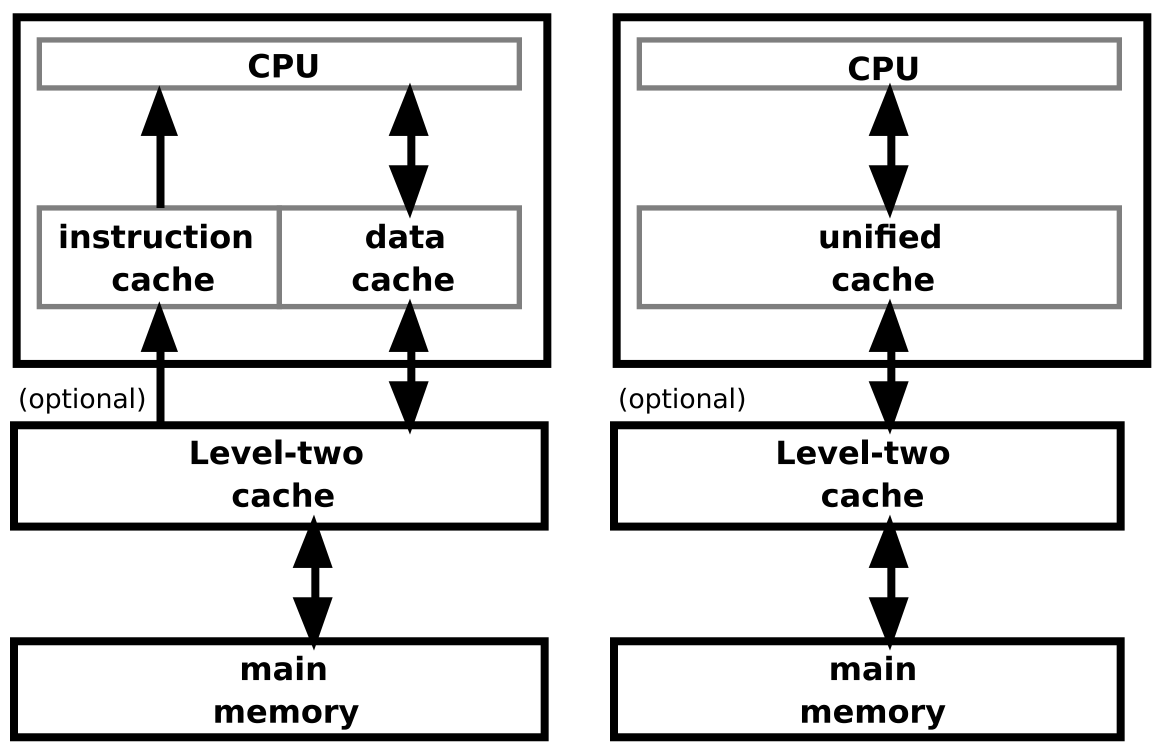
PowerPC Cache Performance

By Bob Wilson, Alan Dail, Matthew Brandt, David B. Chorlian

This report describes some PowerPC benchmarks that measure the performance effects of cache and virtual memory. Cache is a high-speed memory buffer that makes main memory appear to run faster. There are two type of cache, level-one (L1) cache that is built into the CPU and level-two (L2), an external add-in.

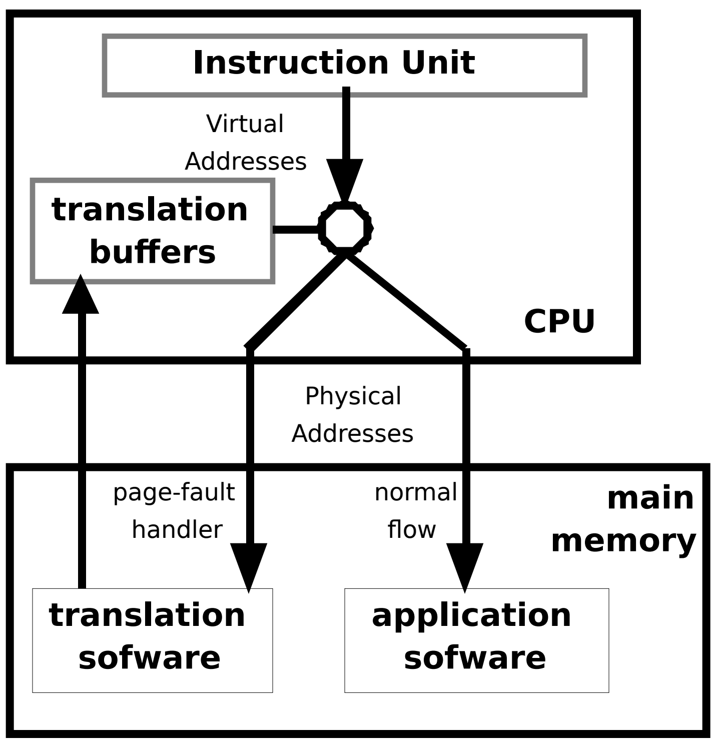
Each type of cache is classified as either split instruction-data (i.e., Harvard architecture) or unified where both instructions and data share the same buffer (See figure). Split cache simplifies the instruction cache circuits, they don't have to update main memory. Furthermore, the code does not have to compete for cache space with the data it is processing. The PowerPC 601 systems have a unified 32 KB L1 cache and optional L2 cache up to 1 MB. However, the 603 uses a split L1 cache.



Cache Architectures and Types

Apple and RAM Doubler by CPU Connectix provide alternate virtual memory (VM) systems. Apple VM is adjustable in size but needs a page file large enough for all virtual memory, both free and active. RAM Doubler has a fixed virtual memory space twice the size of real memory. Furthermore, it allocates the paging file as needed and exploits free memory to reduce paging file size.

Virtual memory speed is driven by how often the software is interrupted by page-faults to update the virtual memory translation buffers (See figure). Translation buffers, part of the CPU, convert virtual addresses to physical addresses but there are a limited number. When a virtual address is needed that is not in the translation buffers, a page-fault occurs which stops the current program. The translation software handles the page-fault but this usually takes tens of microseconds while the application waits. Occasionally, it can take tens of milliseconds if the virtual memory has to be written or read from the page file. In either case, cache entries will be used during the page-fault and the application will briefly run a little slower.

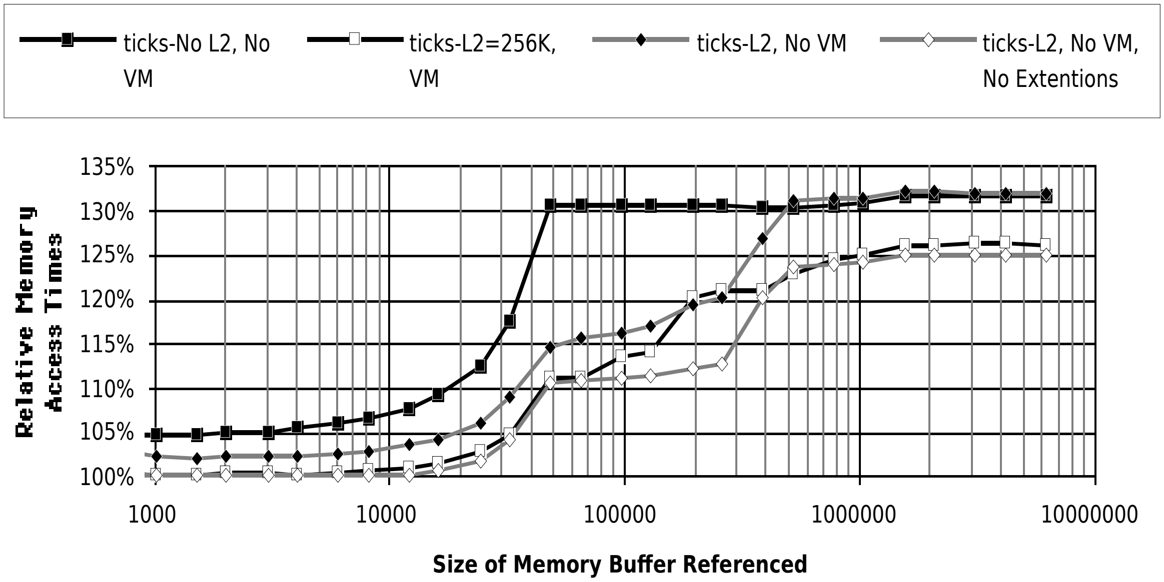


Page-Fault Processing

Virtual memory and cache work because programs usually spend most of their time working in small blocks of code and data. This is called "locality of code and data." Bob Wilson wrote a benchmark program, "memtest.c," that identifies cache and virtual memory effects by using a series of variable-sized memory shifts. These memory shifts slow down as the shift-range grows larger than cache and VM translation boundaries.

The first benchmark was by Alan Dail, adail@infi.net, whose system is:

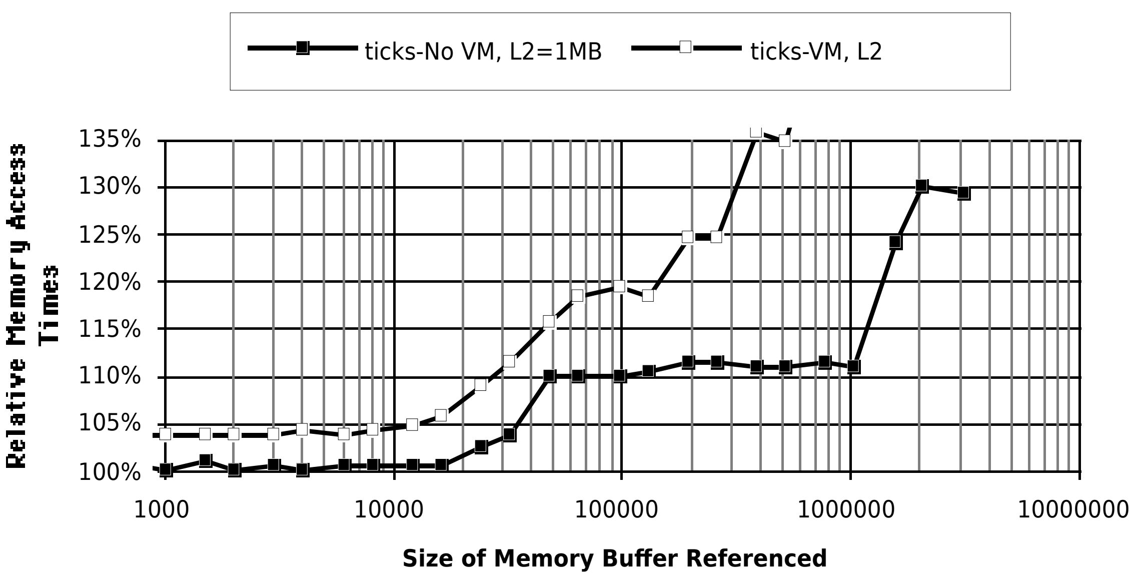
- PowerPC 6100, 80 MHz, 40 MB RAM
- No L2 and 256 KB L2
- VM via RAM Doubler



Alan's data shows that non-native extensions can significantly reduce performance by consuming cache resources. In fact, these extentions impact all cache performance and the signature of their existence is the difference in L1 performance between virtual and real memory operations. This data also shows VM has a 3-8% slow-down in the upper part of the L2 memory range.

The second benchmark series is from Matthew Brandt, matt@atl.sps.mot.com, whose system is:

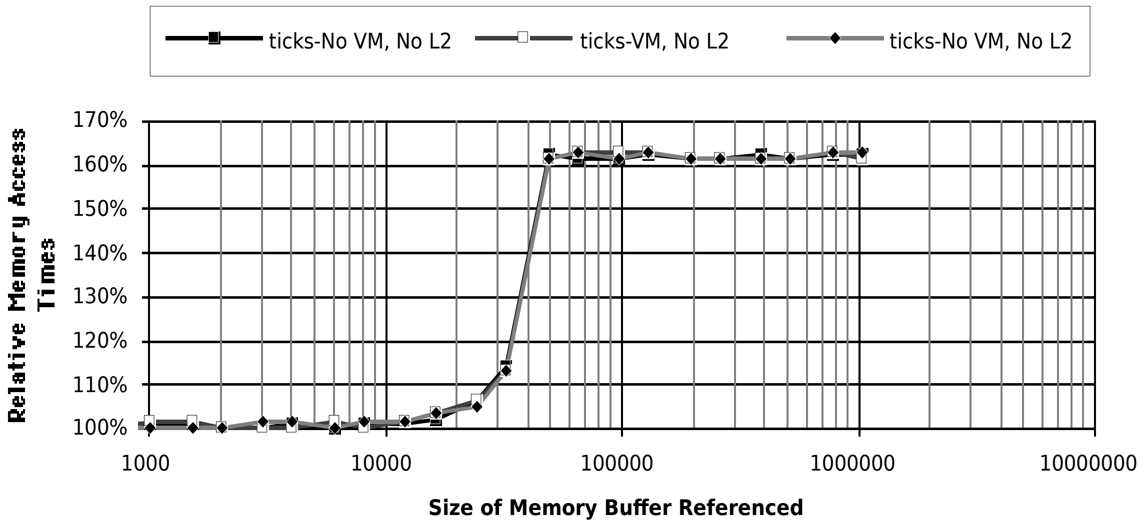
- PowerPC 6100, 8 MB RAM
- 1 MB L2
- VM via Apple
- FAX: (404)729-7132



Matthew's data shows that VM on a limited RAM system can suffer significant performance problems, 5-25%(+), that all but mask comparative L2 performance speedup. The split in L1 performance with and without virtual memory strongly suggests execution of a non-native extension. Finally, L2 is about 10-13% slower than L1 and main memory is about 30% slower.

David B. Chorlian, davidc@panix.com, ran the third set of benchmarks on the following system:

- PowerPC 6100/82, 8 MB RAM
- No L2
- VM via Apple
- Metrowerks Code Warrior (V1.1)
- FAX: (718)270-4081



David's data shows the worst performance slow-down between L1 cache and main memory, 60%. The others had a 25-30% slow-down between L1 and main-memory. Also, Alan and David's data indicate that virtual and real memory operations can have identical performance in the L1 and main-memory ranges. This suggest a large translation buffer and no significant page-fault handling.

PowerPCs are sensitive to non-native extensions. This effect is magnified on an 8 MB system and can wash-out L2 cache performance. Finally, for our benchmark, memory shifts, L2 cache gives at least a 15-20% performance improvement over main memory alone. In fact, software tuning by the programmer, the clustering of code and data, will often give significant performance improvements as L1 cache is engaged.

About the authors:

We met via the Internet in response to a question about cache on PowerPCs. Bob had just finished benchmarking some SGI systems but had no PowerPC. Since Alan, Matthew, and David had PowerPCs, we shared the code and data. Bob wrote the report. We are:

Bob Wilson

Always recommend PCs and Windows -- to your competition
robert.j.wilson@msfc.nasa.gov

```
+Alan Dail - Developer--+--/-\--+-"The journey is the reward" - S.Jobs--+
|804/867-7202           |  \_/  | "The best way to predict the future   |
|AppleLink: AlanDail    | j-+-{ |         is to invent it" - Alan Kay   |
|Internet adail@infi.net|  -|-  | "Hate is not a family value" - Anon   |
+-----------------------+---V---+"Race,in the space I mark Human"-Prince+
```

My L2 cache is made by kingsington and appears to be identical to what Apple supplies (it includes the apple part number).

Matt Brandt
Motorola Semiconductor
330 Research Court
Suite 200
Norcross, GA 30092
matt@atl.sps.mot.com

The manufacturer of my cache card is:
L2 Systems
L2@atlanta.com
(404) 641-9650

David B. Chorlian
Neurodynamics Lab SUNY/HSCB
chorlian@sp1p.neurodyn.hscbklyn.edu
davidc@panix.com