CCIT Report #514
January 2005

Performance, Power Efficiency and Scalability of Asymmetric Cluster Chip Multiprocessors

Tomer Y. Morad† Uri C. Weiser‡ Avinoam Kolodny† Mateo Valero* Eduard Ayguade*

†Department of Electrical Engineering
Technion, Haifa, Israel
{tomerm@tx , kolodny@ee}.technion.ac.il

‡Intel Corporation
Petach Tikva, Israel
uri.weiser@intel.com

*Departament d'Arquitectura de computadors
Universitat Politècnica de Catalunya
Barcelona, Spain
{mateo, eduard}@ac.upc.es

Abstract—**This paper evaluates asymmetric cluster chip multiprocessor (ACCMP) architectures as a mechanism to achieve the highest performance for a given power budget. ACCMPs execute serial phases of multithreaded programs on large high-performance cores whereas parallel phases are executed on many small and simple cores. Theoretical analysis reveals a performance upper bound for symmetric multiprocessors, which is surpassed by asymmetric configurations at certain power ranges. Our emulations show that asymmetric multiprocessors can reduce power consumption by more than two thirds with similar performance compared to symmetric multiprocessors.**

Index Terms—**ACCMP, CMP.**

I. INTRODUCTION

ACHIEVING high performance within a given power envelope is a major concern for microprocessor architects. In the past, the constant decrease in feature sizes has enabled to increase uniprocessor performance by packing more transistors in the same die area. Nowadays, due to the complexity of current state of the art microprocessors, a large increase in power and area results in only small performance improvements. According to Pollack's empirical rule [7], performance of uniprocessors is proportional to the square root of their area. These trends favor the use of Chip Multi Processors (CMP) [5].

Software applications contain serial phases as well as parallel phases. The lowest execution time for the serial phases is achieved by the highest performance uniprocessor available, since the serial phases can be executed on one processor only. On the other hand, parallel phases can be executed on numerous processors in parallel. Therefore, the lowest execution time for the parallel phases is achieved by executing them on many simple processors that consume less energy per instruction (EPI) [4]. We claim that a choice of symmetric cores is suboptimal due to the contradicting requirements of the serial and parallel phases within the same application.

We propose placing clusters of different cores on a single die. All of the cores within the Asymmetric Cluster Chip Multi Processor (ACCMP) share the same instruction set architecture and memory address space in order to let threads migrate from core to core. The larger high-performance cores will execute single-threaded programs and the serial phases of multithreaded programs. The smaller cores, though each providing lower performance than their larger counterparts, will execute the parallel phases at better performance versus power ranges. In the general case, ACCMPs will include clusters of different cores, since numerous multithreaded applications with varying parallelism execute in parallel. An example of an ACCMP with three clusters is illustrated in Fig. 1.

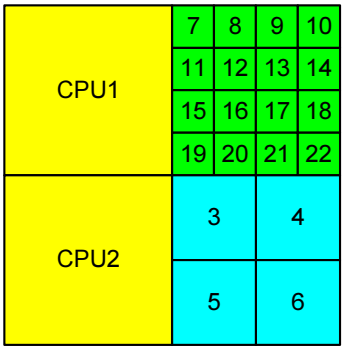

Fig. 1. An ACCMP floorplan with 22 cores. On the left is a cluster of large high performance general-purpose cores that consume high EPI. On the right are two clusters of smaller cores that consume less EPI.

Placing heterogeneous cores on a single die has been proposed in recent research [1], [4], [6]. Kumar et al. [6] have shown how a heterogeneous multiprocessor could achieve similar performance to a homogeneous multiprocessor for less power and area. Grochowski et al. [1], [4] have proposed and demonstrated an asymmetric multiprocessor by employing dynamic voltage and frequency scaling on a symmetric multiprocessor.

In this paper we propose a simple theoretical model for the performance of symmetric chip multiprocessors. We then find an upper bound for the performance per unit power (power efficiency) of such multiprocessors. By comparing this upper bound to the power efficiency of asymmetric chip multiprocessors we conclude that asymmetric structures can achieve higher power efficiency than any symmetric chip multiprocessor. Finally, we validate our findings by experimental emulation of ACCMP structures.

II. Analysis

A. Symmetric Chip Multiprocessors

Software applications have two types of phases, serial phases and parallel phases. The serial phases can be executed only on a single processor whereas the parallel phases can be executed on more than one processor. The variable λ denotes the fraction of dynamic instructions that reside in the parallel phases out of the total dynamic instructions.

For the purpose of mathematical analysis, we evaluate multithreaded applications whose parallel phases can be executed by up to the number of available hardware processors. We assume that the performance of a uniprocessor is a function of its area [7]. Thus, a processor with area size a will have performance of Perf(a), measured in instructions per second.

The number of dynamic instructions in the parallel phases of a program with M total dynamic instructions is λM. Since the parallel phases are executed by n identical cores in parallel, the execution time for the parallel phases is given by

$$T_{parallel} = \frac{1}{n} \cdot \frac{\lambda M}{\mathrm{Perf}(a)}. \quad (1)$$

The serial phases contain $(1-\lambda)M$ instructions, and are executed on one core only whose performance is Perf(a). Therefore, the time it takes to execute the serial phases is given by

$$T_{serial} = \frac{(1-\lambda)M}{\mathrm{Perf}(a)}. \quad (2)$$

When multiple processors share the same address space, memory coherence must be achieved by using a coherence protocol such as MESI. Two types of memory coherence conflicts exist. The first type is caused by a constant number of lock manipulations in the workload. The second type is caused by dynamic workload balancing mechanisms. Only the latter depends on the number of cores on the die.

We model the coherence penalty due to interactions between processors as a time penalty. We define k_1 to be the fraction of dynamic instructions out of the total dynamic instructions in the parallel phases that manipulate shared locks related to the workload itself. We define k_2 to be the fraction of

additional dynamic instructions out of the total dynamic instructions in the parallel phases required for workload distribution that manipulate shared data. These coefficients are measured in coherence transactions per instruction.

The time needed to complete a coherence transaction also depends on the distance cache lines need to travel on the chip. We assume that this distance is roughly proportional to the diameter of the chip. Therefore, this latency is proportional to $v^{-1}\sqrt{na}$, where $\sqrt{na}$ is the diameter of the die and v is the effective signal propagation velocity. The coherence time penalty is thus given by

$$T_{coherence} = \lambda M\left(k_1 + nk_2\right)v^{-1}\sqrt{na}. \tag{3}$$

The execution time of a multithreaded program on a symmetric CMP is given by $T_{SCMP}=T_{parallel}+T_{serial}+P_{coherence}$. Therefore, the performance of a symmetric CMP is

$$\mathrm{Perf}_{\mathrm{SCMP}} = \frac{M}{\mathrm{T}_{SCMP}} = \frac{n\mathrm{Perf}\left(a\right)}{\lambda + \left(1-\lambda\right)n + \lambda v^{-1}\sqrt{na}\left(k_1 + nk_2\right)n\mathrm{Perf}\left(a\right)}. \tag{4}$$

We estimate Perf(a) for a uniprocessor of size a by incorporating Pollack's empirical rule [7], which states that performance grows as the square root of area,

$$\mathrm{Perf}\left(a\right) = \eta\sqrt{a}. \tag{5}$$

For simplicity, we assume that power consumption is proportional to the area of the die. Power consumption of the memory system is not modeled. The total chip power is thus

$$P = \gamma na. \tag{6}$$

Using the above equations we derive the performance as a function of total chip power and the area of a single core,

$$\mathrm{Perf}_{\mathrm{SCMP}}\left(P,a\right) = \frac{\eta\sqrt{a}}{\lambda\left(\left(\frac{P}{\gamma a}\right)^{-1} + \frac{1-\lambda}{\lambda} + \eta\frac{\sqrt{a}}{v}\left(k_1 + \frac{P}{\gamma a}k_2\right)\sqrt{\frac{P}{\gamma}}\right)}. \tag{7}$$

We now find the performance upper bound of symmetric multiprocessors given a constant power budget. For mathematical simplicity, we assume static symmetric load balancing which limits the interactions between processors related to workload distribution, hence $k_2 \to 0$. This results in a higher upper bound. We find the optimum ($\partial \mathrm{Perf}_{\mathrm{SCMP}}\left(p,a\right)/\partial a = 0$) number of processors and the optimum area of each core in order to get the highest performance for a given power budget,

$$n_{opt} = \frac{\lambda}{1-\lambda}, \quad a_{opt} = \frac{P}{\gamma n_{opt}}. \tag{8}$$

The above results imply that maximum performance for symmetric multiprocessors is achieved when the execution time of the parallel phases is equal to the execution time of the serial phases. The maximum performance as a function of total multiprocessor power is thus:

$$\mathrm{Perf}_{\mathrm{SCMP,max}}\left(P\right) = \eta\sqrt{\frac{P}{\gamma}}\frac{1}{\lambda\left(2\sqrt{\frac{1-\lambda}{\lambda}} + \eta k_1\frac{P}{\gamma v}\right)} = \eta\sqrt{a_{opt}}\frac{1}{2\left(1-\lambda\right) + \lambda k_1\eta\sqrt{a_{opt}}v^{-1}\sqrt{a_{opt}n_{opt}}}. \tag{9}$$

When only one core is used, there is no performance loss due to coherency. The performance of one core is given by substituting (6) into (5) where n=1,

$$\mathrm{Perf}_{uniprocessor} = \eta\sqrt{\frac{P}{\gamma}}. \tag{10}$$

The upper bound for symmetric systems is the maximum between equations (9) and (10). Uniprocessors (10) achieve more performance for a given power budget than multiprocessors (9) when λ=0 or when P is sufficiently large:

$$P \geq \gamma v \frac{1-2\sqrt{\lambda(1-\lambda)}}{\lambda \eta k_1}. \tag{11}$$

Fig. 2 shows the performance versus power for λ=0.75. Four symmetric multiprocessors are modeled, with normalized core areas of 1, 2, 4, and 8 respectively. Power increases in Fig. 2 as additional cores are added to the system. When the core count is low, adding an additional core to symmetric multiprocessors results in improved performance. However, when the core count is high, performance degrades when additional cores are added due to coherence penalties. The multiprocessors are plotted with $k_2 \neq 0$, and do not cross the theoretical upper bound given by equations (9)-(11).

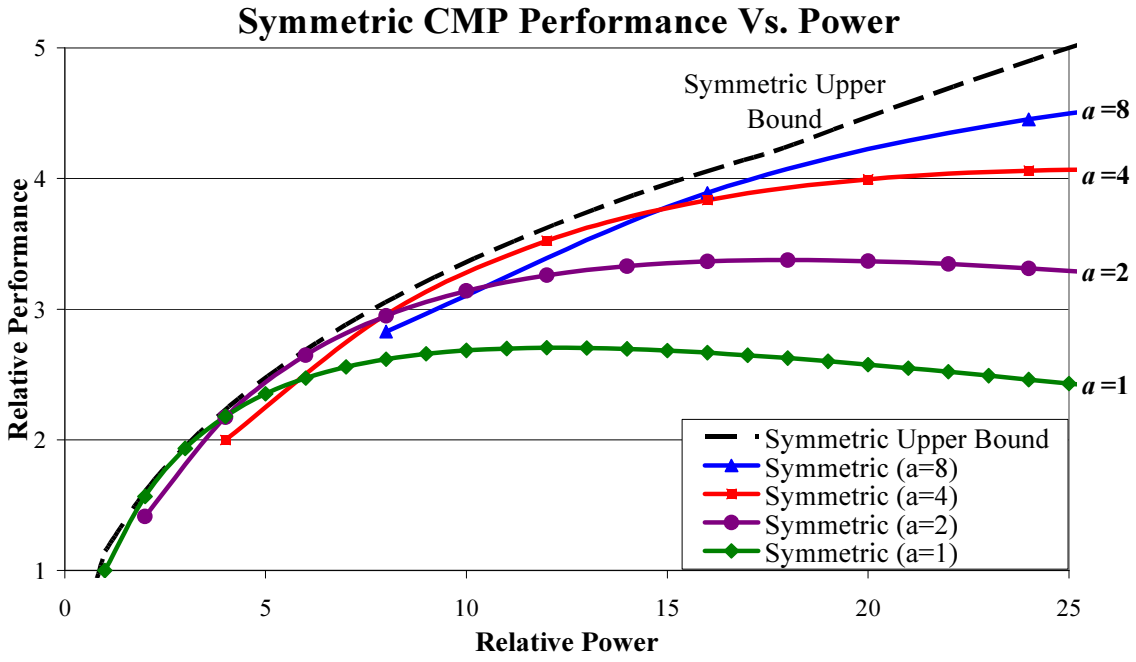

Fig. 2. The performance versus power for symmetric multiprocessors for λ=0.75, k_1=10^{-2}, k_2=10^{-3}, γ=1, η=1. Each data point in the graph represents a symmetric CMP with a different number of cores.

B. Asymmetric Cluster Chip Multi-Processors (ACCMP)

We focus on a special case of ACCMP that contains one large core of area βa and many small cores of area a, where β>1. We assume that the serial phases of programs are executed on the large core, whereas the parallel phases are executed on all n available cores. The time to execute a multithreaded program on the ACCMP is thus:

$$T_{ACCMP} = T_{ACCMP,Parallel} + T_{ACCMP,Serial} + T_{ACCMP,Coherence} =$$

$$= \frac{M\lambda}{\left(n-1+\sqrt{\beta}\right)\eta\sqrt{a}} + \frac{M(1-\lambda)}{\eta\sqrt{\beta a}} + M\lambda v^{-1}\sqrt{(n-1+\beta)a}\left(k_1 + nk_2\right). \tag{12}$$

Power of the ACCMP is given by

$$P = \gamma a(n-1+\beta). \tag{13}$$

The performance of ACCMP can be expressed as a function of total chip power,

$$\mathrm{Perf}_{ACCMP} = \frac{\eta\sqrt{a}}{\lambda\left(\frac{P}{\gamma a} - \beta + \sqrt{\beta}\right)^{-1} + (1-\lambda)\beta^{-1/2} + \lambda v^{-1}\eta\left(k_1 + nk_2\right)a\sqrt{\frac{P}{\gamma a}}}. \tag{14}$$

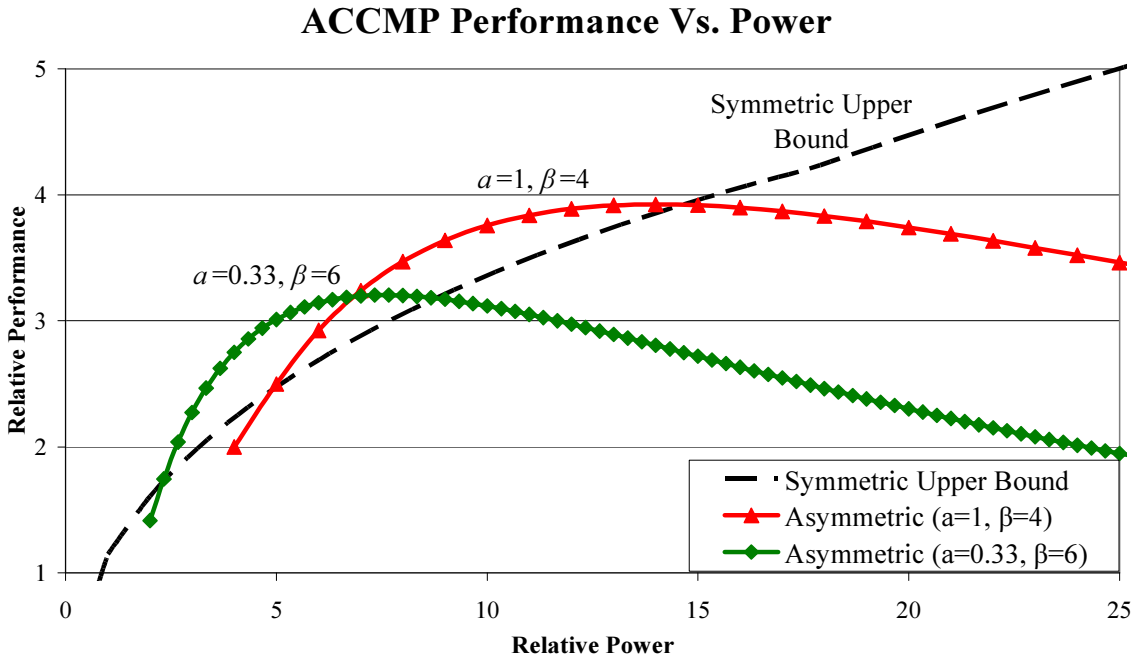

Fig. 3. The performance versus power for symmetric and asymmetric multiprocessors for λ=0.75, k_1=10^{-2}, k_2=10^{-3}, γ=1, η=1. Each data point in the graph represents a multiprocessor with a different number of cores. The graph shows two cases where an ACCMP crosses the upper bound of symmetric multiprocessors.

Performance versus power predictions for λ=0.75 is shown in Fig. 3. The curves show two asymmetric structures modeled according to (14). Their performance is plotted alongside the upper bound for symmetric multiprocessors given by equations (9)-(11). As can be seen, the asymmetric multiprocessors exceed the performance versus power achieved by symmetric multiprocessors at certain ranges of power. This is because asymmetric multiprocessors optimize the parallel phases and serial phases separately, whereas symmetric multiprocessors optimize both kinds of phases together.

III. Emulation Environment

We validate our theory by emulations. The emulations were done using one node of 16 Power3 microprocessors of a 162 processor IBM RS-6000-SP [3]. Different core sizes (performance) are modeled with the help of a "leech" program that executes in the background and degrades specific microprocessor performance. For example, in order to model 80% of the Power3 processor, the leech process would have to consume 20% of the CPU cycles. By using different parameters for the leech program, processors of different performance can be emulated. Software threads were bound to specific processors with the help of operating system calls.

A. Synthetic Benchmark

In order to test the emulation environment we have developed a synthetic benchmark using OpenMP. The synthetic benchmark contains simple computation loops with accesses to shared locks. 75% of the computational loop iterations reside inside parallel code regions whereas the rest reside in serial code regions (λ=0.75). The loop iterations in the parallel regions were distributed dynamically among the processors in chunks of 64 iterations, on a first come first serve basis.

B. SPEC-OpenMP Benchmarks

Apart from our synthetic benchmark, we have run a number of SPEC-OpenMP [2] benchmarks on our emulator. Special care was taken to avoid seeing a significant slowdown when adding a small core to a configuration with one large core. Unless specified otherwise, OpenMP compilers divide parallel work evenly among available processors. When the same amount of work is distributed between a high performance core and a low performance core, the net performance is reduced to that of two low

performance cores. The SPEC-OpenMP benchmark sources were augmented to include dynamic workload distribution. Additionally, processor binding code was added at the beginning of each benchmark.

IV. Emulation Results

The emulation results for the synthetic benchmark are shown in Fig. 4. For certain power regions, the three emulated asymmetric systems demonstrate better performance versus power than any symmetric system, in accordance with the theoretical equations. The highly asymmetric multiprocessor with a large core of βa=4 and 14 small cores of a=0.04 (β=100) achieved the best performance versus power. Note that a choice of β=100 implies a performance factor of 1:10 and power factor of 1:100 between the large and the small cores. A comparison between interesting symmetric and asymmetric configurations in Fig. 4 is shown in Table 1.

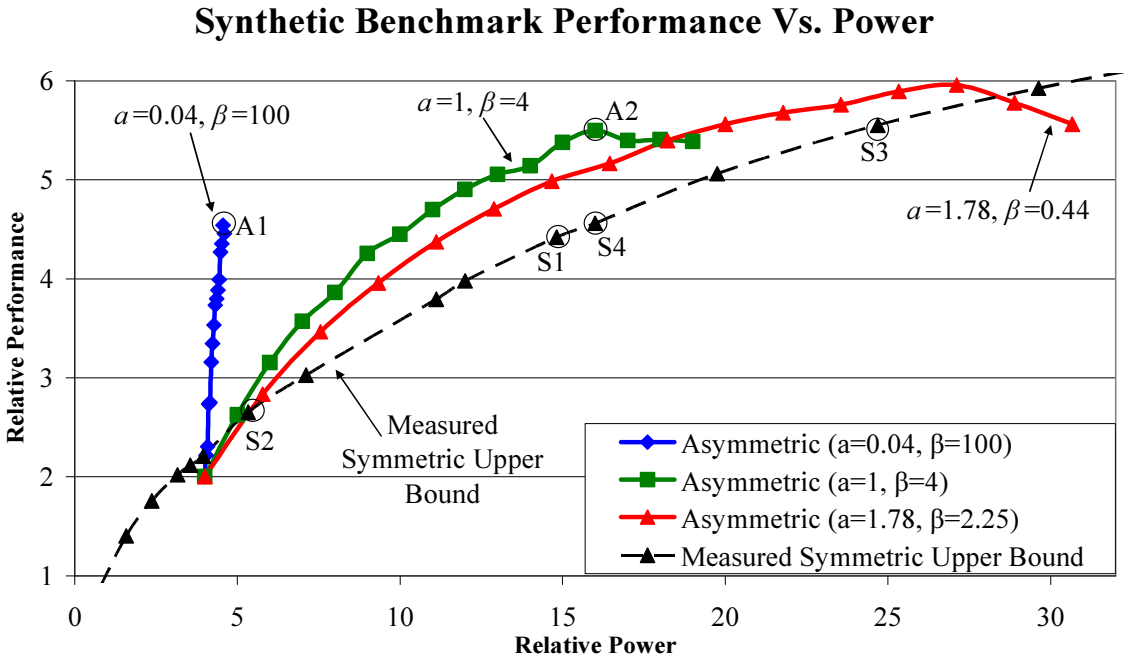

Fig. 4. Synthetic benchmark performance versus power. Various symmetric configurations were emulated in order to plot the symmetric upper bound. The asymmetric multiprocessors cross this boundary.

The results for the SPEC-OpenMP Wupwise benchmark with the "test" input set are shown in Fig. 5. When the "test" input sets are used, a substantial portion of the execution time is spent in serial initialization code. The asymmetric multiprocessors surpass the performance of every symmetric multiprocessor emulated in the shown power range.

TABLE 1 – COMPARISON OF DESIGN POINTS.

ACCMP				Symmetric CMP			Comparison	
Pt.	a	α	# Cores	Pt.	a	# Cores	Perf.	Pow.
A1	0.04	100	15	S1	4.96	3	+3%	-69%
A1	0.04	100	15	S2	1.78	3	+71%	-14%
A2	1	4	13	S3	4.96	5	-1%	-35%
A2	1	4	13	S4	4	4	+20%	0%
A3	1	4	7	S5	4	4	+2%	-31%

The Wupwise benchmark with the "train" input set required over an hour to complete on a single Power3 processor. Since this benchmark is highly parallel, we have not seen significant benefits from using ACCMP over symmetric multiprocessors. This is because the time required to execute the serial phases is negligible compared with the total execution time.

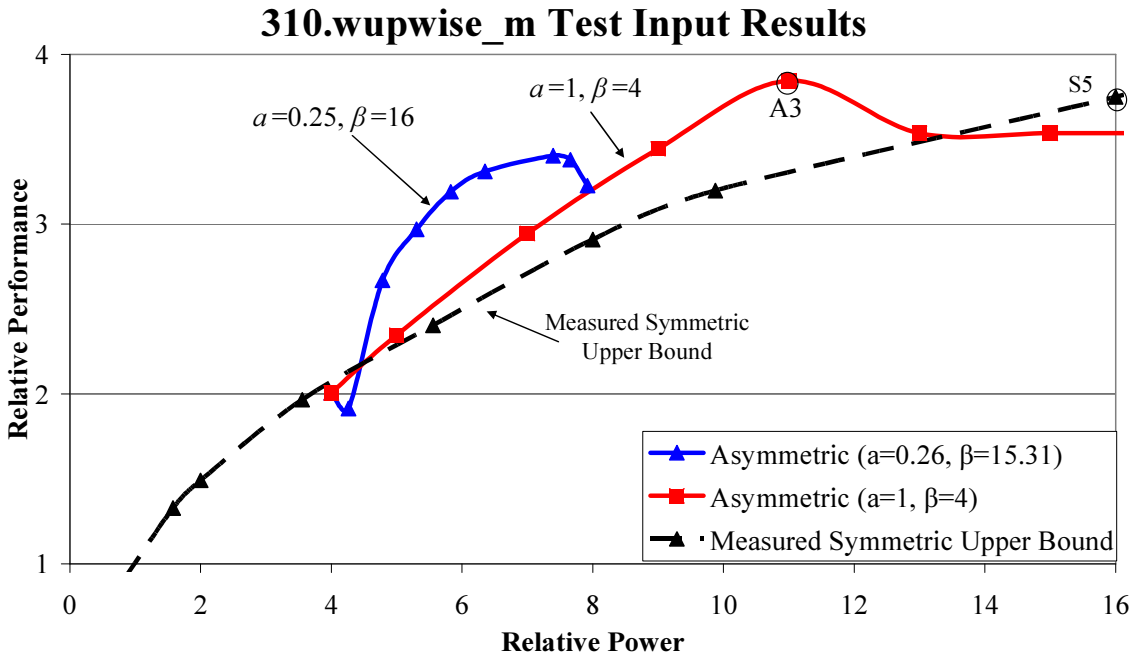

Fig. 5. Wupwise performance versus power with the "test" inputs. Various symmetric configurations were emulated in order to plot the symmetric upper bound. The asymmetric multiprocessors demonstrate better performance versus power than any symmetric multiprocessor.

Most SPEC-OpenMP benchmarks exhibited degradation when small cores were added next to large cores. The reasons for this stem from the fact that these benchmarks were written for symmetric systems. For example, the Galgel benchmark frequently calls the MATMUL function, which is a matrix multiplication function. This function was written for symmetric systems, so when small cores are introduced beside larger cores, performance degrades to the level of the small cores. Performance tuning for asymmetric load balancing for each benchmark should solve these problems.

V. Discussion

We have shown by theoretical analysis that at certain power ranges, asymmetric cluster chip multiprocessors can achieve higher performance for multithreaded applications than any symmetric CMP configuration. Our emulations of ACCMP structures show reduction of more than two thirds in power for similar performance, as well as more than 70% higher performance for the same power budget. ACCMPs excel in executing multithreaded programs in power constrained environments since the parallel and serial phases of software applications are executed on different types of cores.

Placing multiple asymmetric cores on a single die introduces many new challenges. Operating systems and compilers must support asymmetric scheduling and asymmetric load balancing. Additionally, ACCMP architectures must enable fast thread migration between cores, efficient inter-core communications and a scalable coherent memory system.

VI. Acknowledgements

This work was supported by HPC-Europa, funded by the European Commission's Research Infrastructures action, contract number RII3-CT-2003-506079. We also acknowledge the European Center for Parallelism of Barcelona (CEPBA) for supplying the computing resources for our experiments.

References

[1] M. Annavaram, E. Grochowski, and J. Shen. "Mitigating Amdahl's Law Through EPI Throttling." Submitted to the International Symposium on Computer Architecture (ISCA-35), 2005.

[2] V. Aslot et. al.. “SPEComp: A New Benchmark Suite for Measuring Parallel Computer Performance.” In Proceedings of WOMPAT 2001, Workshop on OpenMP Applications and Tools, July 2001.
[3] M. R. Barrios, M. Bono, M. Hennecke, T. Supapunpinyo, B. Woo, and S. Yap. “The RS/6000 SP Inside Out.” IBM, International Technical Support Organization, May 1999.
[4] E. Grochowski, R. Ronen, J. Shen, and H. Wang. “Best of Both Latency and Throughput.” In proceedings of the 22nd International Conference on Computer Design, October 2004.
[5] L. Hammond, B. A. Nayfeh, and K. Olukotun. “A Single-Chip Multiprocessor.” IEEE Computer, September 1997 (Vol. 30 No. 9).
[6] R. Kumar, D. Tullsen, P. Ranganathan, N. Jouppi, and K. Farkas. "Single-ISA Heterogeneous Multi-core Architectures for Multithreaded Workload Performance." In proceedings of the 31st International Symposium on Computer Architecture, June 2004.
[7] F. Pollack. "New Microarchitecture Challenges in the Coming Generations of CMOS Process Technologies." Micro 32, 1999.