**Paper review of**

**Amdahl’s Law in the Multicore Era**

As we enter the multicore era, we have reached a point where computing vendors are able to double the number of cores on a chip and continue doing that regularly. Designers now have to think in terms of multicore systems where number of cores, the type of pipelining involved and the architecture for each core as well as power management for both dynamic and static power devices become deciding factors. When the number of cores multiple further it becomes tougher to keep track of above factors.

This paper provides a corollary to the Amdahl’s Software Model to encourage designers to view the chip’s performance rather than concentrate on core efficiencies.

**A corollary for multicore chip cost:**

Assume, a multicore chip of a give size and a technology generation has *n* *base core equivalents*, where 1 BCE implements the baseline core. Also the non-processor resources are assumed to remain roughly constant over multicore variations. Let’s also assume that there are techniques that can account for better sequential performance when multiple BCEs are involved. If the core built using *m* BCEs can perform better than using *m* BCE individually, then increasing cores enhances both sequential and parallel execution. Otherwise, increasing no. of cores results in poor parallel execution which can affect processor speeds. Let the performance be defined by a function *perf(m)*.

**Amdahl’s law:** If a fraction *f* of a computation can be sped up by *S*, the overall speedup is given by:

Speedupenhanced (*f, S*) = Speedupparallel (*f, n*) =

* The amount of parallelism depends on the fraction *(f/n)*. If n starts to increase, parallelizability starts to decrease.
* Amdahl’s law assumes that when running on an enhanced machine the portion that can be parallelizable remains same.

The above assumption seems unfair because when the size of the computational problem increases, the size of the dataset increases and on an enhanced machine, the computational time doesn’t necessarily increase proportionally with the input size.

By analyzing the performance of *m* cores, it can be concluded that *perf(m)* = . So if no. of cores increase by 9, performance triples.

**Symmetric multicore chips:**

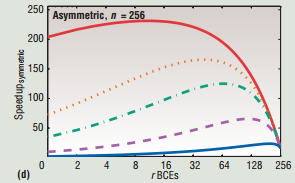
A symmetric multicore chip requires that all its cores perform evenly. If the chip has a resource budget of *n* BCEs, if can support *n/r* cores of r BCEs each. A symmetric multicore chip uses 1 core to perform at *perf(r)* for sequential execution and uses n/r cores with performance of *perf(r)\*n/r* for parallel execution. The overall speedup we get is:

Speedupsymmetric (*f, n, r*) =

The performance is observed to vary as *perf(r)* = where *r* is the number of BCEs. That result is obtained from the graph.

1. It can be observed from the graph that keeping *f* close to 1 gives best speedup.
2. Increasing the number of BCE cores can also be helpful as the maximum speedup can occur at any level of cores.
3. Moving to denser chips increases the likelihood that cores will be non-minimal. Even at f = 0.99, minimal base cores are optimal at chip size n = 16, but more powerful cores help at n = 256.

**Asymmetric multicore chips:**

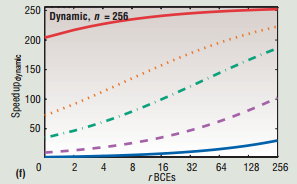
Instead of using all cores of same capability, we use a one larger core that performs better. It use up more resources – *r* out of *n*. The remaining 1-BCE cores utilize the *n-r* resources. This chip uses one larger core to perform sequentially with *perf(r)*. For parallel fraction, the larger core contributes *perf(r)* and (n-r) BCE cores contribute (n-r). Overall speedup is:

Speedupasymmetric (*f, n, r*) =

The symmetric curves typically show either immediate performance improvement or performance loss as the chip uses more powerful cores, depending on the level of parallelism.

1. Asymmetric core can offer much better speedups than symmetric cores. Working on building asymmetric multicore chips along with taking scheduling and overhead into consideration can result in better performing chips than symmetric multicore chips.
2. Denser multicore chips increase both the speedup benefit of going asymmetric and the optimal performance of the single large core.

**Dynamic multicore chips:**

This procedure combines dynamically *r* cores to boost only the sequential performance and obtaining the parallel performance from the *n* cores. It offers a sequential performance of *perf(r)*. The overall performance then becomes:

Speedupdynamic (*f, n, r*) =

1. Dynamic multicore chips can perform better than asymmetric chips but considering Amdahl’s assumption they need more cores for sequential performance than that is possible today.

**Simple as possible, but no simpler:**

* With the current technology and research available, building cores that can give performance as per the need without trading off for performance and hardware resource overhead looks improbable.
* In addition, we are constrained in terms of dynamic and static power a chip uses. Also, the on and off-chip memory and interconnect designs add significant overhead.
* The software tasks and data movements add overhead. Scheduling adds overhead to asymmetric and dynamic multicore chips.
* The Amdahl’s law can undergo changes or become inapplicable for future systems. On the other hand, if the future technology allows accumulating more cores on the chip, greater parallelism can be implemented.