p690 Architecture
Overview

- Power4 CPUs
- Caches
- Memory
- Prefetching
IBM Power4 CPU

• Basic features
  – 1.3 GHz clock speed
  – two independent floating point units
  – single instruction for floating point multiply-add (FMA)
  – theoretical peak is therefore 5.2 GFlops per CPU
• Many typical features of modern RISC processors
• Difficult to attain high percentage of peak performance
  – dense linear algebra is an exception
  – “good” applications realise 10-20% of peak
    • easy to get much less than this!
Power4 processors

• Superscalar processor
  - capable of issuing up to 5 instructions per clock cycle
  - 2FP, 2 integer, 2 load/store, 1 branch, 1 logical
• Two integer addition/logical units
• Two floating point units
  - Single instruction for multiply-add
  - Non-pipelined divide and square root
• 80 integer, 72 FP registers
  - only 32 virtual registers in the instruction set
  - hardware maps virtual registers to physical ones on the fly.
Other features

• Long pipeline
  - up to 20 cycles for each instruction from start to finish
  - FMA takes 6 cycles from reading registers to delivering result back to registers
  - not enough virtual registers to keep both FPUs busy all the time
    • not even Linpack approaches 100% of peak

• Out-of-order execution
  - hardware can reorder instructions to make best use of the hardware resources
  - requires a great deal of internal bookkeeping!
• **Branch prediction**
  - lots of hardware to try and predict branches
  - mispredicted branches cause pipeline to stall
  - 16 Kbit local and global branch predictor tables
  - overkill for scientific codes
    - most branches are back to the start of a loop

• **Speculative execution**
  - can issue instructions ahead of branches
  - instructions are killed if they are not required
  - keeps pipeline full
Caches

- Caches rely on temporal and spatial locality
- Caches are divided into lines (a.k.a. blocks)
- Lines are organized as sets
- A memory location is mapped to a set depending on its address
- It can occupy any line within that set

<table>
<thead>
<tr>
<th>Word address</th>
<th>tag</th>
<th>set index</th>
<th>line offset</th>
</tr>
</thead>
</table>
Cache terminology

- A cache with 1 line per set is called direct mapped
- A cache with k lines per set is called k-way set associative
- A cache with only 1 set is called fully associative
Replacement policy

- When a line is loaded into the cache, its address determines which set it goes into.
- In a direct mapped cache, it simply replaces the only line in the set.
- In a k-way set associative cache, there are k lines which could be ejected to make room for the new one:
  - usual policy is to replace the least recently used (LRU)
  - better than random, but not always optimal
  - LRU line may still be the one required next!
Caches and writes

- **Caches may be:**
  - write-through
    - data written to cache line and to lower memory level
  - write-back
    - data is only written to the cache. Lower levels updated when cache line is replaced

- **Caches may also be:**
  - write allocate
    - if write location in not in cache the enclosing line is loaded into the cache (usual for write-back)
  - no write allocate
    - if write location is not in memory only the underlying level is modified (usual for write-through)
p690 Memory System

- p690 has 3 levels of cache
  - separate L1 data and instruction caches
  - unified L2 shared between 2 CPUs on a chip
  - global L3 cache (more of a memory buffer)
L1 caches

- **Instruction cache**
  - 64Kbytes, direct mapped
  - 128 byte lines

- **Data cache**
  - 32Kbytes
  - 2-way set associative
  - LRU replacement
  - 128-byte lines
    - write-through, no write allocate
  - 2x8-byte reads and 1x8-byte write per cycle.
  - 4-5 cycle latency.
Power4 Chip

- A single chip comprises
  - 2 independent CPUs
  - shared L2 cache
L2 cache

- 1440 Kb Unified (data+instructions)
- 8-way set associative
- Shared by both CPUs on the chip.
  - effectively each processor has 720 Kb of cache
- 128-byte lines
  - write-through, write allocate
  - loads in 32-byte chunks.
- 14-20 cycle latency L2 → registers
- Cache has 3 independent sections of 480 Kb
  - Lines within the 1440 Kb unit are hashed to sections
    (consecutive lines never go to the same section).
Power4 Chip
Multi-Chip Modules

• Chips are packaged up in groups of four
  - each Multi-Chip Module (MCM) has eight CPUs
  - all sharing the same L3 cache
L3 cache

- Really a memory buffer rather than a cache.
- 128Mb per MCM (4 chips, 8CPU)
- 8-way set-associative
- 512 bytes lines
- approx. 100 cycle latency
- Usually only caches memory locations attached to the MCM
- Shared by all CPUs
  - single CPU jobs get access to ALL the L3 cache in the system.
- Does not allocate if already “busy”
Main memory

• 8 Gbytes of main memory per MCM
  - 1 Gbyte per processor
• Accessible by all CPUs
• 350–400 cycles latency from main memory to registers
• Running one CPU on an MCM, a memory bandwidth of around 2.5 Gbyte/s is observed.
• However, when running all 8 CPUs the aggregate bandwidth is around 8 Gbyte/s
  - poor scaling, or good single CPU performance?
  - beware of single CPU benchmarking
TLB

- Translation lookaside buffer
  - processor works on effective addresses
  - memory works on real addresses
  - TLB is a cache for the effective->real mapping

- 1024 entries, 4 way set associative
  - each entry corresponds to a page (4 Kbytes)
  - whole TLB addresses 4 Mbytes
  - larger than L2 cache
Larger Shared-Memory Nodes

- Four MCMs make up a p690 frame
  - also called Regatta H
  - 32 CPUs and 32 Gb memory per frame
  - peak of 166.4 Gflops

- Each frame configured as 4 machines
  - called Logical PARtitions
  - each LPAR maps to one MCM
LPARs are almost completely independent
- run separate operating systems
- cannot access memory on a different LPAR

The 4 MCMs in a frames are connected by multiple busses
- some cross-LPAR traffic does occur
- cache coherency mechanisms cannot be turned off

Single LPAR performance can be impacted by jobs running on other LPARs in the same frame
- can be on the order of 10% in worst case
- not drastic, but noticeable on some benchmarks.
• p690 has a hardware prefetch capability
  - helps to hide the long latencies
  - make use of the available memory bandwidth
• Simple algorithm for guessing which cache lines will be required in the near future
  - fetch them before they are requested
• Prefetch engine monitors loads to cache lines
  - detects accesses to consecutive cache lines (128b)
    • either ascending or descending order in memory
  - two consecutive accesses trigger a prefetch stream
• Accesses to subsequent consecutive cache lines cause data to be fetched into the different caches
  - next line in sequence is fetched to L1 cache
  - line 5 ahead is fetched into L2 cache
  - lines 17, 18, 19 & 20 ahead (512 bytes) are fetched into L3 cache.
• Distance ahead is long enough to hide the memory latency
• Up to 8 streams can be active at the same time
• Stream stops when page boundary is crossed
  - every 4 Kbytes, unless large pages enabled
Where to find out more

• The Power4 Processor Introduction and Tuning Guide