Exploring Performance and Power Scaling in Multi-Core Processors

Jonathan Dorn, Robbie Hott, Michelle McDaniel

May 13, 2011

Abstract

The demand for lower power over higher performance processors has increased over the past few years, leading to multi-core architectures. The current trend being favoring few powerful cores over a larger number of simpler, smaller cores. We address the tradeoff between these two design decisions by considering Pollack's Rule for single cores and applying it with Amdahl's law to multi-core architectures. We use McPAT to obtain power and area data for cores with varying pipeline widths, then gather performance data of the cores from the M5 simulator and SPEC2006 benchmarks. For all configurations tested, given either a power or area budget, a processor consisting of a moderate number of medium-sized cores gives the best overall performance-per-watt.

1 Introduction

Recently, performance has become less of a leading market demand. Now more than ever, the demand for lower power has started to outweigh the demand for increased performance at the expense of power. In the personal computing market, laptops, smartphones, and tablet devices have spurred the desire for longer battery life at the expense of performance. Likewise, with the rise in cloud computing, computation has shifted to data centers, which have strict power limits to meet.

In order to achieve higher performance while consuming less power, there has been a shift towards multi-core computing. The current general-purpose processing trend is to include a few powerful cores onto a single die. However, more processors are emerging with up to 100 simpler cores on a single chip, such as the RAW–now Tilera–chip [14]. Tilera promises hundreds of cores per rack with the same power budget as only a few dozen current general-purpose cores. These new architectures raise an important tradeoff question: in terms of cost, performance, and power efficiency, which is better, more cores with lower complexity or fewer more complex cores?

To discuss this tradeoff, we consider Pollack's Rule [3]: "performance increases roughly proportional to the square root of the increase in complexity." We link processing capabilities–complexity–with core area to extend Pollack's Rule to multi-core architectures with homogeneous cores. We utilize McPAT [11] to collect power and area data for various core choices. Similarly, using the M5 simulator [2], we collect performance data for each of these choices to guide our analytic model.

The major contributions of this paper are:

- An identification of the tradeoffs between power, performance, complexity and number of cores
- An analytic model we use to extend Pollack's Rule to multi-core architectures
- And the introduction of a variant of Pollack's Rule for pipeline width.

The remainder of this document is structured as follows. We first discuss related work in Section 2. In Section 3, we present our analytic model that will be used to extend Pollack's Rule to multi-core architectures. We present our methodology in Section 4, including a summary of the parameters we will use as descriptors of the various core types, our benchmarks, and our McPAT constraints. In Section 5, we summarize the results of our simulation, including a discussion of both area versus power and area versus performance, and a variant of Pollack's Rule based on our experimentation. Finally, we present future work and conclude in Section 6.

2 Related Work

Recent research projects have begun to look at the design space for multi-core architectures. Esmaeilzadeh *et al.* use Pareto frontiers to model single-core performance/power and performance/area trade-offs at different technology nodes [6]. They assume that $Perf \propto \sqrt{Area}$ and fit Pollack's Rule to real-world processors (three Intel processors and one AMD processor) to build a model for performance/area, mapping the curve to the maximum performance at each area node. They model multi-core performance using Amdahl's Law as an upper bound for performance and develop a more realistic model that takes into account changing microarchitectural features. They find that multi-core scaling is power limited, and that with smaller technology nodes, increasingly more portions of the chip must be powered off, decreasing the amount of speedup that can actually be achieved as the technology node shrinks.

Huh *et al.* compare the area and performance trade-offs for CMP organizations in order to advocate how many cores future CMPs should have and the types of cores that should be used [8]. They advocate for fewer out-of-order cores (fewer larger cores), rather than a larger number of in-order cores (more smaller cores) to achieve maximal throughput. Contrastingly, Davis *et al.* find that a large number of smaller cores with lower performance achieves the best overall CMP performance, compared to a smaller number of larger cores [9]. Lee *et al.* developed a performance model to predict multi-core performance for a given area budget based on Amdahl's law [10]. They additionally create a model of a resource-constrained system to more realistically model multi-core performance. They find that, in a resource-constrained system, the highest speedup can be achieved with a smaller number of medium to large sized cores.

Finally, while several studies assume Pollack's Rule [4][5][7][12][15], we seek to actually examine it. We find that the general form of the rule that these studies assume ($Perf \propto \sqrt{Area}$) is not accurate and propose a variation that more accurately models our simulated results.

3 Analytical Modeling

In order to extend Pollack's Rule to multi-core architectures, we use Amdahl's Law for parallelism. Amdahl's Law [1] states that multi-core performance is related to the single-core performance based on the amount of parallelism, according to

$$Perf_{mc} = \frac{Perf_{sc}}{(1-\rho)+\frac{\rho}{N}}, \tag{1}$$

where N is the number of cores and ρ is the parallelizable fraction of the workload. For our initial model, we assume Pollack's Rule holds true for single-core performance; that is, we assume $Perf_{sc} \propto \sqrt{A_{sc}}$, or equivalently,

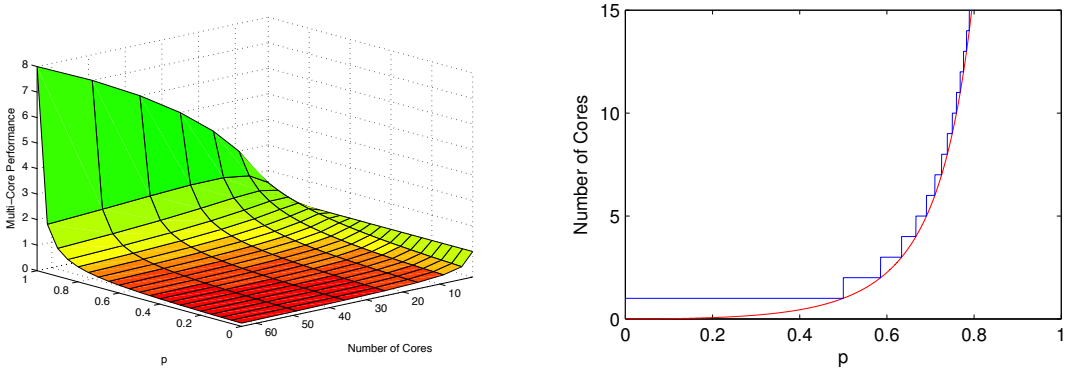
$$Perf_{sc} = c\sqrt{A_{sc}} \tag{2}$$

and therefore rewrite (1) as

$$Perf_{mc} = \frac{c\sqrt{A_{sc}}}{(1-\rho)+\frac{\rho}{N}}.$$

Assuming that $A_{die} \approx NA_{sc}$, we can further simplify the multi-core performance equation:

$$\begin{aligned} Perf_{mc} &= \frac{c\sqrt{A_{sc}}}{(1-\rho)+\frac{\rho}{N}} \\ &= \frac{cN\sqrt{A_{sc}}}{N(1-\rho)+\rho} \\ &\approx \frac{c\sqrt{N}\sqrt{A_{die}}}{N(1-\rho)+\rho} \end{aligned}$$



(a) (b)

Figure 1: Visualizations of the (a) analytic model for multi-core performance over differing numbers of cores in a fixed die area and (b) number of cores for which multi-core performance matches the performance of a single core occupying the entire die.

Rearranging to emphasize the relationship between $Perf_{mc}$ and $\sqrt{A_{die}}$, we get

$$Perf_{mc} = \frac{\sqrt{N}}{N(1-\rho)+\rho} c\sqrt{A_{die}}. \tag{3}$$

Thus, for fixed values of N and ρ, we expect the multi-core performance to be proportional to the square root of the area. Therefore, assuming Pollack's Rule for single cores, it is also valid for multi-core architectures.

In the common problem of selecting a design within a fixed area budget, this equation also offers insight into the tradeoffs between performance, complexity, and the number of cores. Figure 1(a) shows the tradeoff of varying both N and ρ with a fixed die area. As expected, for workloads with low parallelism (i.e. small values of ρ) fewer larger cores give better performance while more smaller cores perform better with highly parallel (large ρ) workloads.

Solving the equation

$$\frac{\sqrt{N}}{N(1-\rho)+\rho} = 1,$$

for N yields two solutions,

$$N = 1$$

or

$$N = \frac{\rho^2}{\rho^2 - 2\rho + 1}.$$

The first characterizes a single-core processor, for which we assumed performance would be proportional to the square root of the area. The second describes multi-core machines, where the number of cores needed to attain performance equivalent to a single core that occupies the entire die depends on the level (ρ) of parallelism. Figure 1(b) charts this tradeoff.

4 Methodology

In order to evaluate Pollack's Rule and obtain performance, power, and area results for varying processor sizes, we consider various configurations of the Alpha core. Table 1 shows the different configurations we consider. In section 5.1, we discuss the similarities in area and power between in-order and out-of-order cores, as well as in various other parameters, which led to these configuration choices.

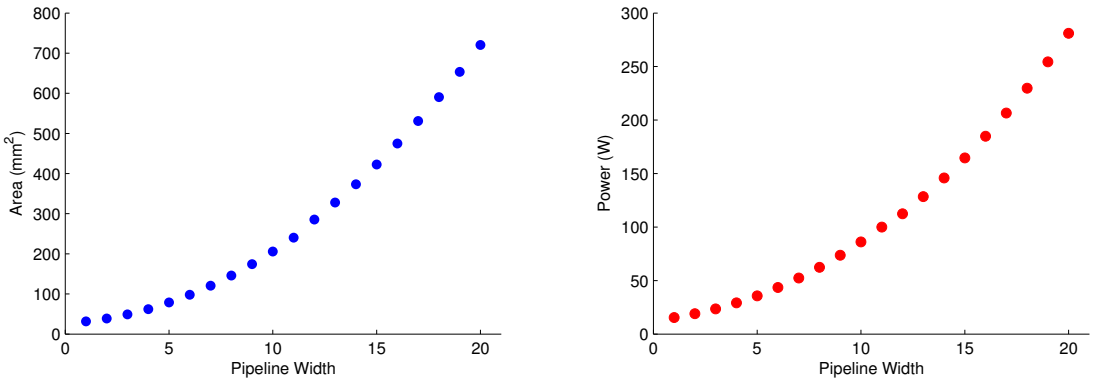
The basis of each experimental configuration consists of the constant parameter settings listed in Table 1 and the baseline settings of all variable parameters. To generate the experimental configurations from this

Variable Parameter	Range of Values	Baseline Value
Core Type	out-of-order and in-order	-
Frequency	1 - 4 GHz	-
Pipeline Width	1 - 8	4
ROB Entries	32 - 256	192
Local Predictor Size	1024 - 4096	2048
Global Predictor Size	4096 - 16392	8192
Load/Store Buffer Entries	16 - 64	32
Constant Parameter	**Value**	
ARF Registers	32 INT / 32 FP	
Physical Registers	256 INT / 256 FP	
ITLB/DTLB	128 entries	
L1 Cache	32 KB I / 32 KB D; 2-way associative	
L2 Cache	2 MB; 8-way associative	

Table 1: Simulated configurations.

Benchmark	FP or INT	MEM or CPU
gobmk	INT	CPU
perlbench	INT	CPU
libquantum	INT	MEM
omnetpp	INT	MEM
calculix	FP	CPU
povray	FP	CPU
tonto	FP	CPU
soplex	FP	MEM

Table 2: Benchmarks measured.



(a) (b)

Figure 2: McPAT comparison of varying pipeline widths vs (a) total core area and (b) peak power.

basis, we vary the frequency and one of the variable parameters together; each of these drives a simulation of an out-of-order processor as well as (when applicable) an in-order processor. This restriction helps us to avoid a combinatorial explosion in the number of configurations to explore through simulation.

4.1 McPAT

For each of our configurations, we use McPAT to determine the area and power requirements of the core. Since McPAT was used to find configuration points for the Alpha core, we ran the sample workload consisting of 400,000 total instructions: half are integer instructions, while floating point and branch instructions comprise one-quarter each. From the McPAT data, we utilize total processor area for multi-core designs, total core area, and peak power for the chip.

For initial results, we ran McPAT on a sample Alpha processor of varying pipeline widths. Our sample processor is simulated at 45nm for pipeline widths varying from 1 wide to 20 wide, for both the integer and floating point pipelines. Figure 2 shows that as the width of the pipeline increases, both power and area increase, but not linearly as originally expected. However, as pipeline width increases, the change in area between each design increases, since power is linear with respect to area.

4.2 M5 Simulation

To measure the performance of each core configuration, we run a subset of the SPEC2006 benchmarks [13] using the M5 simulator. Table 2 lists the benchmarks used. In order to characterize the performance of the chips based on different workloads and to gain insight into the differences in performance for these different workloads, we have separated these benchmarks into four categories—INT-CPU, INT-MEM, FP-CPU, and FP-MEM—based on whether they are floating point or integer benchmarks, CPU or memory intensive. All performance measurements use the average IPC reported by M5 for the entire run.

4.3 Multicore Modeling

Using the results from M5 and McPAT to gain performance and area results for individual cores and assuming Amdahl's law for parallelism, we augment our analytical models of multi-core performance with simulation results. We want to see if, using our models, we can determine which processor configuration at a given area level is best. To do so, we consider the number of cores, performance, and power of each configuration at a given area overhead. Whichever configuration (i.e. number of cores for the area) gives us the best performance per watt of power, we consider "optimal" at that area. This allows us to discover, when considering both power and performance on relatively equal importance, what configuration is best: many small cores or fewer larger cores. Of course, this also depends on the amount of parallelism in the applications that are running, since the amount of parallelism determines the performance. We envision a graph similar to Figure 1(a) with fraction of parallelism and number of cores as our independent variables and performance/power as our dependent variable.

4.4 Pollack's Rule

Equation 3 indicates that, assuming that parallel performance is adequately modeled by Amdahl's Law, it is sufficient to consider fixed values for N and ρ when relating multi-core performance to Pollack's Rule. Specifically, the fixed values $N = 1$ and $\rho = 0$ characterize applications on a single-core processor. That is, if Pollack's Rule is valid for single-core, it follows that the rule also holds for multi-core. Likewise, if a variant of Pollack's Rule is valid for single-core, it will also follow for multi-core as well.

5 Experimental Results

In this section, we present the experimental results of our simulation study. We first summarize the relationship between area and power for single-core and multi-core configurations. We then consider the relationship between performance and area for single and multi-core and conclude that the generally accepted form of Pollack's Rule that $Perf \propto \sqrt{Area}$ is not an accurate representation of the true relationship between area and performance. Instead, we find that $Perf \propto \Delta\sqrt{Area}$, where $\Delta\sqrt{Area}$ is directly related to change in pipeline width. We conclude the section with a discussion of the relationship between area and performance-per-watt for a given area budget and a given power budget, and conclude that for a realistic amount of parallelism, a configuration made up of medium sized cores is optimal for achieving the highest performance-per-watt. Because of inconsistent results for in-order cores, we focus our discussion on the results for out-of-order core configurations.

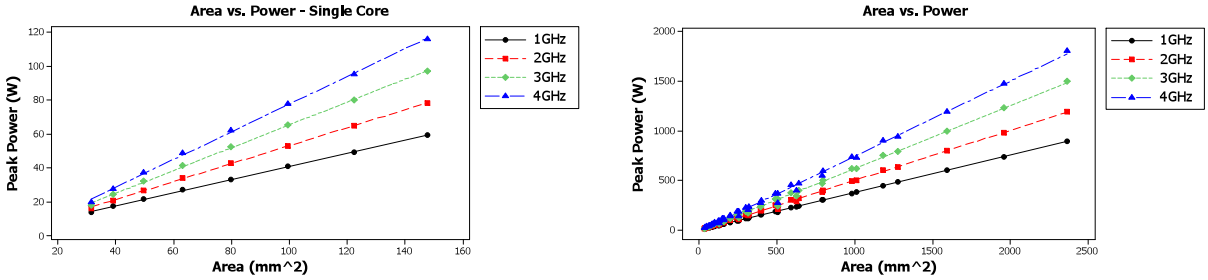
5.1 Area vs. Power

McPAT was used to ensure diversity in the area and power requirements of the configurations under test for the 45nm technology node. The entire design space, consisting of all possible combinations of parameters from Table 1, was explored. All of the combinations, except for varying pipeline width, had an insignificant impact on the total area and power requirements of the core, as compared with changing base-line pipeline width. The ranges in area and power for both in-order and out-of-order cores are shown in Table 3. Because of the low variance in area and power, we focused mainly on pipeline width to enlarge cores. Likewise, since in-order cores provide less performance for approximately the same core area, we use the higher-performing out-of-order cores as our baseline for power and performance.

In-Order Configurations		
Width	Core Area (mm^2)	Power (W)
1	28.40 - 29.25	12.67 - 13.12
2	35.10 - 36.09	15.85 - 16.42
3	44.68 - 45.87	19.87 - 20.58
4	57.00 - 58.45	24.71 - 25.60
5	72.07 - 74.53	30.36 - 31.49
6	89.91 - 93.00	36.83 - 38.22
7	110.56 - 114.35	44.12 - 45.81
8	133.95 - 138.54	52.21 - 54.25

Out-of-Order Configurations		
Width	Core Area (mm^2)	Power (W)
1	31.32 - 32.32	14.09 - 14.68
2	38.64 - 39.88	17.43 - 18.17
3	49.07 - 50.61	21.74 - 22.65
4	62.56 - 64.50	27.06 - 28.18
5	79.08 - 82.20	33.41 - 34.80
6	98.67 - 102.63	40.83 - 42.51
7	121.34 - 126.26	49.35 - 51.38
8	146.06 - 153.06	58.98 - 61.45

Table 3: Ranges in area and power for all single-core in-order (left) and out-of-order (right) configurations for each pipeline width.



(a) (b)

Figure 3: Core area vs. power scaling for (a) single core and (b) all core configurations (multi-core and single-core).

Figure 3(a) shows that power scales linearly with respect to area for each of the frequencies used. Note that area changes as a factor of pipeline width: each point represents a core of a one-size-wider configuration. For each width increase, the difference in areas also increases, therefore making Figure 2 linear with area as opposed to super-linear with pipeline width. Including multi-core configurations, Figure 3(b), power still scales linearly with area, even when the area includes not only wider cores, but also more copies of each core per die. For example, 8 copies of a 6-wide core will consume roughly the same area and power as 16 copies of a 3-wide core. This linear property gives us the freedom to easily consider different design configurations with similar power usage and area requirements, but with varying core performance levels. Figure 4 gives individual views for each of the configurations in a multi-core design, from 2 to 16 cores.

5.2 Area vs. Performance

Similar to the area and power numbers computed with McPAT, the ranges in measured IPC at each fixed pipeline width were significantly smaller than the difference in performance due to increasing the width by one. Thus, for simplicity, the graphs show only the performance numbers for the baseline configuration at each pipeline width.

In Figure 5, we plot the IPC at each pipeline width against the square root of the core area. Note that the relationship between the square root of the area and the performance is not linear, as would be expected from Pollack's Rule. This tapering in performance improvement is not entirely surprising, since each successive increase in pipeline width allows the processor to exploit successively less additional instruction-level parallelism in the benchmark workloads. We expect that, for each workload, there is some pipeline width beyond which no further IPC improvement is possible. However, that width may not be practically achievable, due to other design constraints such as increasing wire delays and logic complexity to manage

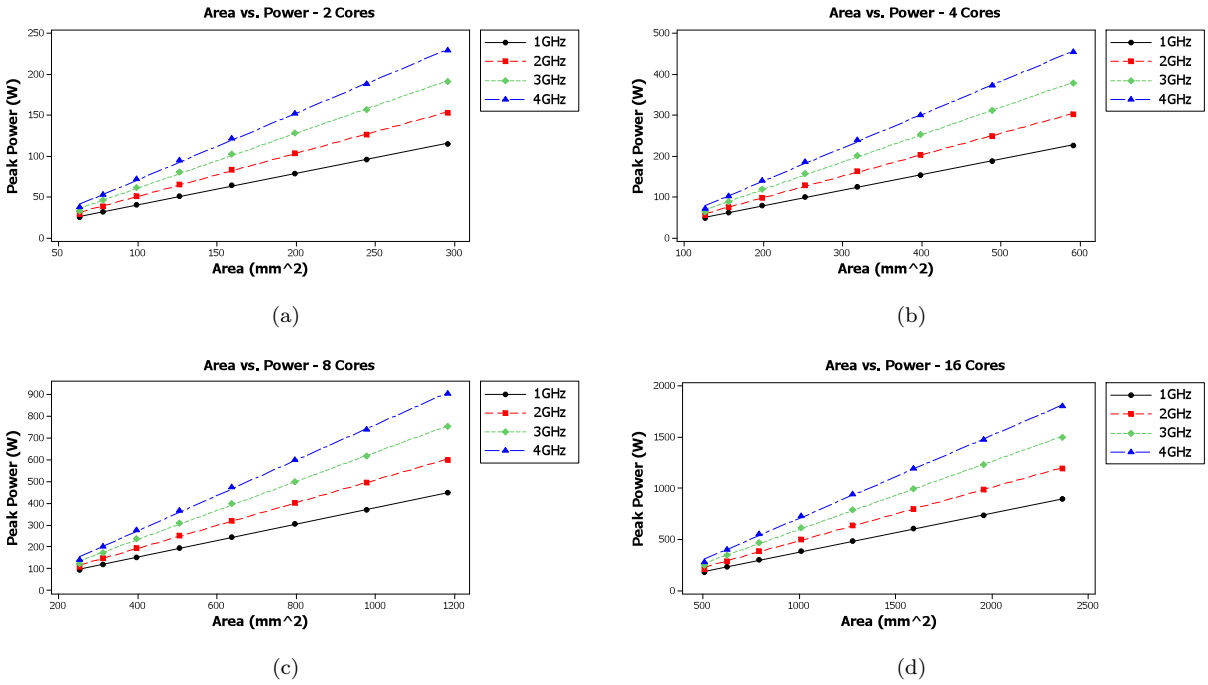


Figure 4: multi-core area vs. power scaling for (a) 2, (b) 4, (c) 8 and (d) 16 cores.

such a wide pipeline.

In Figure 6, instead of using the square root of the area, we plot the increase in the square root of area due to increasing pipeline width. This relationship is much more linear, as can easily be seen in the trend lines. As discussed above, there is still a theoretical maximum IPC that can be achieved by increasing the pipeline width. However, as the IPC reaches this theoretical maximum, the difference in square roots of area also approaches a stable value. Therefore, for realistic widths, we can expect this linear relationship to hold.

Thus, instead of equation (2) above, we have a different relationship between performance and power,

$$Perf_{sc,i} = c\left(\sqrt{A_{sc,i}} - \sqrt{A_{sc,i-1}}\right) \tag{4}$$

where $Perf_{sc,i}$ and $A_{sc,i}$ refer to the performance and area of a processor with an i-wide pipeline. The same derivation that gave us equation (3) now produces this equation for multi-core performance (the mc subscripts have been removed for clarity):

$$Perf_i = \frac{\sqrt{N}}{N(1-\rho)+\rho} c\left(\sqrt{A_i} - \sqrt{A_{i-1}}\right). \tag{5}$$

5.3 Area vs. Performance/Power

In a multi-core setting, we cannot simply look at performance and power separately. A configuration that achieves extremely high performance due to a high amount of parallelism and a large number of cores may also have extremely high power consumption. Likewise, a configuration with lower power may also have lower performance due to having fewer cores. Therefore, we must consider the amount of performance we can achieve per watt of power in order to choose the configuration that optimizes for both.

To consider which of the configurations is optimal for performance-per-watt, we constructed a linear regression for each of our core configurations. We used McPAT to calculate the peak power for single-core, 2, 4, 8, and 16 cores for each of the widths of our cores. As shown in Figure 7, for all of the configurations,

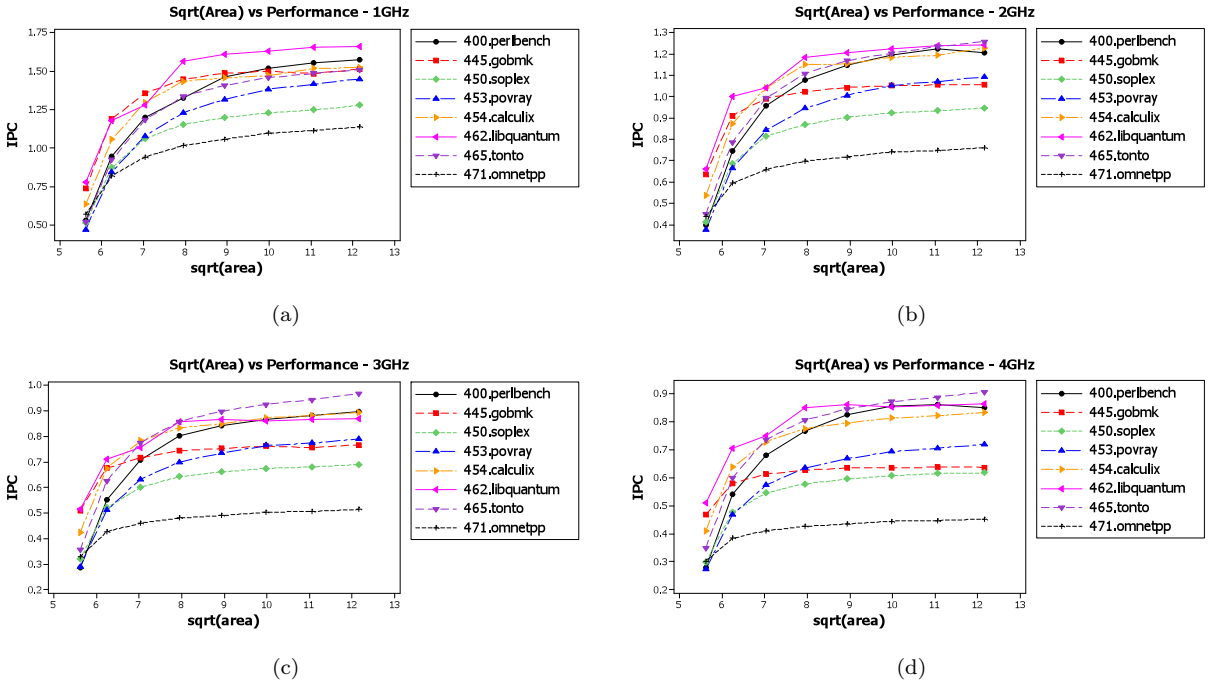


Figure 5: Single-core square root of the area vs. performance (IPC) for (a) 1GHz, (b) 2GHz, (c) 3GHz and (d) 4GHz.

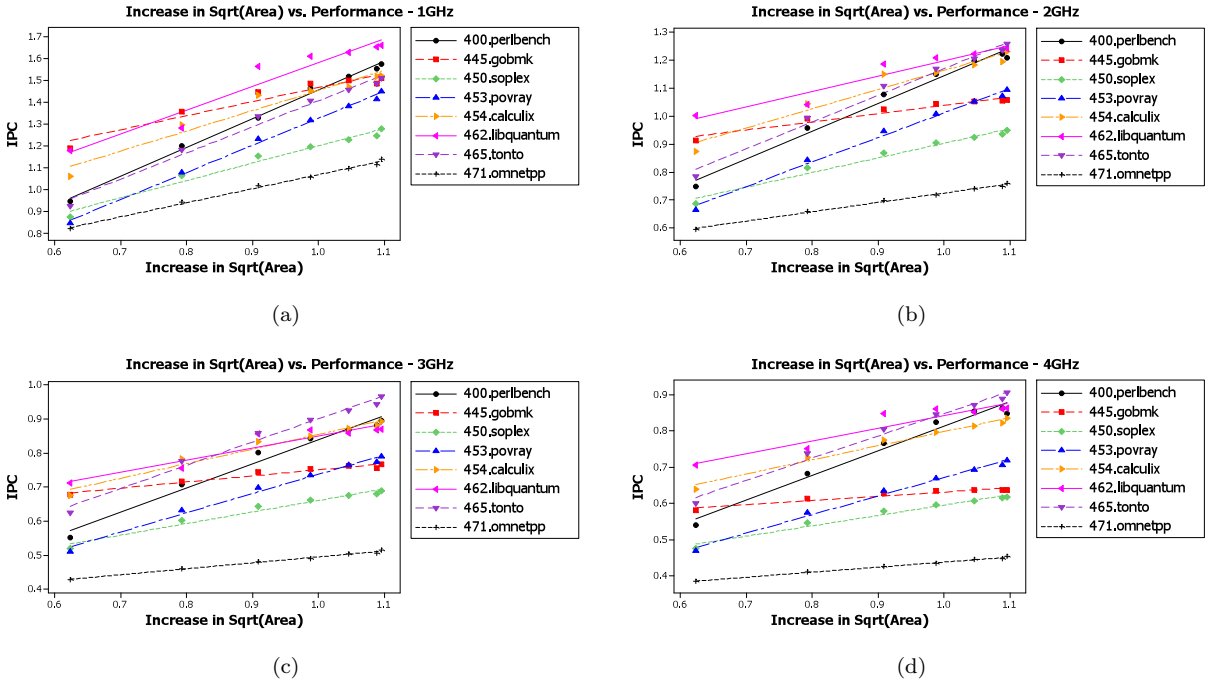


Figure 6: Single-core increase in square root of the area (core complexity) vs. performance (IPC) for (a) 1GHz, (b) 2GHz, (c) 3GHz and (d) 4GHz.

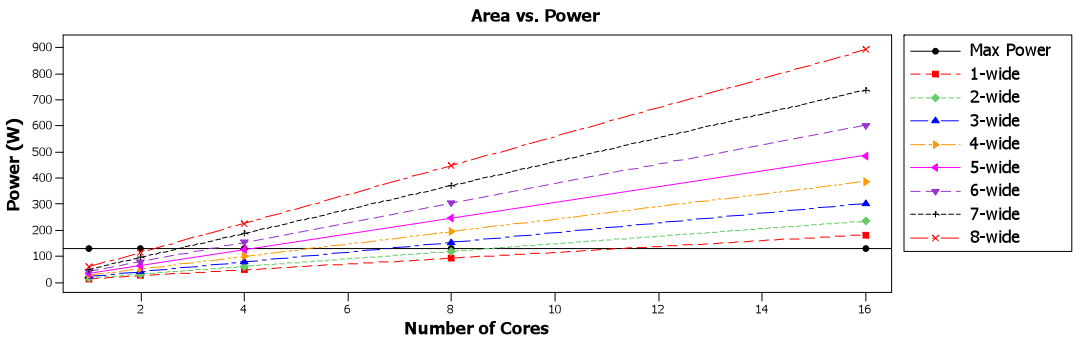


Figure 7: Number of cores vs. power for varying pipeline widths. As we can see, the relationship is linear for all pipeline widths, so we can use the data to predict the power for varying number of cores at different pipeline widths. Also shown is where our power budget of 130W cuts off each of the configurations.

the relationship between number of cores and peak power is linear. Therefore, we can apply a regression to each configuration in order to estimate the peak power for varying numbers of cores. Using the single-core performance data extracted from M5 and Amdahl's law, we calculated the performance of the multi-core configurations.

5.3.1 Fixed Area Budget

For this experiment, we set our area budget to the area needed for 64 1-wide cores (our smallest configuration). The number of cores at each configuration that fit into this area budget is shown in Table 4. For this experiment, we considered three levels of parallelism: no parallelism ($\rho = 0$), perfect parallelism ($\rho = 1$), and a realistic amount of parallelism ($\rho = 0.9$). The results of this experiment are shown in Figure 8.

Figure 8(a) shows that for no parallelism, the highest performance-per-watt of peak power is achieved by having 13 copies of the 8-wide configuration. This is expected, as performance is set to single-core performance, and power is generally equal for all of the configurations. In this case, a more interesting result to consider is the performance-per-watt for *average* power, rather than peak power.

Likewise, the results of perfect parallelism, Figure 8(c), are what we expect. Since the performance of the single-core 2-wide configuration is almost double for most of our benchmarks over the 1-wide core, and the number of cores is still large, we expect the performance of the 2-wide multi-core configuration to be higher than the 1-wide core with perfect parallelism. Additionally, as with the case where there is no parallelism, the power across the eight configurations is generally equal, so the performance-per-watt of peak power is

Area Budget: $2017.971 mm^2$

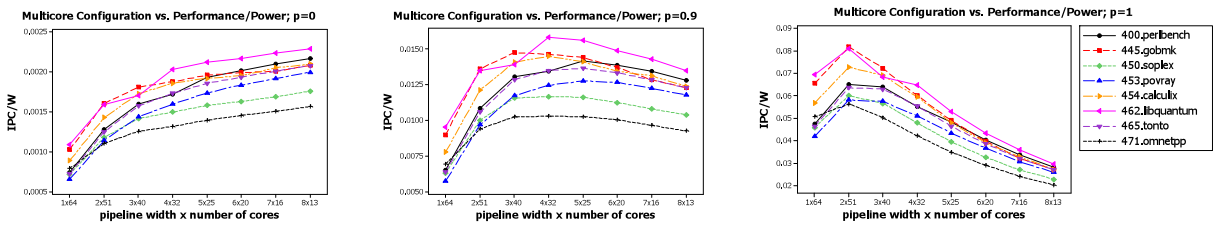
Pipeline Width	Number of Cores
1	64
2	51
3	40
4	32
5	25
6	20
7	16
8	13

Table 4: Number of cores for the multi-core configuration experiments with an area budget of 2017.971 mm^2.

Power Budget: 130W

Pipeline Width	Number of Cores
1	11
2	8
3	6
4	5
5	4
6	3
7	2
8	2

Table 5: Number of cores for the multi-core configuration experiments with a peak power budget of 130W.



(a) (b) (c)

Figure 8: Number of cores of varying single-core size vs. performance-per-watt, with an area budget of 2017.97 mm^2 (64 copies of the 1-wide core) and the amount of parallelism set at (a) 0 (no parallelism), (b) 0.9 and (c) 1 (perfect parallelism). For (a), we achieve the highest performance-per-watt in the 13-core 8-wide setup (fewer larger cores is better). For a realistic amount of parallelism as shown in (b), the 32-core 4-wide configuration is generally the best option. For perfect parallelism (c), the best configuration is 51 cores with 2-wide pipelines.

highest for the 2-wide multi-core configuration. Because all of the cores are being used in this configuration, we believe that peak power is a reasonable metric for this experiment, and that our results would be the same if we considered average power.

Between perfect and no parallelism lies a "reasonable" amount of parallelism that is representative of real parallel programs. For this experiment, we chose to set parallelism to 0.9. The results of this experiment are shown in Figure 8(b). We can see that, in this case, the configuration that balances performance and power best varies across the different benchmarks. For two of the benchmarks (`445.gobmk` and `471.omnetpp`), the optimal configuration is the 3-wide multi-core configuration. `450.soplex`, `454.calculix` and `462.libquantum` all have the best balance between performance and power with the 4-wide configuration. Finally, the 5-wide configuration is best for `400.perlbench`, `453.povray` and `465.tonto`. This experiment would also benefit from considering the average power case, rather than only peak power, since not all of the cores will be active throughout the duration of execution.

5.3.2 Fixed Power Budget

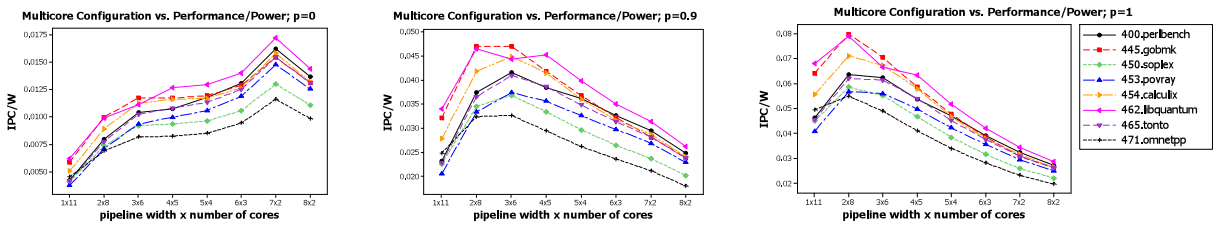
We constructed a similar experiment as above where we set the power budget to 130W and allowed area to be determined by the number of cores achievable within the power budget. The number of cores for each of our core configurations for this experiment are listed in Table 5. Parallelism was again set to 0, 0.9 and 1. The results are shown in Figure 9.

Figure 9(a) shows that for no parallelism, the optimal configuration is the dual-core 7-wide setup. Because the power consumption of three 7-wide cores is just greater than our power budget, we are only allowed two 7-wide cores in this space. Therefore, this configuration uses significantly less power than the other configurations, and the performance-per-watt is higher for this setup than for the others.

Figures 9(b) and 9(c) show the performance-per-watt for parallelism set to 0.9 and 1, respectively. These figures show that, similar to the area budget experiments, for a power budget of 130W, the configurations with the highest performance-per-watt are the 6-core 3-wide configuration for $\rho = 0.9$ and the 8-core 2-wide configuration for $\rho = 1$. This suggests that for both a set area budget and a set power budget, we want to have a larger number of small-to-medium sized cores in order to achieve the highest performance-per-watt.

6 Conclusions and Future Work

In this paper, we presented an analytic model for multi-core performance. We showed that peak power scales linearly with area. Our data did not match or validate the generally accepted form of Pollack's Rule. Rather than performance scaling with the square root of the area, performance scales with the increase in the square root of the area caused by increasing pipeline width. Therefore, we derived Equation 5 for the



(a) (b) (c)

Figure 9: Number of cores of varying single-core size vs. performance-per-watt, with a power budget of 130W and the amount of parallelism set at (a) 0 (no parallelism), (b) 0.9 and (c) 1 (perfect parallelism). For (a), we achieve the highest performance-per-watt in the dual-core 7-wide setup (fewer larger cores is better). For a realistic amount of parallelism as shown in (b), the 6-core 3-wide configuration is generally the best option. For perfect parallelism (c), the best configuration is the 8-core 2-wide pipeline.

performance of a multi-core processor.

$$Perf_i = \frac{\sqrt{N}}{N(1-\rho)+\rho} c \left(\sqrt{A_i} - \sqrt{A_{i-1}} \right),$$

where $Perf_i$ is the multi-core performance of an $i-$wide pipeline core and A_i is the area of the die with core pipeline width i. Finally, we evaluated our configurations in terms of the amount of performance per watt of power. We found that for a realistic amount of parallelism ($\rho = 0.9$), the highest performance-per-watt can be achieved with a configuration consisting of a moderate number of medium-to-small sized cores.

For future work, we would like to look at a greater range of processors to improve our discussion of Pollack's Rule and to provide validation for it. We would like to include a more comprehensive set of in-order versus out-of-order configurations, in addition to considering deeper pipelines and more complicated branch prediction units. Additionally, we would like to be able to include special microarchitectural features such as the Loop Stream Detector found in recent Intel processors, which can significantly improve performance. For our power experiments, we would like to include average power based on the performance results from simulation. Finally, we would like to examine Pollack's Rule using simulated multi-core performance rather than analytically derived performance using Amdahl's Law.

References

[1] G. M. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the Spring Joint Computer Conference*, pages 483–485, 1967.

[2] N. Binkert, R. Dreslinski, L. Hsu, K. Lim, A. Saidi, and S. Reinhardt. The m5 simulator: Modeling networked systems. *Micro, IEEE*, 26(4):52 –60, july-aug. 2006.

[3] S. Borkar. Thousand core chips: a technology perspective. In *Proceedings of the 44th annual Design Automation Conference*, DAC, pages 746–749, 2007.

[4] K. Chakraborty. A case for an over-provisioned multicore system: Energy efficient processing of multithreaded programs. Technical report, 2007.

[5] S. Cho and R. Melhem. Corollaries to amdahl's law for energy. *Computer Architecture Letters*, 7(1):25 –28, Jan. 2008.

[6] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger. Dark Silicon and the End of Multicore Scaling. In *Proceedings of the 38th International Symposium on Computer Architecture*, June 2011.

[7] M. Hill and M. Marty. Amdahl's law in the multicore era. *Computer*, 41(7):33 –38, july 2008.

[8] S. W. K. Jaehyuk Huh, Doug Burger. Exploring the design space of future cmps. In *10th International Conference on Parallel Architectures and Compilation Techniques*, 2001.

[9] K. O. John D. Davis, James Laudon. Maximizing cmp throughput with mediocre cores. In *14th International Conference on Parallel Architectures and Compilation Techniques*, pages 51–62, 2005.

[10] J.-G. Lee, E. Jung, and D.-W. Lee. Asymptotic performance analysis and optimization of resource-constrained multi-core architectures. In *International Conference on Microelectronics*, pages 462 –465, Dec. 2008.

[11] S. Li, J. H. Ahn, R. Strong, J. Brockman, D. Tullsen, and N. Jouppi. Mcpat: An integrated power, area, and timing modeling framework for multicore and manycore architectures. In *42nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 469 –480, dec. 2009.

[12] T. Sato and A. Chiyonobu. Multiple clustered core processors. In *SASIMI*, 2006.

[13] C. D. Spradling. SPEC CPU2006 benchmark tools. *ACM SIGARCH Computer Architecture News*, 35(1):130–134, March 2007.

[14] M. Taylor, J. Kim, J. Miller, D. Wentzlaff, F. Ghodrat, B. Greenwald, H. Hoffman, P. Johnson, J.-W. Lee, W. Lee, A. Ma, A. Saraf, M. Seneski, N. Shnidman, V. Strumpen, M. Frank, S. Amarasinghe, and A. Agarwal. The raw microprocessor: a computational fabric for software circuits and general-purpose programs. *Micro, IEEE*, 22(2):25 – 35, mar/apr 2002.

[15] D. H. Woo and H.-H. Lee. Extending amdahl's law for energy-efficient computing in the many-core era. *Computer*, 41(12):24 –31, dec. 2008.