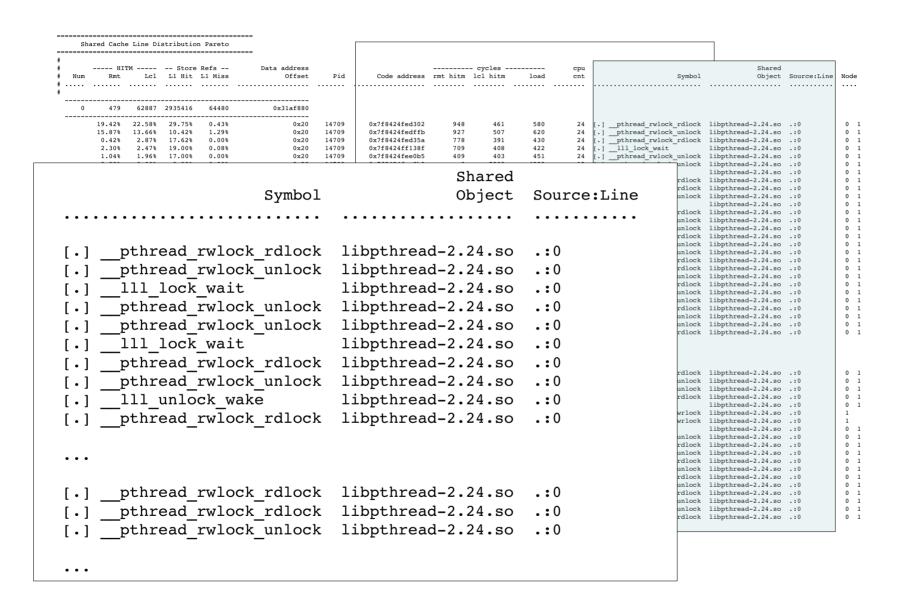
Shared	Data	Cache	Line	Table	

#																		
#		Total	Tot	LL	C Load Hi	tm	Sto	re Refere	nce	Load	Dram	LLC	Total	Co:	re Load H	it	LLC Loa	ad Hit
# Index	Cacheline	records	Hitm	Total	Lcl	Rmt	Total	L1Hit	LlMiss	Lcl	Rmt	Ld Miss	Loads	FB	L1	L2	Llc	Rmt
#																		
#																		
0	0x31af880	5203483	92.68%	63366	62887	479	2999896	2935416	64480	0	150	649	2203587	929482	967733	188052	54784	20
1	0x31afa40	218888	6.44%	4406	4366	40	11007	10979	28	0	2	46	207881	119310	66592	8991	8576	4

Ŧ		Total	Tot	LL	C Load Hit	m	Store	e Refere	ence	· Load D	ram	LLC	Total	Co	re Load H	it	LLC Lo	ad Hit	
# Index	Cacheline	records	Hitm	Total	Lcl	Rmt	Total	L1Hit	L1Miss	Lcl	Rmt	Ld Miss	Loads	FB	L1	L2	Llc	Rmt	
# #	• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •		• • • • • • • • • • • • • • • • • • • •			• • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	• • • • • • • • • • • • • • • • • • • •	
π 0	0x31af880	5203483	92.68%	63366	62887	479	2999896	2935416	64480	0	150	649	2203587	929482	967733	188052	54784	20	
1	0x31afa40	218888	6.44%	4406	4366	40	11007	10979	28	0	2	46	207881	119310	66592	8991	8576	4	
					Total	L	Tot	t -·	LL	ıC Lo	oad H	itm ·			St	ore 1	Refere	ence -	
Index		Cad	cheli	ne 1	Total record		Tot Hit		LL Total		oad H Lc		 Rm		St Tota		Refere		 Miss
Index		Cac	cheli	ne 1														t L1	 Miss
Index 	•••••	• • • •	cheli	• •		ds •••		tm		•		1		t • ••		1		t L1	

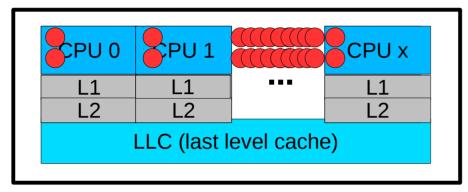
### • ONLY 2 shared cachelines

S	hared Cach	e Line Di	stributio												
	HI	TM	Store	Refs	Data address				- cycles -		cpu		Shared		
Num	Rmt	Lcl	L1 Hit		Offset	Pid	Code address			load	cnt	Symbol		Source:Line	
		• • • • • • • • • • • • • • • • • • • •			• • • • • • • • • • • • • • • • • • • •		•••••						•••••		• • • • •
	479	62887	2935416	64480	0x31af880										
-															
	19.42%	22.58%	29.75%	0.43%	0x20	14709	0x7f8424fed302	948	461	580	24	[.]pthread_rwlock_rdlock		.:0	0 1
	15.87%	13.66%	10.42% 17.62%	1.29%	0x20	14709 14709	0x7f8424fedffb	927 778	507 391	620 430	24	[.]pthread_rwlock_unlock		.:0	0 1
	0.42%	2.87%	17.62%	0.00%	0x20 0x20	14709	0x7f8424fed35a 0x7f8424ff138f	778	408	430	24 24	[.]pthread_rwlock_rdlock [.] lll lock wait	libpthread-2.24.so libpthread-2.24.so	.:0	0 1
	1.04%	1.96%	17.00%	0.00%	0x20	14709	0x7f8424ff138f	409	408	422	24	[.] pthread rwlock unlock		.:0	0 1
	0.00%	0.00%	0.00%	0.00%	0x20	14709	0x7f8424fee0b8	0	5208	4983	15	[.]pthread_rwlock_unlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.00%	0.00%	0.00%	0x20	14709	0x7f8424ff1393	0	4811	5048	14	[.] 111 lock wait	libpthread-2.24.so	.:0	0 1
	0.00%	0.00%	0.00%	0.00%	0x20	14709	0x7f8424fed307	ō	0	4993	13	[.] pthread rwlock rdlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.00%	0.00%	0.00%	0x20	14709	0x7f8424fed35e	0	0	5223	9	[.] pthread rwlock rdlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.00%	0.00%	0.00%	0x20	14709	0x7f8424fedfff	0	0	5706	7	[.] pthread rwlock unlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.00%	0.01%	0.63%	0x20	14709	0x7f8424ff1412	0	0	0	24	[.]lll_unlock_wake	libpthread-2.24.so	.:0	0 1
	0.00%	0.26%	0.00%	0.00%	0x24	14709	0x7f8424fed337	0	161	332	24	[.]pthread_rwlock_rdlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.23%	0.00%	0.00%	0x24	14709	0x7f8424fee022	0	158	332	24	[.]pthread_rwlock_unlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.09%	0.00%	0.00%	0x24	14709	0x7f8424fedfe1	0	222	383	24	[.]pthread_rwlock_unlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.00%	3.05%	46.84%	0x24	14709	0x7f8424fed340	0	0	0	24	[.]pthread_rwlock_rdlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.00%	3.15%	50.73%	0x24	14709	0x7f8424fee02a	0	0	0	24	[.]pthread_rwlock_unlock	libpthread-2.24.so	.:0	
	0.21%	0.01%	0.00%	0.00%	0x30 0x30	14709 14709	0x7f8424fee041 0x7f8424fed470	385 0	244 360	252 283	24 24	[.]pthread_rwlock_unlock [.] pthread_rwlock_rdlock	libpthread-2.24.so libpthread-2.24.so	.:0	0 1
	0.00%	0.01%	0.00%	0.00%	0x34	14709	0x7f8424fed327	0	155	331	24	[.] pthread_rwlock_rdlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.01%	0.00%	0.00%	0x34	14709	0x7f8424fee033	0	154	245	24	[.] pthread_rwlock_rulock	libpthread-2.24.so	.:0	0 1
	0.84%	2.73%	0.00%	0.00%	0x34	14709	0x7f8424fed31f	487	151	330	24	[.] pthread_rwlock_unlock	libpthread-2.24.so	.:0	0 1
	0.42%	1.94%	0.00%	0.00%	0x38	14709	0x7f8424fee017	280	148	331	24	[.] pthread rwlock unlock	libpthread-2.24.so	.:0	0 1
	0.21%	0.53%	0.00%	0.00%	0x38	14709	0x7f8424fedfd8	337	183	378	24	[.] pthread rwlock unlock	libpthread-2.24.so	.:0	0 1
	43.84%	35.76%	0.00%	0.00%	0x3c	14709	0x7f8424fed2d1	346	151	390	24	[.] pthread rwlock rdlock	libpthread-2.24.so	.:0	0 1
	15.03%	14.28%	0.00%	0.00%	0x3c	14709	0x7f8424fedfc0	355	163	392	24	[.] pthread rwlock unlock	libpthread-2.24.so	.:0	0 1
	0.21%	0.21%	0.00%	0.00%	0x3c	14709	0x7f8424fee0b2	333	174	341	24	[.] pthread rwlock unlock	libpthread-2.24.so	.:0	0 1
	0.21%	0.21%	0.00%	0.00%	0x3c	14709	0x7f8424fed356	315	192	345	24	[.]pthread_rwlock_rdlock	libpthread-2.24.so	.:0	0 1
1	40	4366	10979	28	0x31afa40										
						14700	0.7504046.3000	_	207	265			111 11 11 11 11 11 11		
	0.00%	2.06%	19.54%	0.00%	0x0	14709	0x7f8424fed302	0	321 337	289 340	24	[.]pthread_rwlock_rdlock	libpthread-2.24.so	.:0	0 1
	0.00%	1.28%	26.36% 18.35%	3.57% 0.00%	0x0 0x0	14709 14709	0x7f8424fedffb 0x7f8424fee0b5	0	337	340 304	24 24	[.]pthread_rwlock_unlock [.] pthread_rwlock_unlock	libpthread-2.24.so libpthread-2.24.so	.:0	0 1
	0.00%	0.23%	22.60%	0.00%	0x0	14709	0x7f8424fed35a	0	384	276	24	[.]pthread_rwlock_unlock [.]pthread_rwlock_rdlock		.:0	0 1
	0.00%	0.14%	1.99%	0.00%	0x0	14709	0x7f8424ff138f	0	263	269	24	[.] lll lock wait	libpthread-2.24.so	.:0	0 1
	0.00%	0.00%	0.02%	0.00%	0x0	14709	0x7f8424fed8ec	0	0	122	1	[.] pthread rwlock wrlock		.:0	1
	0.00%	0.00%	0.02%	0.00%	0x0	14709	0x7f8424fed923	0	0	137	1	[.] pthread_rwlock_wrlock	libpthread-2.24.so	.:0	1
	0.00%	0.00%	0.01%	7.14%	0x0	14709	0x7f8424ff1412	0	0	0	3	[.] 111 unlock wake	libpthread-2.24.so	.:0	0 1
	0.00%	0.05%	0.00%	0.00%	0x4	14709	0x7f8424fedfe1	0	111	246	24	[.] pthread rwlock unlock		.:0	0 1
	0.00%	0.00%	0.82%	17.86%	0x4	14709	0x7f8424fed340	0	0	0	23	[.] pthread rwlock rdlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.00%	10.29%	71.43%	0x4	14709	0x7f8424fee02a	0	ō	0	24	[.] pthread rwlock unlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.05%	0.00%	0.00%	0x10	14709	0x7f8424fed470	0	136	242	24	[.] pthread rwlock rdlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.18%	0.00%	0.00%	0x18	14709	0x7f8424fee017	0	125	254	24	[.]pthread_rwlock_unlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.16%	0.00%	0.00%	0x18	14709	0x7f8424fed31f	0	112	248	24	[.]pthread_rwlock_rdlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.09%	0.00%	0.00%	0x18	14709	0x7f8424fedfd8	0	120	246	24	[.]pthread_rwlock_unlock	libpthread-2.24.so	.:0	0 1
	92.50%	88.52%	0.00%	0.00%	0x1c	14709	0x7f8424fed2d1	271	119	246	24	[.]pthread_rwlock_rdlock	libpthread-2.24.so	.:0	0 1
	2.50%	3.83%	0.00%	0.00%	0x1c	14709	0x7f8424fedfc0	244	118	261	24	[.]pthread_rwlock_unlock	libpthread-2.24.so	.:0	0 1
	0.00%	0.05%	0.00%	0.00%	0x1c	14709	0x7f8424fee0b2	0	155	264	24	[.]pthread_rwlock_unlock	libpthread-2.24.so	.:0	0 1
	5.00%	2.98%	0.00%	0.00%	0x20	14709	0x7f8424fed2e9	286	120	231	24	[.] pthread rwlock rdlock	libpthread=2.24.so	.:0	0 1



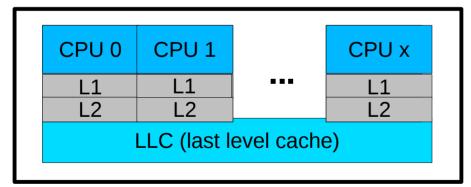
### And both are touching a mutex

#### Node 0



 pin all threads on single socket

#### Node 1



results:

BASE -4000 sec

FIX – 2189 sec (45% faster)

1 SOCKET - 1879 sec (53% faster)

- multi thread perf report does not scale so well ;-)
- c2c identified showed hot variables
- moving to single socket shows the remote/local HITMs speedup

# Steps to help minimize contention:

- 1) Pack read-only/read-mostly variables together.
- 2) Place the hottest written variables in their own cacheline.
- 3) Pad cachelines as a small tradeoff for reducing contention.
- 4) Align your data/buffers/structs/c++classes on cacheline boundaries.
- 5) Lower the granularity of locks (lock smaller chunks of data to reduce contention).
- 6) Use compile-time asserts to guarantee struct member alignment:



# Random Tips

To align dynamically allocated C++ classes on a cacheline boundary:

```
Change:
foo *ctx = new Foo(this, tid);
to:
void *p = aligned_alloc (64, sizeof(Foo));
foo *ctx = new (p) Foo(this, tid);
```

The above ensures the beginning of the class is allocated on a cacheline boundary. And then use "\_\_attribute\_\_((aligned (64))) on individual class members needing cacheline alignment.

C2C prototype copy available at:

http://people.redhat.com/jmario/rhel7\_c2c/perf.rhel7.c2c

Extensive usage info in blog at:

https://joemario.github.io/

# Random Tips

To get more HITMs play with record options:

```
-F <freq> perf record
```

-I <latency> perf c2c record

Higher frequency and latency settings generate more HITMs events.

#### Check following files:

```
echo 100000 > /proc/sys/kernel/perf_event_max_sample_rate
echo 100 > /proc/sys/kernel/perf_cpu_time_max_percent
```