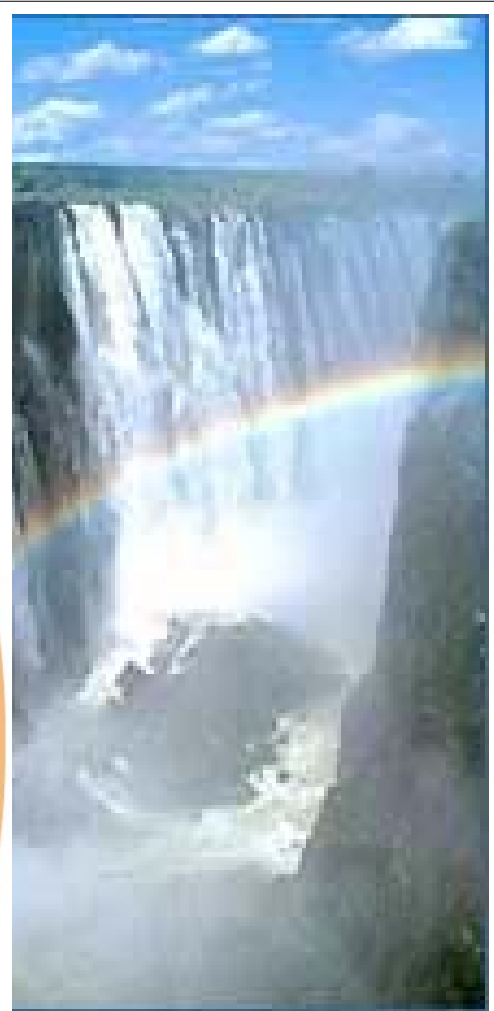


## T2 Solaris changes

### Denis Sheahan

Distinuated Engineer  
Niagara Architecture Group  
Sun Microsystems Inc



## Solaris for T2

- Release vehicle for T2 is S10 Update 4
- Number of performance fixes did not make the release so need patches
- Number generic T2 optimisations
- Also a number of T5x20 specific optimisations
- Also some optimizations coming in S10U5 such as shared context

# Solaris for T2

- Crypto support for T2 hardware
- nxge driver for 10Gig
- T2 Perf counters including counter overflow support
- Page coloring optimizations - Hashed Cache Index
- Out Of the Box Page Performance
- MPSS for non ISM shared memory
- trapstat changes
- Processor Groups optimization
- Stack address skew.
- bcopy and memcpy optimizations including uiomove
- Shared context optimization

·  
·

Page 3

# Performance Counters

- Significant changes from T1 which only had 7, much deeper insight into performance
- Counter overflow works in T2, can be used in the Workshop Analyzer
- Cpustat -h will give details of counters
- Note the format of the command in T2 is back to the standard Sparc implementation where event 1 and event 2 can be any counter

·  
·

Page 4

# Performance Counters

- New counters of interest
  - Idle\_strands – the number of cycles no strand could be picked for the pipeline
  - Instruction breakdown with Instr\_Id Instr\_st Instr\_sw Instr\_other Atomics
  - Hardware Table Walk Counters eg ITLB\_HWTW\_miss\_L2
  - Number of Crypto Hash and Cipher operations eg AES\_op as well as the number of cycles the units are busy eg AES\_busy\_cycle

•  
•

# Performance Counters

- There is a new version of corestat
  - Utilization of multiple integer pipelines
  - Utilization of per core FPU
  - <http://cooltools.sunsource.net/corestat/>

•  
•

## Hash Cache Index

- 64 threads can cause a lot of conflicts in the L2 cache
- The key to reducing conflicts is an optimal placement of memory pages ie the choosing of physical addresses to stop memory aligning on hot lines in the L2 cache.
- T2 hashes (xors) the low order physical address bits required for the cache index with high order physical address bits to randomize accesses to the L2 cache – this is Hash Cache Indexing.
- RFEs 6409758 and 6409758 implement HCI. Increased scaling significantly on Applications

## Out of the Box Page Performance

- Number of fixes to optimize page selection on T1 and T2 more optimal. New defaults are
  - T1
    - Private Anon, Heap, Process Stack, Initdata segments 64KB pages
    - Text, Shared Anon segments 4MB pages
    - ISM and DISM 256MB pages
  - T2 – changes because of HCI
    - Initdata segments 64KB pages
    - Private Anon, Heap, Process Stack, Text, Shared Anon segments 4MB pages
    - ISM and DISM 256MB pages

## Out of the Box Page Performance

- Stack, Heap and Anon values can be changed by using MPSS
- With RFE 6371967 Anon pages created using mmap of /dev/zero will have large pages
- New set of variables to control page size – use with caution
  - max\_uheap\_lpsize
  - max\_ustack\_lpsize
  - max\_privmap\_lpsize
  - max\_shm\_lpsize
  - max\_uidata\_lpsize
  - max\_utext\_lpsize
  - max\_shm\_lpsize

## MPSS for non ISM Shared Memory

- SAP allocates a lot of shared memory but cannot use ISM/DISM because of the need to use mprotect()
- SAP SHM currently on 8k which hurts Niagara performance
- RFE 4614772 adds large page support for non ISM shared memory
- Performance gains of 20% on our internal SAP tests

## Stack Biasing

- Data Cache is 4-way associative but on T2 we can have 8 threads sharing
- When running multiple copies of the same binary on T2 variables on the process stack can become aligned in the L1 Data Cache
- This can cause High L1 cache misses and poor scaling
- RFE 6493685 adds a Stack Bias which slews the stack variables
- Have seen a 19% increase in some of our internal codes

## Trapstat

- T2 has Hardware Table Walk (HWTW) acceleration for both Instruction and Data TLB misses
- Because the TLB miss no longer executed in software trapstat cannot determine the number of misses
- trapstat -T on T2 will report zero for all page sizes in both user and kernel mode
- Percentage time reported will also be zero
- TSB miss still in Software so its percentage time still accurate
- Can still use performance counters to get the TLB miss rate

# Multi-level CMT Scheduling Optimizations

- Introduces Processor Groups
  - > Abstraction introduced to capture a group of CPUs with some (hardware) sharing relationship
    - > int/FP pipelines, caches, chips, MMUs, crypto units, etc.
  - > PGs used by dispatcher to implement multi-level CMT load balancing and affinity policies
  - > Replaces chip\_t
- Niagara II
  - > Groupings created for int/FP pipelines
  - > Balances running threads across both levels
    - > 8 threads => 1 per core (FPU), 1 per integer pipeline
    - > 32 threads => 2 per core (FPU), 2 per integer pipeline
  - > 7% SPECfp\_rate performance improvement on huron (8 processes)
  - > Work to characterize FPU consuming vs. non-consuming threads underway (benefits heterogeneous workloads)

Page 13

## Policy implementation

- disp.c: cmt\_balance()
  - > Replaces chip\_balance()
  - > Uses a top-down load balancing policy to balance running thread load across the levels in the load balancing lineage created by the CMT PG class
  - > Try to balance across the chips, then the cores, then the pipelines etc
- disp.c: disp\_getbest()
  - > Is more aggressive about stealing work if that CPU shares a cache with ours

Page 14

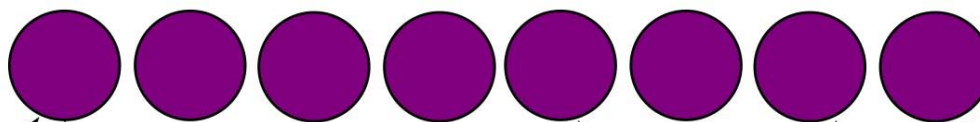
# CMT Processor Group Class

- Example: Niagara II

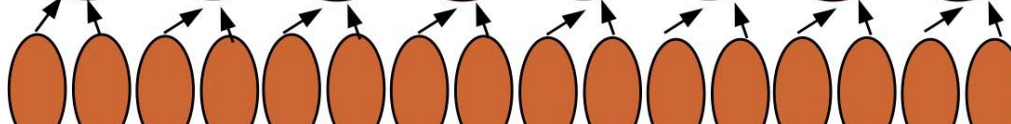
Cache PG



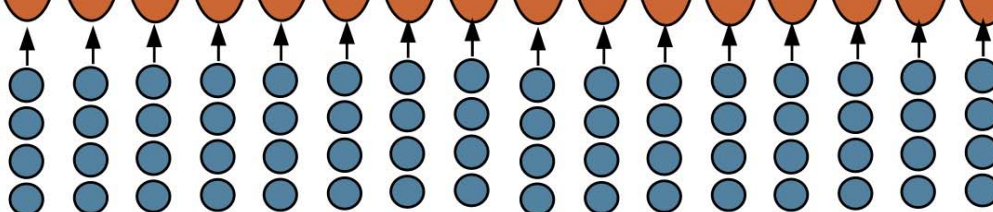
FPU PGs



Integer  
pipe PGs



CPUs



Sun Proprietary / Confidential: Internal Use Only

10

## Bcopy Optimizations

- Bcopy and uiomove are key kernel routines for performance
- RFE 6492718 implements a new bcopy routine for sun4v
- Roughly 2X as fast than current version
- SpecWeb2005 has seen a 10% increase in performance
- Mileage will vary depending on amount of bcopy in a workload
- RFE 6500001 implements the same algorithm in uiomove which copies buffers to and from the kernel

## Performance bugs and Patches

## Patching Huron systems

- A number of key performance fixes have not made it into S10U4
- Kernel patch 127111-05 fixes nearly all the performance issues
- Bottom line make sure you have your system patched with the latest if you are going to do a performance POC with a customer