Using the TOP500 to Trace and Project Technology and Architecture Trends

Peter M. Kogge
Univ. of Notre Dame
384 Fitzpatrick Hall
Notre Dame, IN 46556
kogge@cse.nd.edu

Timothy J. Dysart
Univ. of Notre Dame
384 Fitzpatrick Hall
Notre Dame, IN 46556
tdysart@cse.nd.edu

ABSTRACT

The TOP500 is a treasure trove of information on the leading edge of high performance computing. It was used in the 2008 DARPA Exascale technology report to isolate out the effects of architecture and technology on high performance computing, and lay the groundwork to project how current systems might mature through the coming years. Two particular classes of architectures were identified: "heavyweight" (based on high end commodity microprocessors) and "lightweight," (primarily BlueGene variants), and projections made on performance, concurrency, memory capacity, and power. This paper updates those projections, and adds a third class of "heterogeneous" architectures (leveraging the emerging class of GPU-like chips) to the mix.

Categories and Subject Descriptors

C.0 [**General**]: System Architectures; C.4 [**Performance of Systems**]: Design studies, Performance attributes

General Terms

Design, Performance

Keywords

TOP500, Exascale, heterogeneous systems, system projections

1. INTRODUCTION

With 18 years of semi-annual reporting, the TOP500 list has become perhaps the longest-lived organized source of data on computer architectures and systems, at least as they apply to a specific benchmark. As such, mining these records can provide excellent insight into trends in architecture and technology, especially at the high end.

This paper performs such an analysis. It is certainly not the first to do so; indeed the TOP500 website itself routinely produces graphs of top level characteristics such as performance, architecture style, manufacturer, etc. Earlier papers such as [7] and [12] have used the data to analyze trends, and make predictions about reaching various milestones such as a petaflop/s[1]. More recently, a DARPA-funded working group in 2007 [13][2] used the data to extrapolate whether or not reaching an exaflop/s was feasible in a short time frame. The answer from that report then heavily influenced followup efforts such as UHPC.

This paper updates the analysis from the Exascale report in both data covered and metrics analyzed. In addition, while the prior report identified two generic classes of systems, "heavyweight" and "lightweight," we introduce here a third class "heterogeneous," to reflect new types of entrants that have appeared recently on the list. Further, we have done this analysis with an eye towards repeating it after each list announcement, and in normalizing definitions and measurement characteristics to help guide new list entrants as to what parameters would be of most help in understanding the underlying trends.

The paper is organized as follows. Section 2 standardizes the key definitions we will use in the paper. Section 3 discusses the much more detailed database we have developed about members of the various TOP500 lists. Section 4 overviews the three classes of architectures we considered. Section 5 develops some key metrics from the historical data. Section 6 develops the technology roadmaps and trend assumptions we use. Section 7 uses all this material to analyze the observed trends and make projections as to the future. Section 8 concludes.

2. TERMINOLOGY

From its beginning, the TOP500 has focused on performance of LINPACK on the leading parallel computers of the day. To allow for year-over-year comparison, the rankings have used a few simple terms to specify both the hardware architecture and the performance. While satisfactory for most of the early years, with the rise of SMPs, multicores, and now heterogeneous architectures, making "apples-to-apples" comparisons has become more difficult, especially when detailed analysis and projections are desired as was done for the Exascale report. In the following subsections we review the terminology used in this paper, and suggest that the use of some variants in future TOP500 reports may

SC11 November 12-18 2011, Seattle, Washington, USA
Copyright 2011 ACM 978-1-4503-0771-0/11/11 ...$10.00.

[1] We use "flop" for a single floating point operation, "flops" to refer to multiple floating point operations without a time reference, and "flop/s" to refer to some number of them executed per second.

[2] Hereafter referred to as the "Exascale report." or "Exascale study."

simplify future analysis.

2.1 Basic System Terms

Until the November 2008 list, the key reported parameter used to describe the architecture of a system was "Processor Count." As discussed above, this became quite fuzzy with the rise of multi-core, and was replaced with "core count." Even that, however, is still inadequate for the kinds of projections done here. Consequently, the following is the list of terms used in this paper:

- **Core**: for a CPU, a set of logic that is capable of independent execution of a program thread. For GPUs, the terms *SIMD core* and *Streaming Multiprocessor* (SM) have been used by AMD and NVIDIA respectively. Each such unit has a number (typically 16-48) of replicated simple processors (unified *shaders*) which contain the ALUs and increasingly FPUs.
- **Socket**: a chip that represents the main site where conventional computing is performed. This term reflects that when one looks at a modern HPC system, the biggest area on a board is devoted to the heatsink and socket that sandwich the microprocessor chip. In earlier rankings, a socket was primarily a single-core microprocessor chip, and thus counting sockets was equivalent to counting "processors." Today, a socket typically supports a large number of cores.
- **Node**: the set of sockets, associated memory, and NICs that represent the basic replicable unit of the system, *as seen by application software*. Several years ago the term "node" came into vogue, and was used interchangeably with "processor" and "socket." With the rise first of chip sets that couple multiple identical microprocessor chips into a single unified SMP, and then by the rise of multi-socket machines with mixes of heterogeneous microprocessors (as in Roadrunner), it became important to make this distinction.
- **Core Clock**: the clock rate of a core. In systems with multiple types of cores, we will use here the one that most contributes to overall computation.
- **Memory Capacity**: the aggregate physical memory (typically DRAM) that is directly addressable by programs running in cores within sockets, within nodes, not including caches.
- **Power**: total power consumed by the system. This metric is today often inconsistent across system descriptions, depending on whether or not file systems (disk drives) are included, and/or whether the power required for cooling and power conditioning is counted. In the future it may be more consistent to quote perhaps "CEC power" as the power actually drawn by the computing electronics, "system power" to include secondary storage and local power conditioning, and "facility power" to include cooling and power conversion and redundancy effects.

In the days of single microprocessor-type systems, aggregate numbers for these parameters were adequate. However, with heterogeneous systems, there has to be separate accounting for each type of processor in the system. This is especially true for memory, where different "sockets" with different processor types have different memory interfaces.

2.2 Metrics

A second set of terms is important from the standpoint of comparing systems. The most obvious ones are R_{peak} and R_{max} which drive the TOP500 rankings [8]. R_{peak} is the theoretical peak performance, computed in Gflop/s, as the number of FPUs times the clock rate of each FPU. R_{max} is the performance, in Gflop/s, for the largest problem run.

The Exascale report added the following metrics:

- **Thread Level Parallelism** (**TLP**): the number of distinct hardware-supported concurrent threads that makes up the execution of a program. None of the top systems to date have been explicitly multi-threaded, although newer chips do support at least a low level of "Hyper Threading." Consequently, for this study each core as reported in the TOP500 list is assumed to correspond to a single thread of execution.
- **Thread Level Concurrency** (**TLC**): is an attempt to measure the number of separate operations of interest that can be executed per cycle. For the TOP500 the operation of interest has been floating point. For homogeneous systems, it is computed as the performance metric (R_{max} or R_{peak}) divided by the product of the number of "threads" (i.e. cores) and the clock rate. TLC is meant to be similar to the Instruction Level Parallelism (ILP) term used in computer architecture to measure the number of instructions from a single thread that can either be issued per cycle within a microprocessor core (akin to a "peak" measurement), or the number of instructions that are actually completed and retired per second (akin to the sustained or "max" numbers of the current discussion).
- **Total Concurrency** (**TC**): the total number of separate operations of interest that can be computed in a system at each clock cycle. For the TOP500 such a measure reflects (within a factor of 2 to account for fused multiply-add) the total number of distinct hardware units capable of computing those operations. This metric can be computed as the number of cores times the peak TLC, and is important because it reflects the explicit amount of concurrency that must be expressed by, and extracted from, a program to utilize the hardware efficiently.
- **Flop/s per watt** (**FPW**): the performance of the system divided by system power.
- **Energy per flop** (**EPF**): the reciprocal of the above, in units of Joules (or more conveniently in most cases "picoJoules" (pJ), 10^{-12} Joules).

We note that many of these metrics may actually be computable in two forms depending on whether R_{max} or R_{peak} is used. We also note that the original goal of the Exascale study was for an exaflop/s at 20MW, which is equivalent to an energy expended per flop of 20pJ.

3. THE TOP500 DATABASE

We started our construction of a database from the spreadsheets available from the web. Depending on the year, the base information varied, but always included the performance metrics, the processor/core count, and a text string with some more detail. Often missing was not only enough detail to describe the system at the socket/board/rack level but also other key parameters such as power, memory, and disk space. Also missing was quantifiable insight into evolutions of the same system over time (nodes missing during test runs, the addition or upgrade nodes, etc.).

Consequently, we spent considerable time researching each and every system ever listed in one of the top 10 positions on the list (to do all 500 was beyond our resources). Whenever possible, we added missing data and resolved discrepancies. We separated out "sites" (physical facilities) from ranking positions in the top 10 so that we could look at just unique facilities when appropriate, rather than see duplicates show up simply because of longevity on the list. Within systems, we also separated out processor information to a common list, so as attributes of a particular chip were uncovered (e.g., V_{dd}, power dissipation, properties of their cache hierarchy or the number of memory ports they supported), those attributes could be rolled up in analyzing all systems that used them.

Reflecting the sea change from single core to multi-core to heterogeneous multi-core multi-chip nodes, we also kept separate columns of data for different classes of sockets in the same node (as in a conventional microprocessor paired with a GPU), and sockets whose cores are themselves heterogeneous (as in a Cell).

To the maximum extent possible, these data sets are augmented with references.

4. SYSTEM ARCHITECTURES

To baseline where "business-as-usual" might take us, this section defines three generic "strawmen" system architectures that are emblematic of where evolution of today's leading edge HPC data center class systems might lead. The architectures used as the departure points for this study are:

- "heavyweight node" Jaguar-class machines that use commodity leading edge microprocessors at their maximum clock (and power) limits,
- "lightweight node" Blue Gene-class machines where special processor chips are run at less than max power considerations so that very high physical packaging densities can be achieved,
- and "heterogeneous node" systems where there is a mix of processing chip types in each node.

The first two are extensions of what appeared in the Exascale report[13]; the last reflects the rise of systems using accelerators such as the Cell processor in Roadrunner [3] and GPUs in systems like Tianhe-1A [15]. In particular, we assumed as a baseline NVIDIA's Fermi chip that has 16 SMs, each with 32 shaders, and a fused multiply add (FMA) unit shared between each pair of shaders.

4.1 Heavyweight Architectures

Many machines in the current TOP 500 such as Jaguar, and most systems of the early years, are implemented around the leading edge of microprocessor technology, and consist of multiple large boards that are packaged in racks. Each board includes:

- Multiple (today typically 4) leading edge microprocessors, each of which is the heart of a node. A substantial metal heat sink is needed for each microprocessor, thus leading to the name "heavyweight" node architecture.
- For compatibility with the Exascale report we assume a baseline machine of this type used a 90nm high performance logic technology in a dual core chip. This corresponds to a 2004 chip in a 2006 system such as Red Storm.
- Each node includes a set of relatively commodity daughter memory cards holding commodity DRAM chips (DDRx, DIMMs, or FB-DIMMs are most common).
- Multiple specialized router chips provide interconnection between the on-board nodes and other boards in the same and other racks. These chips also have "heavy" heat sinks on them.

In summary, our baseline matches the Exascale report, namely 4 sockets per board and nominally 24 boards per rack. A 2.5 MW system consisting of 155 racks contained 12,960 sockets for compute and 640 sockets for system, or one system socket for approximately every 20 compute sockets. At 40 TB of storage and an R_{peak} of 127 teraflop/s, the baseline had a ratio of about 0.31 bytes per peak flop/s.

4.2 Lightweight Architectures

The basis for this system class is the Blue Gene series of supercomputers, both the "/L"[1, 4, 5] and the "/P"[10] series. Here the basic unit of packaging is not a large board with multiple, high powered chips needing large heat sinks, but small "DIMM-like" **compute cards** that include both memory and multiple small, low-power, multi-core **compute chips**. For BlueGene/L, each chip anchored a node and a compute card held two chips. 16 compute cards and up to two I/O cards were plugged into a node board, and 16 node boards would then be plugged into a **midplane**. A rack then consisted of two midplanes (32 node boards, 512 compute cards, 1024 compute chips).

The key to this high density stacking is keeping the processing core power dissipation low enough so that only small heat sinks and simple cooling are needed. This is achieved by keeping the architecture of the individual cores simple and keeping the clock rate down. The latter has the side-effect of reducing the cycle level latency penalty of cache misses, meaning that simpler (and thus lower power) memory systems can be used.

The desired properties for the Blue Gene/L core were accomplished by using a 7 stage pipeline with a two instruction issue to 3 pipelines (load/store, integer, and other). Also available is a floating point-unit with 2 fused multiply-adders and an extended ISA that supports simple SIMD operations at up to the equivalent of 4 flops per cycle. The compute chips also included complete memory controllers along with interface and routing functions.

For compatibility with the Exascale report we assume our study baseline machine of this type started with a 130 nm low power logic technology (as was used in the BG/L).

4.3 Heterogeneous Architectures

These systems have only recently appeared in the top 10 of the TOP500 list. Roadrunner [3] was the first to appear and debuted at number 1 on the June 2008 list using the Cell processor as its accelerator. Systems using GPUs as their accelerators first appeared in this portion of the TOP500 list in Nov. 2009 and, on the most recent TOP500 list held three of the top 10 slots (positions 2, 4, & 5). Descriptions of these systems can be found in [19, 14, 15, 9]. In general, the nodes for each of these systems consist of heavyweight processors, with accelerators on attached boards. The heavyweight processors are responsible for managing the accelerators.

While these heterogeneous systems are quite new, we anticipate two major system organizations may appear in the future; in the long term, convergence between these organizations is likely. The first organization utilizes multi-socket nodes where each socket is a homogeneous device. The second organization utilizes uni- or multi-socket nodes where each socket contains identical heterogeneous devices. Clearly, the GPU systems fit into the first category whereas Roadrunner is a bit of a blend given that the Cell processor is itself a heterogeneous device. However, based on the identical device requirement of the second system type, Roadrunner is a better fit for the first system type.

Organization 1: Here, not only are chips of varying types utilized, but we also anticipate that the regular processing cores will continue to be responsible for managing the accelerator resources. Options for the accelerator boards include devices like GPUs, Cell processors, FPGA or other reconfigurable devices, or Intel's *Many Integrated Core (MIC)* architecture. The Intel MIC chips contain a large number of x86 based cores for use as coprocessors. A demonstration version, Knight's Ferry, is discussed in [17, 18].

Systems utilizing the first two types of accelerators are in use today and more systems based on these will be developed. We are unaware of any recent systems in the TOP500 utilizing FPGAs as accelerators. Since the Intel MIC architecture is a recent development, no large systems using this architecture exist yet. Should Intel continue developing the architecture and necessary software tools, we anticipate systems using it will appear in the TOP500 list.

Organization 2: Systems of this type, while not yet in existence, will utilize a number of identical chips in their nodes. These chips will include both regular processing cores and other cores that are architecturally similar to the accelerators listed above. Current examples of these chips are combining CPU and GPU capabilities and include Intel's *Sandy Bridge* architecture [11], AMD's *Fusion* architecture [2], and NVIDIA's Project Denver [6].

This paper focuses on systems of the first type since they are in use today and will probably be the dominant form in the time frame under consideration.

5. OBSERVATIONS

The simplest kind of observations about the TOP500 are found in virtually all discussions, namely fitting simple trend lines to the top level system metrics of the listed systems. Fig. 1 lists R_{peak}, R_{max} (sustained performance), and system memory capacity as a scatter plot versus time. This curve plots just the top 10 systems from each list. Also included are simple curves based on a constant compound annual growth rate that extend from the peak of the current data out through 2016. These historical curves grow at 1.85X and 1.92X for R_{peak} and R_{max}, and 1.79 for memory.[3]

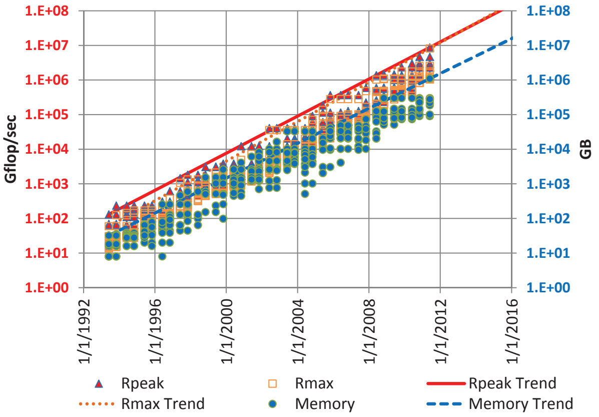


Figure 1: Top Level Characteristics.

While these trend lines are useful, they can be a bit misleading since they only consider the top performing systems.

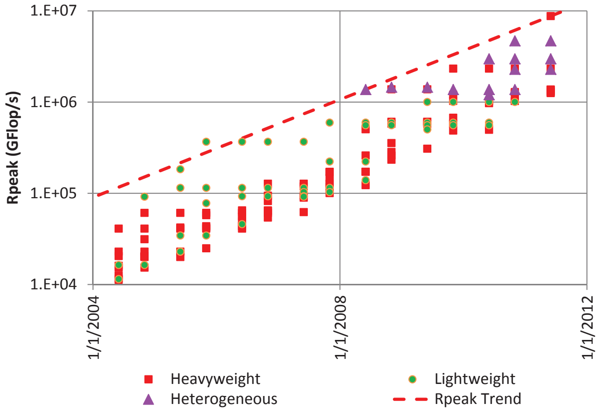


Figure 2: System Organizations since 2004.

Distinguishing between system organizations as in Fig. 2, we note that heavyweight systems were at the top of the list until 2004 when the first lightweight systems appeared. These lightweight systems quickly took over many of the top rankings, but since 2008, they are being replaced by the heterogeneous systems.

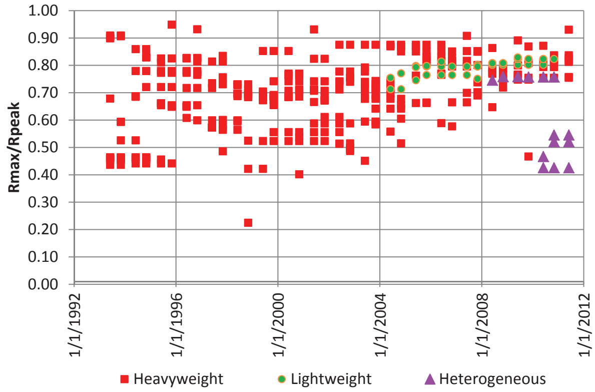


Figure 3: Floating Point Efficiency.

[3] The trend lines for R_{peak} and R_{max} merge on the chart because as discussed later there has been a convergence of their ratio at about 80%.

Fig. 3 shows the percentage of peak flop/s (R_{max}/R_{peak} that are sustained for LINPACK over time. This value varies without discernible trends between about 0.5 and 0.9 up until 2004. From 2004 to 2008 there was a tightening around 0.8. However, the introduction of heterogeneous systems seems to imply a rapid decrease in efficiency. This latter point may be partially explained by the difficulty in keeping large numbers of shaders busy at the same time. Also, during execution only the GPU cores are really used, leaving the heavyweight host capabilities idle.

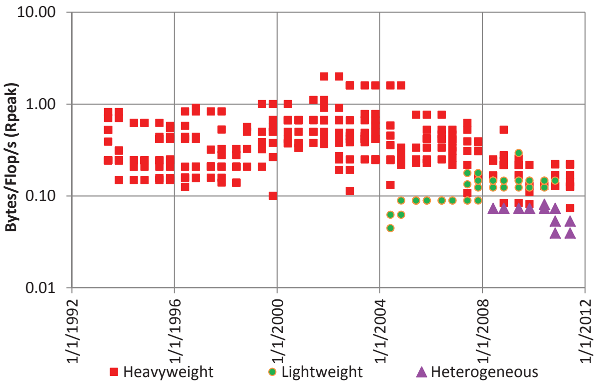


Figure 4: Bytes per Flop/s (R_{peak}).

Fig. 4 displays the bytes of storage provided per flop/s for which the system is designed at peak. As with the efficiency ratio, this value varies widely up until about 2005; since then, there is a distinct trend towards lower ratios, especially for lightweight and heterogeneous designs. We have also plotted this value against R_{max} to more accurately reflect application performance and have found that this downward trend remains obvious.

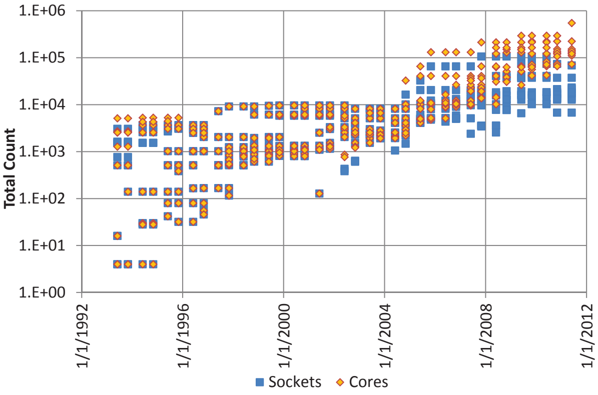


Figure 5: Growth in Socket and Core Count.

Fig. 5 gives a timeline for the growth in sockets and cores. Up until 2004 the core count overlaps the socket count - a core per socket was the norm. Since then, however, there is a growing divergence, reflecting the growth in multi-core. The "core count" is equivalent to the TLP measure introduced earlier. If we plotted "FPU count," we would see additional peaks for vector machines such as the Earth Simulator and in particular for GPUs, with dozens of shaders per "core."

Perhaps of even more interest are details of the way performance is obtained. Fig. 6 graphs the max clock rate of the cores in a system most responsible for computation (in heterogeneous systems the GPU rates are often slower than that of the host processors). This graph demonstrates several interesting observations. First, the clock rate did in fact flatten around 2004. Second, there is a clear bifurcation: the lightweight machines chose a significantly lower clock rate than the commodity heavyweight, as did the new heterogeneous systems.

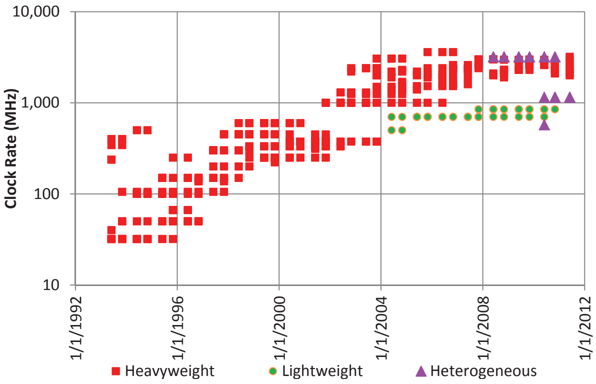


Figure 6: Trends in Clock Rate.

Thread level concurrency (TLC) is a good metric of the complexity of individual cores. As was observed in the Exascale report, there has not been radical changes in this metric until recently. In fact, excluding some vector machines, the value of 2 or 4 has been almost the default for most of the time. The emergence of the GPU-driven heterogeneous architectures has begun to significantly raise this number since each core in a GPU consists of numerous (16-48) shaders, with each shader contributing some amount of flops per cycle. Even at 0.5 flop/cycle, a GPU core with 16 shaders would contribute 8 to TLC.

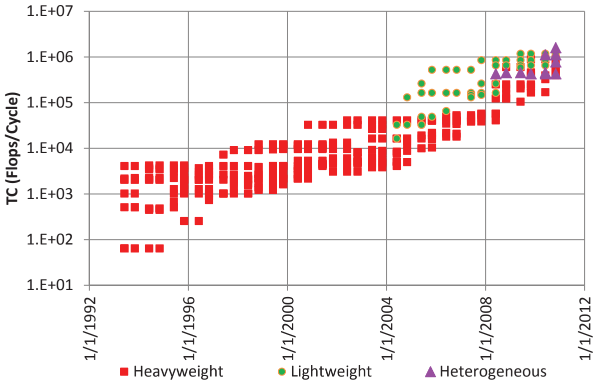
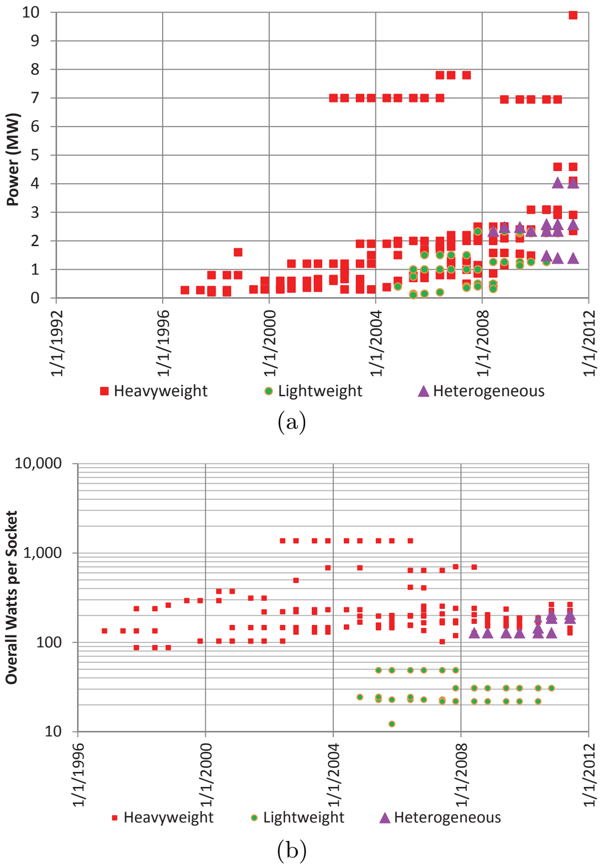


Figure 7: Total Concurrency (TC).

Next, total concurrency (TC) is the total number of operations that must be "in the air" at each and every machine cycle. Fig. 7 shows for that the long stretch of time when technology permitted clock rates to rise, the TC value stayed between a few thousand to a few tens of thousands. However, once the clock rate flattened (about 2004), the only way to get additional performance was through brute parallelism, and this number then began to take off.

Fig. 8(a) diagrams the trends in total system power. The



(a)

(b)

Figure 8: System power (a)in total (b)per socket.

trends here are a bit harder to see, other that what appears to be the beginning of a big uptick starting in 2009. Fig. 8(b) is perhaps more revealing by computing total system power divided by the number of sockets. Here the heavyweights clearly fall in the 200-300 watts per socket range, the lightweights in the 20-30, and the heterogeneous in the 100+ range. We note that in the latter case, the socket count includes the microprocessors and the GPUs.

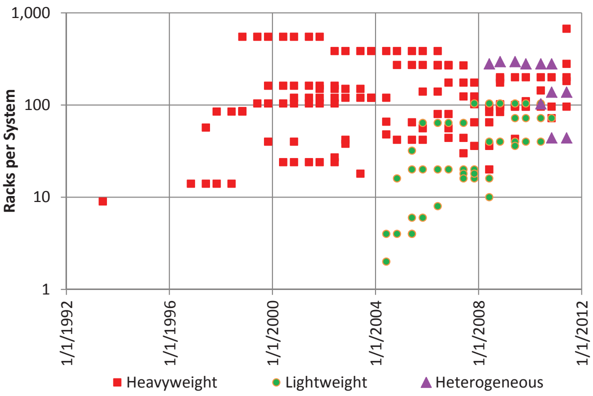


Figure 9: Rack Count.

Finally, Fig. 9 gives a feeling for the overall size of systems in terms of the number of racks. Heavyweight systems have for the most part stayed in the 150-200 range, while the heterogeneous have grown closer to 300.

6. TECHNOLOGY ROADMAPS

This section describes the roadmaps used to project future trends in the three classes of systems described previously. For the most part, the trend-generation approaches assumed are updates to those described in the Exascale report, as are the baseline systems against which the trends are projected. While trend lines could have been drawn from more current systems, we deliberately used the baselines for the heavy and light weight strawmen from the report. Using these strawmen, which are now several years old, allows us to gauge the accuracy of the projection process based on how recent machines have fit the projected trends.

We did, however, assume as a baseline for the heterogeneous a more modern point, since the first system of this type, Roadrunner, debuted in the top 10 in June 2008 with GPU based systems appearing in later lists.

6.1 Technology Trends

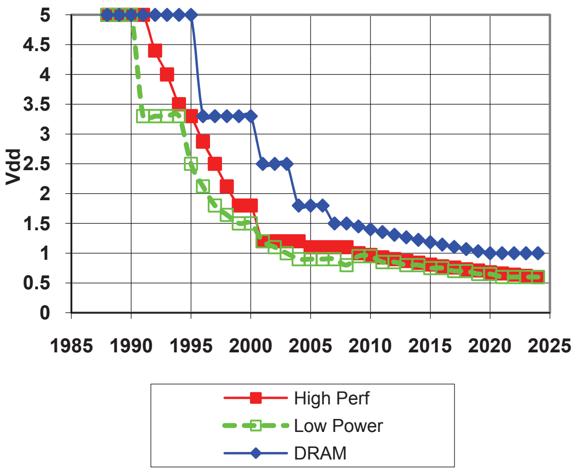


Figure 10: V_{dd}.

The Exascale report includes a set of charts projecting a variety of key chip parameters. Fig. 10 shows the updated chart for V_{dd} based on the most recent ITRS [16] projections. This figure includes both high performance and low power logic, along with projections on the supply voltage for commodity DRAM chips. V_{dd} is an important parameter since, assuming a constant clock rate, dynamic power for logic is often approximated as CV^2, where C is the aggregate capacitance of the circuit. To a reasonable first approximation, the capacitance for the same circuit scaled to a new technology scales with feature size. Fig. 11 then plots this relative improvement in dynamic power.

The above curve assumed a flat clock rate. However, the inherent speed of devices (and thus a maximum clock rate) will continue to improve, roughly in line with the reciprocal of capacitance, and thus the reciprocal of feature size. If such rates were actually used as core clocks, and we kept die size constant through time, then the total power of the die actually *grows* inversely with feature size (halving the feature size doubles the power dissipated by the chip). If we

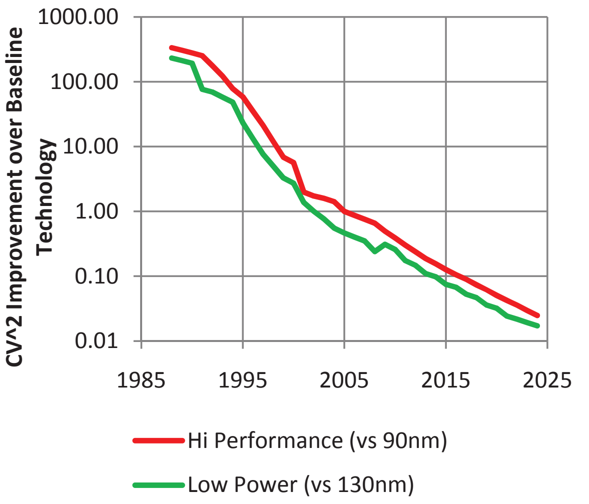


Figure 11: Dynamic Power Decrease over Time.

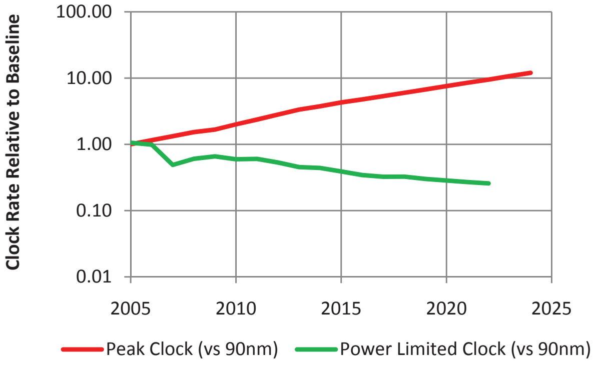


Figure 12: Power Constrained Clock Rate.

constrain the power dissipation of the chip to some limit, this then constrains the operating clock rate. Fig. 12 diagrams the relative change in clock rates assuming chip dissipation is bounded.

Finally, the capacity of commodity DRAM chips is key. Again we assume growth as forecast in the ITRS. We do not consider here the potential changes brought out by newly emerging technology such as chip stacking[4].

6.2 Heavyweight Strawman Scaling Assumptions

The assumptions we make for this paper about how the heavyweight architectures would mature over time from the 2004 baseline are drawn from the Exascale report and are listed below:

1. The microarchitecture for each core in the microprocessor will be approximately unchanged in complexity (i.e. transistor count) over the baseline.
2. The die size for the microprocessor chip will remain approximately constant, meaning that the number of cores on each microprocessor chip may grow roughly as the transistor density grows.
3. We do not account for relatively larger L3 caches to reduce pressure on off-chip contacts. This actually results in a possible overestimate of peak chip performance because we end up with more cores per die.
4. Such chips will continue to use high performance logic, with V_{dd} flattening as in Fig. 10.
5. The power dissipation per die will grow very slowly, as projected in the ITRS.
6. Per core, the microarchitecture improves from a peak of 2 flops per cycle in 2004 to a peak of 4 flops per cycle in 2006, and 8 flops per cycle in 2013.
7. The system will want to maintain the same ratio of bytes of main memory to peak flops as today. This will be done by using whatever natural increase in density comes about from commodity DRAM, but coupled with additional memory cards as necessary if that intrinsic growth is insufficient.
8. The maximum number of sockets (i.e. nodes) per board will double a few times. This is assumed possible because of a possible move to liquid cooling, for example, where more power can be dissipated and allowing the white space to be used and/or the size of the heat sinks to be reduced. This projections assumes that this may happen at roughly five year intervals.
9. The maximum number of boards per rack for 2006 will increase by perhaps 33% again because of assumed improvements in physical packaging and cooling, and reduction in volume for support systems. For this projection we will assume this may happen once.
10. The maximum power per rack will increase by at best a power of 16, to somewhere around 200-300KW. We assume this is a doubling every 3 years.
11. The maximum number of racks in a system was set to 155 in 2006 (to match then current systems), with an increase by 50 every year, up to a maximum of 600.
12. Secondary storage (and its growth) for scratch, file, or archival purposes is ignored. This storage must also undergo significant increases.

The above assumptions are reasonable for estimating power in the microprocessor parts of a system. There is, however, concern for how accurate such scaling rules would be for other parts of the baseline, particularly the memory system and the routers. This is because a significant amount of their power comes not from internal processing but from I/O - the transfer of data across chip boundaries. While such increases can be projected for commodity memory, the number of memory ports per socket is something that is changing. Also, both the protocols and the bandwidth of the router chips, and how they tie into the memory and microprocessors in a node may change in a complex fashion.

To simplify our projections, we thus continue the approach suggested in the Exascale report by adopting two energy models. The *Scaled* model assumes that the power per microprocessor chip grows as the ITRS roadmap has predicted,

[4] see for example: Micron's "Hybrid Memory Cube" http://www.micron.com/innovations/hmc.html

and that the power for the memory associated with each socket grows only linearly with the number of memory chips needed (i.e. the power per memory chip is "constant"). We also assume that the power associated with both the routers and other common logic remains constant. This is the same as the "Simplistically Scaled" model in the Exascale report. In a real sense, we are assuming here that both memory access *energy* and the *energy* cost of moving a bit, either across a chip boundary or between racks, will *decrease* (or "scale down") with technology advances, at least as fast as the increase in flops, with the total power constant because of a concurrent "increasing" of the flops rate of the microprocessor most probably requires a higher number of references to memory and a higher traffic through the routers. This is clearly optimistic.

In contrast, the *Constant* model assumes the microprocessor power grows as above, but that both the memory and router power scale linearly with the peak flops potential of the multi-core microprocessors. This naively assumes that the total energy expended in these two subsystems is used in accessing and moving data, and that the energy to handle one bit (at whatever the rate) is *constant* through time (i.e. no improvement in I/O energy protocol). This is clearly an over-estimate, and liable to be highly pessimistic. It is similar to the "Fully Scaled" model from the Exascale report.

Neither model takes into account the effect of only a finite number of signal I/Os available from a microprocessor die, and the power effects of trying to run the available pins at a rate consistent with the I/O demands of the multiple cores on the die.

6.3 Lightweight Strawman Scaling Assumptions

For consistency, the scaling assumptions used for the lightweight strawman are modified only slightly from those above for the heavyweight, and again are based on those trends used in the Exascale report. In particular, we reuse assumptions 1, 2, 3, 8, 10, and 12 in total. The list below slightly modifies assumptions 4, 6, 7, 9, and 11, in order, from the heavyweight list:

- V_{dd} flattens as for low power, not high performance, logic.
- Per core, the microarchitecture will improve from a peak of 2 flops per cycle to a peak of 8 flops per cycle in two steps.
- The bytes of main memory per flop ratio will match that of BlueGene/L.
- There is one doubling of compute cards per node board.
- The number of racks in 2005 and 2006 are set to the same as observed in real systems in order to validate the mode. After 2006, the model assumes growth in racks as for the heavyweight.

The following assumptions are specific to these systems:

- The power dissipation per chip will be allowed to increase gradually to twice what it is for BlueGene/L.
- The overhead for the rack for power and cooling will be the same percentage as in the Blue Gene/L.
- For this sizing, we will ignore what all this means in terms of system topology.

6.4 Heterogeneous Scaling Assumptions

For the time being, we will focus on heterogeneous systems utilizing GPUs for their accelerated performance. Since the leading chip for each GPU family tends to have a high transistor count, large area, and high power dissipation, the assumptions below are similar in nature to those of the heavyweight case. It is assumed that these accelerators will remain on daughter cards.

- There will be at most a 50% growth in shaders per core, and a doubling of FMAs per shader.
- The conventional microprocessors that provide the noncomputational support for each node will evolve as in the heavyweight case.
- The power dissipation of the GPU will grow as for the heavyweight processor.
- Local memory for individual SMs will double every few years, and the L2 and above caches will grow so as to use the same amount of space they do now.
- The number of memory ports will be pin limited as for the heavyweight, and thus will change at best modestly.
- The directly attached memory to each GPU will grow in density in step with the ITRS predictions, and no faster.
- The ratio of GPU sockets to heavyweight sockets per node will remain the same as today.
- The density of nodes per rack will increase at the same rate as for heavyweight, and with the same power limitations.
- The number of racks in a full system starts with the number in today's biggest system (to validate model) and increases by 50 per year as in the other models, up to 600 at most.

Given that details of energy expenditure in different parts of a GPU chip are not yet available, only one model is projected here - the Scaled model.

7. PROJECTIONS

Using the constraints described in the last section, we have overlayed projections for all three architecture classes over the historical data, with time running out to 2024 (the end of the ITRS roadmap). There are 5 projections - for of the heavyweight and lightweight classes there are two curves, one for the "scaled" energy model and one for the "constant" as defined in Sec. 6.2 and a "scaled" model for the heterogeneous class as discussed above.

Figs. 13 and 14 give projections for multi-rack systems, where the number of racks is allowed to grow as discussed earlier, up to a maximum of 600. The flattening of the constant projections after 2016 are due to the fact that the I/O and memory power costs have taken over (in this model they don't improve on a per bit basis with technology), and once we hit a maximum power per rack and maximum number of racks per system, there is no room to support additional flop/s. Fig. 14 shows this flattening clearly. As we said earlier, such a model is "pessimistic."

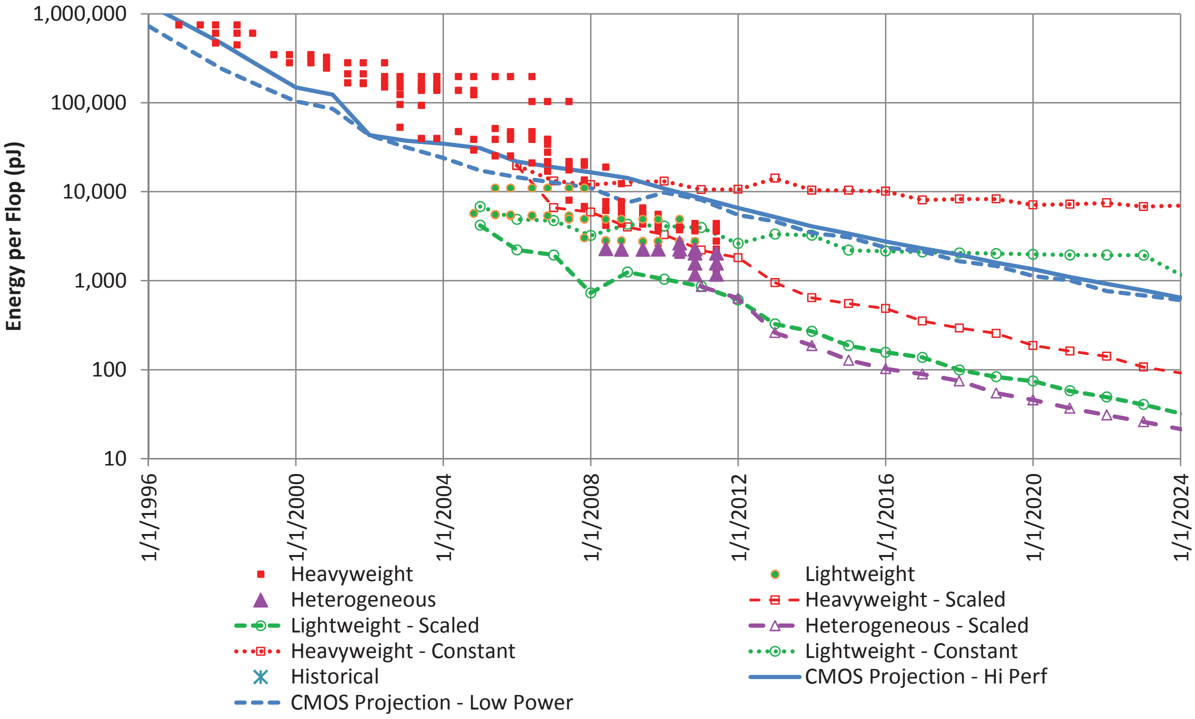


Figure 15: Energy per Flop - Historical and Projected.

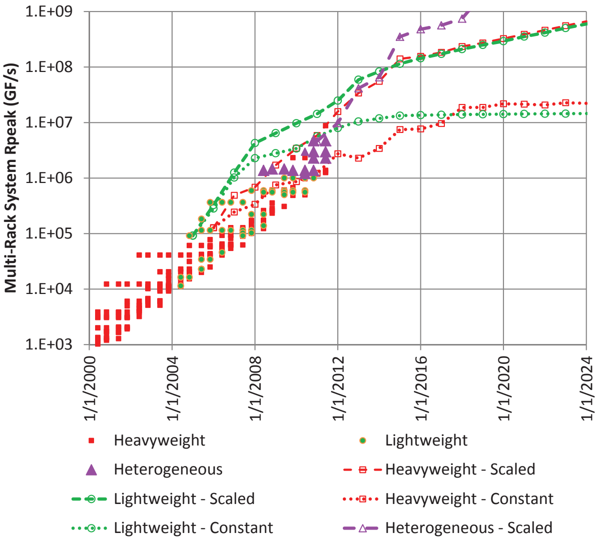


Figure 13: System R_{peak}.

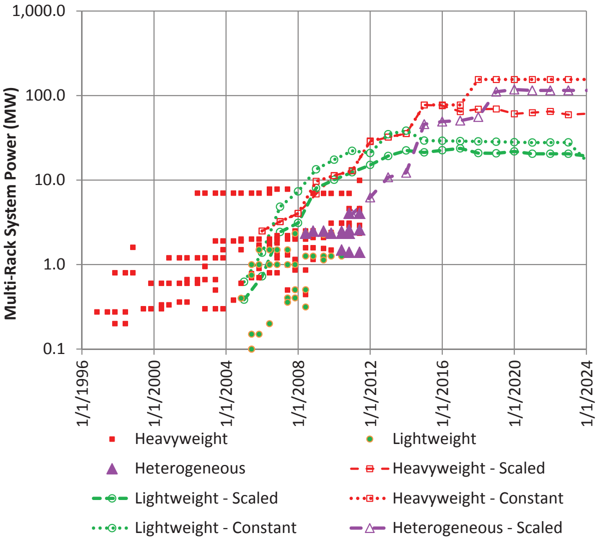


Figure 14: Peak System Power.

Fig. 15 diagrams what is probably the single biggest metric that came out of the Exascale report, namely the energy per flop (using R_{peak} here). Fig. 16 is the same graph but is limited to the region between 2004 and 2016.

It should be noted that the projections assume a technology refresh each and every year using the most up-to-date technology. Clearly, this does not occur in reality at such intervals, but as a model it does tend to bound the realm of what is possible. Also, the number of racks in a deployed system may not increase at the same rate as the model. As can be seen, by starting our extrapolation from earlier technology points, we find that the historical data does in fact tend to stay between the projected bounds, indicating that the models are reasonable, at least in the short haul.

There is also some "jitter" in the projections, especially for some of the constant energy models. This is simply due to the numerology chosen for the model[5]. In real life no

[5] For example, an anomalous dip in lightweight energy appears in 2008, which was due to assuming a switch from 2 chips per compute card to one.

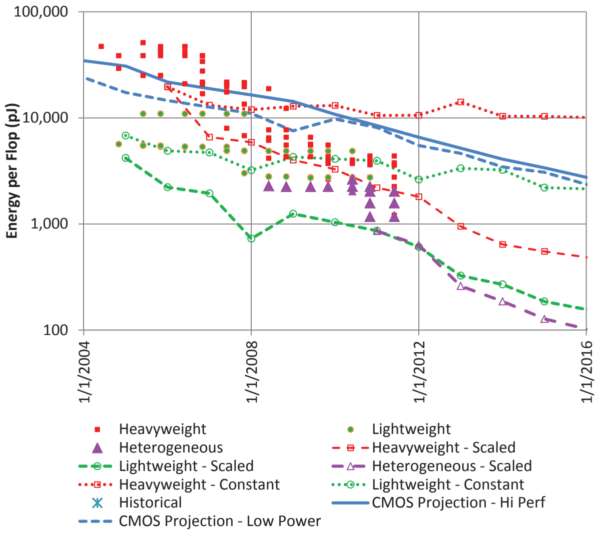


Figure 16: Focused Energy per Flop.

one would actually design a system with "slightly worse" characteristics for one year than for the previous. Again, however, the overall shape of the curves is consistent and bounds the historical data.

8. CONCLUSIONS

The historical record presented here is more complete and up to date than the data used in the Exascale report, but the overall conclusions are the same. Intrinsic capability, R_{peak}, and effective performance against LINPACK, R_{max}, have increased at a remarkably flat compound annual growth rate dating back well before the start of the TOP500 records. There was, however, a sea change starting around 2004 when the clock rate of commodity microprocessors flattened out, and raw parallelism started to take over as the driver of increased performance. Total concurrency in terms of the sheer number of FPUs that need to be directed by the program in each clock cycle has grown by more than 10X since 2004, after having taken over a decade for the prior 10X.

This same inflection point also heralded in the emergence of processor architectures that provided more such parallelism in less area and less power. These were the "lightweight" and, more recently, the "heterogeneous" designs. Both of these architectures, however, had native clock rates 1/2 to 1/4 that of leading heavyweights, meaning that parallelism must have been increased by 2 or 4X just to make up the gap and still increase overall system performance.

Internally, the only real change in Thread Level Concurrency (a property of the basic cores) has been with the emergence of GPUs where dozens of simple shader engines can be under the direction of a single instruction stream. Unfortunately, early data suggests that the efficiency with which such flops can be applied to LINPACK is perhaps around 50%, down considerably from the 70 to 80% seen for the more classical design styles. Looking forward to more complex applications where efficiencies of 10% or less are not uncommon, this raises a serious concern if GPU-driven designs will, in fact, be able to keep up.

Another major concern is the obvious decline in bytes per flop/s, again first appearing around the 2004 inflection point, and accelerating with the non-heavyweight design styles. It will be quite interesting to see if this trend continues, and if such "memory-starved" systems become limited in the classes of applications that they can handle.

In terms of the models (originally formulated in late 2007), the new data points seem to fit reasonably well between the scaled and the constant projections. For the lightweight systems we have, however, seen only two BlueGene/P sites, with the rest dating back to L. It will be interesting to see where BlueGene/Q systems lie when they come on line.

Looking forward, the same power dissipation limits that caused the inflection will continue to have an effect. Using the same amount of silicon area to increase the intrinsic computation power of a chip will likely require the basic clock rate of such chips to not just flatten, but actually decline, meaning that even more brute parallelism will be needed to make advances in high end performance. Matching this performance increase will require increased memory bandwidth, which in turn may not scale as much as logic. In fact, when we look at Fig. 14, the differences between the scaled and constant models is almost totally driven by the memory access and I/O power differences, with the flattening of peak performance due to the models running against both a max power per rack and a max number of racks.

It is also interesting to note that with these power constraints, only the heterogeneous model has a chance at a peak (not sustained) exaflop/s by 2020, but at a power of in excess of 100MW.

Finally, the need to raise energy efficiency and decrease dissipation will have a profound effect on architecture, with the trend towards lightweight and now heterogeneous systems clearly needed to move power in the right direction.

Also clear is that there is a rich design space to explore. The extrapolations made here used just one set of assumptions that often resulted internally in inefficient solutions. Future work is needed to explore these alternatives. To aid in such efforts, we intend on making our data files available on-line, and as resources allow, permit on-line changes to model assumptions to explore alternatives.

9. ACKNOWLEDGMENTS

The original version of the study discussed here was funded in part by DARPA as part of the Exascale Technology Working Group study. The update described here was funded in part by the Univ. of Notre Dame, and in part by DARPA through Sandia National Labs as part of the UHPC program. Much of the paper was written while the first author was on sabbatical at Cal Tech.

10. REFERENCES

[1] G. Almasi, S. Chatterjee, A. Gara, J. Gunnels, M. Gupta, A. Henning, J. E. Moreira, and B. Walkup. Unlocking the performance of the BlueGene/L supercomputer. In *Proceedings of the 2004 ACM/IEEE Conference on Supercomputing*, SC '04, page 57, Washington, DC, USA, 2004. IEEE Computer Society.

[2] AMD. The AMD Fusion family of GPUs. Website. http://sites.amd.com/us/fusion/apu/Pages/fusion.aspx.

[3] K. J. Barker, K. Davis, A. Hoisie, D. J. Kerbyson, M. Lang, S. Pakin, and J. C. Sancho. Entering the

petaflop era: the architecture and performance of Roadrunner. In *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, SC '08, pages 1:1–1:11, Piscataway, NJ, USA, 2008. IEEE Press.

[4] A. A. Bright, M. R. Ellavsky, A. Gara, R. A. Haring, G. V. Kopcsay, R. F. Lembach, J. A. Marcella, M. Ohmacht, and V. Salapura. Creating the BlueGene/L Supercomputer from Low-Power SoC ASICs. In *Solid-State Circuits Conference, 2005. Digest of Technical Papers. ISSCC. 2005 IEEE International*, pages 188–189, San Francisco, CA, 2005.

[5] P. Coteus, H. R. Bickford, T. M. Cipolla, P. G. Crumley, A. Gara, S. A. Hall, G. V. Kopcsay, A. P. Lanzetta, L. S. Mok, R. Rand, R. Swetz, T. Takken, P. L. Rocca, C. Marroquin, P. R. Germann, and M. J. Jeanson. Packaging the Blue Gene/L supercomputer. *IBM J. Res. & Dev.*, 49(2/3):213–248, 2005.

[6] B. Dally. Project Denver processor to usher in new era of computing. Online. http://blogs.nvidia.com/2011/01/project-denver-processor-to-usher-in-new-era-of-computing/.

[7] J. Dongarra. Trends in high performance computing: a historical overview and examination of future developments. *Circuits and Devices Magazine, IEEE*, 22(1):22 – 27, Jan.-Feb. 2006.

[8] J. J. Dongarra, P. Luszczek, and A. Petitet. The LINPACK benchmark: Past, present, and future. Online, Dec. 2001. http://www.netlib.org/utk/people/JackDongarra/PAPERS/hpl.pdf.

[9] GSIC, Tokyo Institute of Technology. Tsubame-2.0 website. http://tsubame.gsic.titech.ac.jp/en.

[10] IBM Blue Gene team. Overview of the IBM Blue Gene/P project. *IBM J. Res. & Dev.*, 52(1/2):199–220, 2008.

[11] Intel. Overview of Sandy Bridge microarchitecture. Website. http://www.intel.com/technology/architecture-silicon/2ndgen/index.htm.

[12] P. Kogge. The challenges of petascale architectures. *Computing in Science Engineering*, 11(5):10–16, Sept.-Oct. 2009.

[13] P. Kogge, K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, M. Denneau, P. Franzon, W. Harrod, J. Hiller, S. Karp, S. Keckler, D. Klein, R. Lucas, M. Richards, A. Scarpelli, S. Scott, A. Snavely, T. Sterling, R. S. Williams, and K. Yelick. Exascale computing study: Technology challenges in achieving exascale systems, 2008.

[14] R. Merritt. China taps NVIDIA for world's second biggest computer. EETimes online article, May 2010. http://www.eetimes.com/electronics-news/4199798/China-taps-Nvidia-for-world-s-second-biggest-computer.

[15] NVIDIA. NVIDIA Tesla GPUs power world's fastest supercomputer. Press release, online, Oct. 2010. http://pressroom.nvidia.com/easyir/customrel.do?easyirid=A0D622CE9F579F09&version=liveprid=678988&releasejsp=release_157.

[16] SIA. *International Technology Roadmap for Semiconductors, 2010 Update*, 2010.

[17] K. Skaugen. Petascale to Exascale: Extending Intel's HPC Commitment. Keynote Presentation at the International Supercomputing Conference, 2010. http://download.intel.com/pressroom/archive/reference/ISC_2010_Skaugen_keynote.pdf.

[18] M. Wolfe. Compilers and more: Knights Ferry versus Fermi. HPC Wire online article, Aug. 2010. http://www.hpcwire.com/features/Compilers-and-More-Knights-Ferry-v-Fermi-100051864.html.

[19] X. Yang, X. Liao, W. Xu, J. Song, Q. Hu, J. Su, L. Xiao, K. Lu, Q. Dou, J. Jiang, and C. Yang. Th-1: China's first petaflop supercomputer. *Frontiers of Computer Science in China*, 4:445–455, 2010. 10.1007/s11704-010-0383-x.