Vol.2016-HPC-155 No.32
2016/8/10

From FLOPS to BYTES: Disruptive Change towards the Post-Moore Era (Unrefereed Workshop Manuscript)

Satoshi Matsuoka[1] Hideharu Amano[2] Kengo Nakajima[3] Koji Inoue[4] Tomohiro Kudoh[3]
Naoya Maruyama[5] Kenjiro Taura[6] Takeshi Iwashita[7] Takahiro Katagiri[8]
Toshihiro Hanawa[3] Toshio Endo[1]

Abstract: Slowdown and inevitable end in exponential scaling of processor performance, the end of the so-called "Moore's Law" is predicted to occur around 2025-2030 timeframe. Because CMOS semiconductor voltage is also approaching its limits, this means that *logic transistor power will become constant*, and as a result, the system FLOPS will cease to improve, resulting in serious consequences for IT in general, especially supercomputing. Existing attempts to overcome the end of Moore's law are rather limited in their future outlook or applicability. We claim that *data-oriented parameters, such as bandwidth and capacity*, or BYTES, are the new parameters that will allow continued performance gains for periods even after computing performance or FLOPS ceases to improve, due to continued advances in storage device technologies and optics, and manufacturing technologies including 3-D packaging. Such transition *from FLOPS to BYTES* will lead to disruptive changes in the overall systems from applications, algorithms, software to architecture, as to what parameter to optimize for, in order to achieve continued performance growth over time. We are launching a new set of research efforts to investigate and devise new technologies to enable such disruptive changes from FLOPS to BYTES in the Post-Moore era, focusing on HPC, where there is extreme sensitivity to performance, and expect the results to disseminate to the rest of IT.

1. Introduction—Implications of Post-Moore Era

1.1 Overview — from FLOPS to BYTES

In 1960s the Intel Founder Gordon Moore proposed the so-called "Moore's Law", whereby the continued advances of semiconductor lithography shrinks would increase the transistor density and thus the complexity of the chips by significant percentage every year, and CPU performance of the systems would improve exponentially over time. This has been continuing for several decades, allowing supercomputing simulation performance to improve over billion-fold, facilitating computer simulation to become the third wave of science, from being able to investigate the origins of matter, to analyzing personal genomes.

However, we are starting to observe slowdown of lithography shrinks due to limitations of physics, and it is predicted that Moore's law would end around 2025-2030 timeframe. Because CMOS semiconductor voltage is also approaching its limits, this means that *logic transistor power will become constant, and as a result, their performance improvements will end*, especially in terms of performance / Watt. Assuming that we will not increase the overall power of the system, this means that system FLOPS will cease to improve, resulting in serious consequences for IT in general, especially supercomputing. In fact it is a serious situation, as much of modern societal advances have been made through quantitative advances in IT resulting from qualitative improvements due to Moore's law, resulting in stagnation of society; for example, artificial intelligence has been receiving considerable attention to automate various aspects of the society, but much of their recent advances have been due to utilization of HPC to increase their learning and processing abilities, but such advances could terminate, well before they are truly useful.

There are existing attempts to overcome the end of Moore's law, initiating a set of research area that could be labelled as *Post-Moore*. However, most of such areas have been either (1) improvements in device technologies beyond CMOS, or (2) novel compute models associated with devices such as quantum computing and neuromorphic computing. However the former research area itself has stagnated considerably over the years despite their promises, as CMOS remains strong when the actual manufacturing is considered, and the latter has become quite promising with recent advances, but their applicability is limited

[1] Global Scientific Information and Computing Center, Tokyo Institute of Technology
[2] Dept. of Information and Computer Science, Keio University
[3] Information Technology Center, The University of Tokyo
[4] Department of I&E Visionaries, Kyushu University
[5] AICS, RIKEN
[6] Dept. of Information and Communication Engineering, The University of Tokyo
[7] Information Initiative Center, Hokkaido University
[8] Information Technology Center, Nagoya University

to very narrow area of overall computing—for example, neither has the ability to be able to compute most of major HPC applications, such as PDEs and particle interactions. Moreover, they lack the system view of computing, in that they are considering very narrow aspects of an entire HPC system from application to hardware.

What we need as the kernel of Post-Moore research is to take a holistic view of the entire computing system, and identify a new parameter that replaces lithography shrinks as the metric to improve performance over time. Our claim is that *data-oriented parameters, such as bandwidth and capacity*, are the new parameters that will allow continued performance gains for periods much longer than what is predicted with Moore ' s law. That is to say, even when computing or FLOPS ceases to improve, data-oriented parameters, or BYTES will continue to improve for at least a few decades beyond, due to advances in storage device technologies and optics, manufacturing technologies including 3-D, and packaging again including 3-D and low-power optics.

Such transition *from FLOPS to BYTES* will lead to disruptive changes in the overall systems from applications, algorithms, software to architecture, as to what parameter to optimize for, in order to achieve continued performance growth over time. Our research focus is on HPC where there is extreme sensitivity to performance, and as such are the forefronts in all aspects, and its results that will make computing BYTES centric from FLOPS centric will be disseminated to other branches of IT later on.

1.2 Underlying Technology Trends Towards FLOPS to BYTES in the Post-Moore Era

In order to transition from FLOPS to BYTES, in the Post-Moore era, we must envision the device and packaging technologies that are imminent as well as being having continued growth over decades that will result in continued improvements in BYTES as the collective metric that improves over time for us to utilize for performance improvements. We then need to construct a computing system, from applications, algorithms, to software and hardware that will be able to exploit these devices, again over time for continued performance growth.

The first observation is that, although the number of logic transistors we can *turn on simultaneously* will become constant when Moore ' s Law ends, the total number of transistors can continue to grow over time, memory/storage devices in particular but the same goes for logic transistors. For the latter, they are often referred to as *dark silicon*[12], and could still increase over time by exploiting the aerial / spatial packaging, as the amount of silicon occupied in the current systems is still miniscule relative to their size. However, since only a constant number of transistors can be turned on simultaneously, in order to attain maximal power efficiency, compute units must be made extremely efficient, customized to the data types and their associated (data-centric) algorithms. This might also lead to dynamic customization through reconfigurable logic such as future versions of FPGAs, and how they would be composed (pseudo-) dynamically to construct a hardware system for a given set of target applications.

The more fundamental disruption comes from memory and storage systems in the post-Moore era. DRAM has been the dominant memory technology in the CMOS-Moore era, since the 1980s. However, as Moore ' s law ends, they are faced with two problems that will prevent their capacity from growing further, due to their nature being capacitance memory devices. One is that, they require certain feature size to hold sufficient capacitance to hold the charges stably without causing significant number of errors that become uncorrectable. This phenomenon has been highlighted by recent works such as row-hammering[33] where crosstalk between the DRAM lines causes massive multi-bit failures with lithography shrinks. In fact, some research indicates that DRAM feature scaling is reaching its limits before lithography shrinks terminate for logic. By going 3-D stacking this imitation can be overcome, but since DRAM requires refresh which will require constant charge energy per memory transistor cell, their power will increase linear to capacity.

To overcome these limitations, there has been a plethora of proposed new memory devices, especially non-volatile memory or NVMs, such as flash, STT (Spin Torque T)-RAM, PC(Phase Change) RAM and Re(Resistive)-RAM. They all share the common characteristics that their static energy is essentially zero, and their read performance is competitive or even faster than DRAM. As such, they are subject to extremely dense stacking, either by dense VIAs or by direct stacking structurally in the device, without increasing power. Their problem is with writes, where more energy is deemed to be required due to persistent change in the device state. There are several potential solutions to this problem proposed however, including the combined use of SRAM, whose static energy can be minimized by minimizing leakage with FinFETs, to voltage-controlled NVMs where energy requirement for device state change could be extremely minimized.

Local memory bandwidth can be increased significantly with extreme 3-D stacking, such as those using tungsten VIAs instead of traditional copper, or non-VIA interconnect technologies such as inductive or capacitance coupling across chip layers[27]. For example, with 100,000 VIAs across chips instead of the 1,000 or so enabled by today ' s Wide I/O specifications, one could implement massive memory bandwidth reaching 10s of Terabytes/s attaining massively high BYTES/FLOPS ratio at very low energy, due to minimal distance required to move data in the Z-direction, with capacity raging to multi Terabytes as well.

Of course, data locality will still be of importance even in this case, as moving the data in X-Y direction would involve high latency and more energy; as such, algorithms and software systems to exploit data locality would still be of significant importance.

Communication going off-chip, for short distances it may still be advantageous, both cost and energy-wise, to resort to short distance electrical signaling such as silicon interposers, as we are starting to see in today ' s high performance chips such as GPUs and Xeon Phis. However, their range is in centimeters, and any large HPC systems one would immediately have to resort to optics, as they have favorable characteristics that, once Electro-optical conversion is done, power is constant over distance, which is not the case for copper. However, LAN-level interconnects, as represented by today ' s Infiniband and 100GigE, has slow relative growth, and much slower to long-range carrier optics that easily reaches multi Terabits/s. This is because the latter uses various

modulation technologies such as DWDM as well as amplitude modulation. The reason these are not being used in LAN is their expense, power, area as well as cost. However, in the future they are expected become part of short- to medium range optics, realizing Terabits on a single fiber, sufficient to interconnect chips with massive memory bandwidth as mentioned above at massive scale, at constant power per chip. Such network will also have to be combined with ultra low-latency optical circuit switching, as optical packet switching will be difficult for foreseeable time. Although good for bulk-transfer, circuit switching will present a whole sets of problems especially in terms of latency.

There are other technologies that are complimentary and could be part of the overall architecture. Certainly the specialized architectures for new computing models such as quantum computing, neuromorphic computing etc. can be used as accelerators. Advanced cooling and packaging technologies could be employed, such as direct liquid cooling can be employed to volumetrically cool 3-D chips; optical high bandwidth circuit switching can be combined with conventional low-power, low-latency electronic packet switching and used on a case-by-case basis. Packaging of the nodes can be made extremely dense to optimize for power, cost, and latency, etc. with aggressive cooling, using extremely dense embedded technology as we see in today ' s smartphones, rather than sparse packaging we observe in today ' s servers. While becoming somewhat difficult to repair on a component-by-component basis due to their complexity and wiring, rather components are expected to degrade and fail-in-place design with sufficient redundancy to maintain performance, as we have demonstrated in our recent work [8].

1.3 Our Approach towards the Post-Moore Era — From FLOPS to BYTES

Putting all the technological elements together, a Post-Moore supercomputer would embody immense capacity and bandwidth, orders of magnitude above what we observe in today ' s machines, while being extremely power efficient and dense, immersively cooled with fail-in-place design. The question is, how we put all the elements together to formulate a HPC architecture, so that they can be utilized as a real architecture that will continue to evolve in performance over time until the BYTES technology faces its inevitable end, much further in time however compared to that of FLOPS. For this, our group is proposing a comprehensive and disruptive research program, from new architectures that effectively compose the elements, to algorithms and applications that exploit the immense BYTES, as well as software elements to allow effective control of the hardware as well as providing appropriate abstractions so as to be able to make the machines effectively programmable.

For algorithms and applications, we will investigate the library of applied mathematics and algorithms, to identify those that can be effectively accelerate and evolve over time with BYTES as the parameter. Although such parameters will increase over time, overall their hierarchy will become deeper as they scale, due to latency improvements being stagnant, as such physical locality being still important. This largely implies algorithms that exploit data motion rather than increasing the computing density, i.e. what could be considered as resurrection of implicit methods. By all means the algorithmic details will be different because of vastly different latency vs. bandwidth ratio, as well as the requirement to save energy.

In order to allow programming of such algorithms and applications in a Post-Moore architecture, proper abstractions must be provided for the hierarchical memory elements of the architecture, as well as being able to specialize and configure the compute part of the system according to the datatypes and the associated processing. Moreover, resiliency of the system needs to be handled in an automated way, with assumptions that various components of a given system will fail in place over time. Also, intra-node communication needs proper abstraction to cope with hybrid optical circuit switching and electronic packet switching.

As for the hardware architecture, the memory and compute components should be reconfigurable so that maximum BYTES can be achieved while the compute will be customized to the given problem. Such customization has been achieved mostly in embedded systems, where scaling nor massive bandwidth has been the major subject matter. State-of-the-art optics is the major component that must be integrated into the system, and scalable control given massive multi-terabit bandwidth must be achieved. The entire architecture must be properly controlled to optimize for power; although power control of HPC systems have been one of the major research subjects, there will be significant challenges because of the disruptive changes in the hardware, the metrics in BYTES, as well as increasing process variations. [*1]

2. Post-Moore High-Performance Architecture

2.1 A Super-hub Chip for Connecting Various Types of Post-Moore Architectures

2.1.1 Motivation

In the CMOS era, a digital system is formed with combination of CMOS chips, thus System-on-a-Chip which integrates everything into a chip or 3D or 2.5D integration using Through-Silicon-Via (TSV)s or silicon interposer are major implementation methodology. However, in the post-Moore era, chips are not implemented on a single technology. Thus, various types of devices each of which has different electric characteristics must be connected as a single system. The key technique in the post-Moore era is not for individual chips but for a methodology to integrate various heterogeneous devices into a system.

We have been focused on wireless inductive coupling Through-Chips Interface (TCI)[36] which uses magnetic field generated between a pair of transmitter and receiver coils. It can achieve more than 8 Gbps serial data transfer with extremely low error rate ($< 10^{-9}$) and reasonable energy consumption (tens of mW per channel). TCI can be used to connect any devices which work at various electrical characteristics only if it can implement wires to form the coil and SERDES. Intellectual Property (IP) for a CMOS process has been already available. Our proposal is to build a super-hub chip which can connect various future devices and form an integrated digital system.

[*1] This is an extended version of [25]

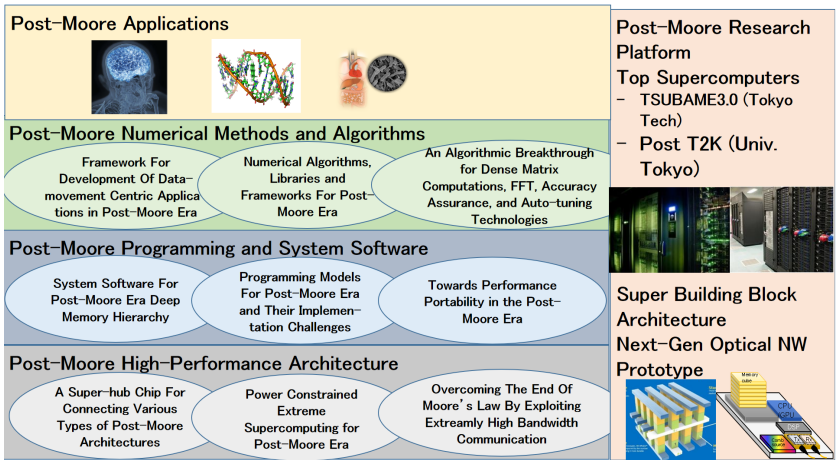


Fig. 1 An overview of research topics towards the Post-Moore era

2.1.2 A super-hub chip

Figure 2 shows the concept of the super-hub chip. It is a mixture of FPGA, tiny embedded CPU, and powerful reconfigurable switching fabrics with TCI interfaces. It can be used as an active interposer and a lot of daughter chips can be connected through the TCI interfaces. No signals except a system clock and reset are connected from outside the super-hub chip. All data are transferred in serial packets, and they are transferred directly through the combination of embedded reconfigurable switches. Fine-grained reconfigurable logics like FPGA are provided for protocol translation and offloaded operations which can be efficiently executed near switches.

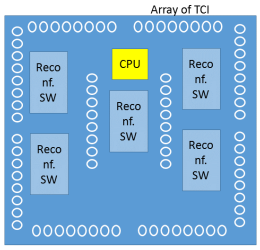


Fig. 2 Concept of the super-hub chip

In order to connect other super-hub chips located on the distant place, optical interconnect board described in the later section is used as shown in Figure 3. Since TCI is electrically contactless, various types of chips including neural chips and brain chips can be connected. Also, novel non-volatile memory modules are placed on the super-hub chip as well as traditional CPUs and accelerators. Since TCI can form the bus system[21], 3-D memory system in which all chips are connected with the TCI can be directly connected.

In the current art of technology, the power is supplied from backside of the chip by using TSV. However, our final goal is to deliver the power also through the TCI, then we can replace and add chips freely on the super-hub chip.

There are a lot of challenges for realizing such a system. The most difficult problem is an operating system running on the embedded CPU which manages the connection between daughter chips and the super-hub chip. Automatic adjustment mechanism between chips must be developed. Handling various types or speed of packets transferred between various types of chips is another important problem. We need some standardization, but it is difficult to use only a single packet format. The protocol translation will be needed on the FPGA part. The architecture of the powerful reconfigurable switches is still open problem, although we are trying to develop a prototype by using a traditional FPGA[41].

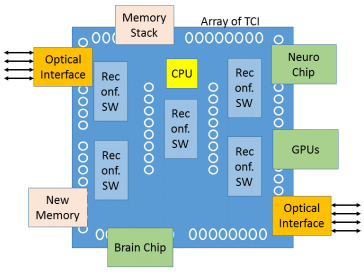


Fig. 3 Mounting daughter chips on the super-hub chip

2.2 Power Constrained Extreme Supercomputing in the Post-Moore Era

2.2.1 Motivation

Power is one of the most critical concerns on state-of-the-art supercomputing. For instance, US department of Energy has summarized that energy efficient designs are required to achieve exascale performance under power envelop of 20 MW, and it has been indicated that the effective performance is clearly limited by the power constraint [1], [4]. Unfortunately, the end of Moore's law makes the power issue more serious. Although the end of Dennard Scaling has brought difficulty in improving performance without affecting power consumption, researchers have found that integrating a large number of computing cores with slow clock frequency and low supply voltage can overcome the issue. However, in the Post-Moore era, such sophisticated design

strategy does not scale any more.

Power constrained extreme supercomputing is a key technology in order to overcome the unavoidable power issue. Hardware overprovisioning, in which the system designers install hardware resources that can exceed a given power limitation, is a promising approach to bring a breakthrough for power-efficient large-scale computing. A power management software applies power capping in order to ensure that the system does not violate the given power constraint at run time. Since such overprovisioned system can adaptively optimize power-performance tradeoff by considering dynamic behavior of software workloads, it can translate the given power resource into performance effectively. Although such hardware overprovisioning is a key area in HPC community, unfortunately, available techniques are not enough due to its restricted optimization space, e.g. considering only CPU and DARM devices, optimizing only clock frequency and supply voltage, and so on. *Power constrained BYTE-oriented supercomputing* is a new challenge to overcome such limitation, and our current focus for specific research topics are 1) revisiting HPC computing model and 2) accepting extreme heterogeneity in power management.

2.2.2 Revisiting HPC Computing

In HPC applications, maintaining calculation precision such as floating-point operations is essential. Since emerging scientific applications tend to require tremendous amount of computations, system designers have so far been forced to sustain performance improvement in terms of computing ability, i.e. *FLOPS*. However, two strong approaches to gain computing performance, i.e. increasing clock frequency and enlarging computing-core counts, do not work any more in the Post-Moore era. Therefore, we need to consider another new direction to achieve high-performance and low-power at the same time.

One of the most promising approaches is to aggressively reduce the amount of computation required to complete HPC executions, and the concept of *approximate computing* is attractive. If it is not so severe to maintain the output precision, we can relax the computing accuracy, resulting in reduced amount of computation required to complete its execution. In computer architecture community, resent researches demonstrated that approximation has a great potential to improve power-performance efficiency. Such kind of relaxation makes it possible to implement *BYTE-oriented Acceleration* such as neural-acceleration [11] and memoization. Since current researches mainly target vision computing, in which highly accurate computing is not always required, considering *approximate supercomputing* is a new challenge.

2.2.3 Accepting Extreme Heterogeneity in Power Management

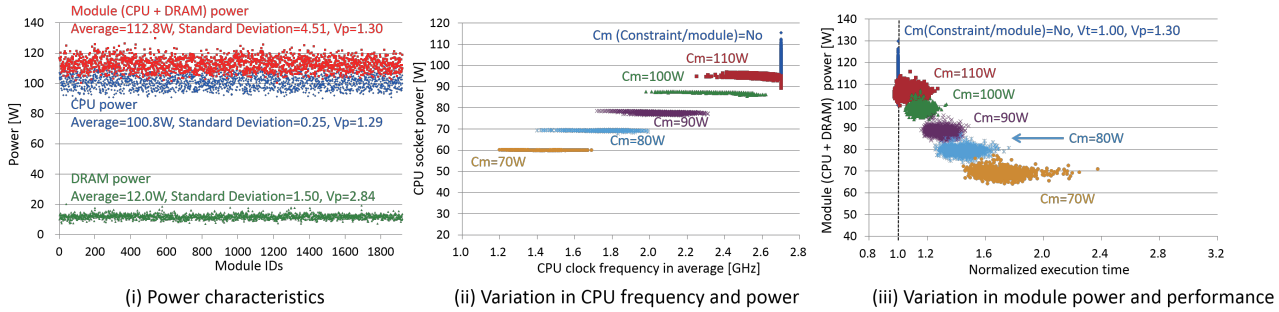
Power management is an essential function on power constrained supercomputers, and the current researches mainly focus on the management of CPU and DRAM devices. In the Post-Moore era, however, it is predicted that the system includes various devices such as GPUs, FPGAs, VMRAMs, optical devices, and so on. Also, as demonstrated in [20], manufacturing variation seriously affects effective performance. Figure 4 shows power-performance characteristics of a power-constrained large-scale supercomputer. The machine includes 1,920 modules, each of which consists of an Ivy Bridge CPU and 128 GB DRAM, and **DGEMM* from HPC Challenge benchmark suite has been executed as a representative benchmark. For the detail, please refer the paper [20]. Figure 4 (i) reports power consumption measured via RAPL (Intel's Running Average Power Limit) without any power constraint, and around 30% of difference in power consumption across modules is observed. Unfortunately, such power inhomogeneity is translated to the CPU clock frequency variation as shown in Figure 4 (ii) when we apply uniform power capping, and it causes module-level load-imbalancing as demonstrated in Figure 4 (iii), resulting in poor system performance. So, it is required to consider device-level (such as manufacturing variation) and function-level (like CPU, DRAM, GPU, FPGA, NVRAM) heterogeneity at the same time. This issue has to be addressed on power-constrained extreme supercomputing in the Post-Moore era.

2.3 Overcoming the End of Moore's Law by Exploiting Extremely High Bandwidth Communication

To overcome the end of Moore's law, the design of HPC systems must be re-designed from scratch. Since we can no longer expect performance of general CPUs to improve, special purpose processing cores, such as GPUs, FPGAs, neuromorphic chips and ising chips must be used, which have superior performance in specialized computations. In current computing systems, so called data affinity scheduling has been implemented to process data without moving whenever possible in order to overcome the difficulties with moving data between computing nodes. However, when special purpose cores are used, data have to be moved according to the required processing at each stage of a job. To support such a computing paradigm, which we call function affinity scheduling, improvement of inter-node communication performance coupled with reduced energy cost is critical.

On conventional HPC systems, the inter-node I/O bandwidth is about 1/10 of the intra-node memory access bandwidth. Breakthroughs in performance for memory access bandwidth may be brought about by the 2.5D or 3D packaging of processor with embedded memory. However, breakthroughs in performance for the I/O bandwidth by the optical interconnect technology is much more dramatic. Even though the bandwidth of a single optical channel is at most 25 100Gbps, polarization division multiplexing and wavelength division multiplexing (WDM) allow us to bundle tens of channels without complicated cabling. For future HPC systems, dense-WDM (DWDM) is most promising way to create interconnections with bandwidth on the order of tens of Tbps that will fill the performance gap between inter- and intra-node memory access.

Issues in implementing DWDM in HPC systems are size, cost and power consumption of DWDM light sources. Each wavelength requires a light source with temperature control and complex structural elements to achieve precise wavelength control for DWDM. Therefore, it is not realistic to implement DWDM light sources in each computing node. To address this problem, National Institute of Advanced Industrial Science and Technology (AIST) has proposed a wavelength distribution system using wavelength bank (WB) or optical comb source and a silicon pho-



(i) Power characteristics (ii) Variation in CPU frequency and power (iii) Variation in module power and performance

Fig. 4 Impact of Manufacturing Variation on Performance for Power Constrained Supercomputing [20]

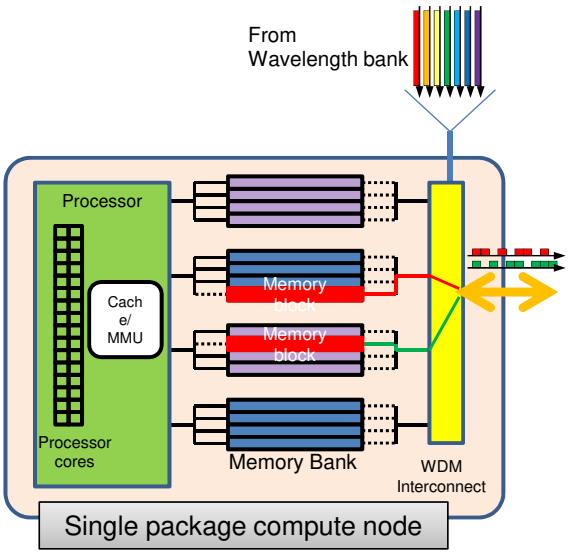


Fig. 5 Wavelength Bank and computing node

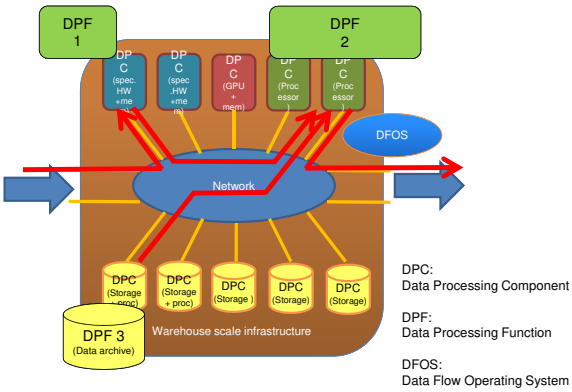


Fig. 6 A slice on a pool of resources

tonics modulator. WB is a centralized generator of wavelengths for DWDM. The wavelengths are distributed to computing nodes through optical amplifiers, thus eliminating the need for light sources at each computing node (Fig. 5). The distributed light is de-multiplexed to individual wavelengths, modulated, multiplexed again, and transmitted from each computing node. Silicon photonics optical circuits can be used for whole light wave processing, including modulation, at a computing node. Therefore, size, cost and power consumption can be quite small, and hybrid implementation with electrical circuits is easy. Up to 50Gbps/channel (i.e. wavelength) modulation may be possible by future silicon photonics modulators, and approximately 200 channels can be in a fiber through the use of different wavelengths and polarizations. Thus a total of 10Tbps bandwidth can be realized in a single fiber. This bandwidth is comparable to memory access bandwidth of a future single package processor-memory node. In addition to using electrical switches which use WB in the same way, such high bandwidth DWDM signals can be switched in one bundle by fiber cross connect switches, or can be switched individually by wavelength selective switches.

There are many issues to efficiently utilize a HPC system with such a high bandwidth communication. Future HPC systems will be a pool of specialized cores and modules with a high bandwidth optical network connecting the components. Since there will be many different kinds of special purpose cores and the cores required may differ from job to job, we propose to implement dynamic ”slices”. A slice will be dynamically composed and provided to each job as shown in Fig. 6. A slice will be a set of dynamically assembled cores, memory modules, and high bandwidth communication paths required for each job. To realize such a system, software architecture including operating systems, programming models and memory systems must be re-designed as well.

3. Post-Moore Programming and System Software

3.1 System Software for Post-Moore Era Deep Memory Hierarchy

System software and/or middleware have roles to be bridges between application software and architecture, which will continue to be changed. In the Post-Moore era, architecture will have even more complex structure with the BYTES-centric technology; heterogeneous memory hierarchy that will be deeper including 3D-stacked memory and next generational non-volatile memory, which are accessed by heterogeneous and specialized processor core. On such architecture, while the access bandwidth will be improved, larger access latency will become larger costs for application software.

Against the situation where architecture becomes specialized and divergent, it will be necessary to systemize how programming languages and libraries are mapped to underlying architecture, and realize the techniques as working system software. More concretely, research topics on system software towards the Post-moore era include:

- Intra-node/inter-node resource scheduling that supports deep

memory hierarchy efficiently
- Fault tolerance that harnesses non-volatile memory with high-bandwidth, large-capacity and high-latency
- Extremely high-performance I/O with non-volatile memory with high-bandwidth, large-capacity
- Interaction with programming models for deep memory hierarchy
- BYTES-centric performance modeling

Considering the above directions, we have started several research projects as follows. Deeper memory hierarchy becomes one of obstacles for application programmers, especially when they have existing software written in traditional styles. However, there are already deeper memory hierarchy on HPC systems with many-core accelerators, that embody device memory, host memory and memory class Flash device. In order to realize fairly efficient usage of these memory types from application software written in MPI and CUDA, a middleware called Hybrid Heterogenous RunTime (HHRT) has been developed[9]. HHRT, which is a wrapper library of MPI and CUDA, provides dynamic management mechanisms of mapping between user-allocated data and heterogeneous memory resources, by *transparent swapping* that considers memory hierarchy among device memory, host memory and Flash devices. We have observed that co-design among middleware and application algorithm layer is important to harness high bandwidth of memory in upper layer and large capacity of lower layer[10].

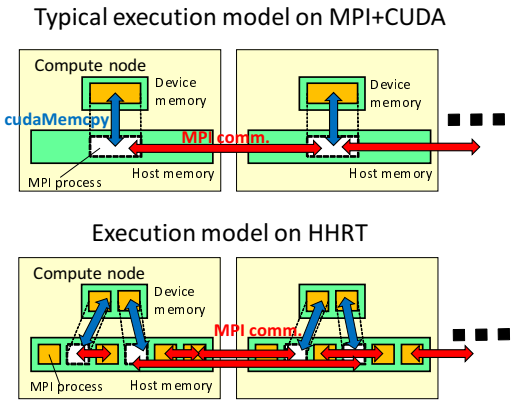


Fig. 7 Execution model of HHRT. Device memory and host memory layers are shown

Achieving fault tolerance on HPC systems with checkpointing is a important topic that has a long history. Architecture change towards the Post-Moore era causes both challenges and opportunities; while deeper memory hierarchy makes the structure of user-allocated data more complex and larger, the existence of non-volatile memory offers new optimization methods to store checkpointing data. Based on this observation, we have proposed *a checkpointing strategy that harnesses burst buffers*[32]. Here a batch of Flash devices are treated as a single burst buffer, which fills the performance gap between node-local memory/storage and parallel file systems. The resultant checkpointing system is more reliable than that with node local storage and provides higher performance than that based on parallel file systems.

Towards the Post-Moore era, research activities including the above mentioned projects should be expanded in order to support more complex memory/storage hierarchy/networks.

3.2 Towards Performance Portability in the Post-Moore Era

Designing high performance applications for the current petascale architectures is a difficult problem, and will be more difficult as we approach the post-Moore era. The shift of performance characteristics in the post-Moore era requires rethinking of algorithmic tradeoffs in HPC application kernels. Although a significant effort has been undertaken to optimize application kernels for the flops-rich, bandwidth-poor performance spectrum, such kernels can perform suboptimal as the increase of transistor density is gradually slowed down while memory performance is still expected to improve with the emergence of stacked memories. As such, redesigning algorithms that are robust against such shift towards the bandwidth-rich architectures will be crucial, yet, at the same time, substantial effort will be required to rewrite application implementations since the current state of practice in application development is not likely to allow for such algorithmic changes to be done automatically. Furthermore, while specialization of architectures to applications with reconfigurable fabric is considered to be one of the promising directions to mitigate the slowing-down processing performance, such a trend of widening architectural diversity will in turn cause significant difficulties in developing applications that are performant across diverse architectures.

Realizing performance portability in the post-Moore era requires a concerted effort that spans the whole stack of software and hardware. In particular, effectively exploiting the capability of architecture specialization realized with, e.g., FPGAs in scientific computing and data analytics poses both recurring and new challenges. Although the recent development of standardized programming interfaces for FPGAs such as OpenCL is certainly a big step towards wider adoption in various computing domains rather than a small niche market, realizing high performance is a completely different problem that is not yet well understood for many of application kernels. Furthermore, even if it is feasible to achieve high performance with OpenCL, its optimization techniques would be most likely very different from what are employed for other accelerators such as GPUs, resulting in the performance portability problem.

A lot of research has been done in performance analyses, modeling, and optimizations for the past and current generations of high performance computing systems, however, it is still unclear whether such existing methods are applicable to systems with reconfigurable fabric since most of them assume that the underlying architecture is fixed with some known performance profiles such as peak FLOP/s and bandwidths. While case studies of FPGAs with scientific applications were conducted in the past, there has been little work that comprehensively covers a variety of computation patterns with detailed performance comparison with CPUs and GPUs. The limited studies of FPGA performance for scientific computing is partially caused by the fact that past FPGAs did not have hardened FPUs, making it difficult to compete with other accelerators such as GPUs. The lack of standardized programming interfaces such as OpenCL was also another major hurdle

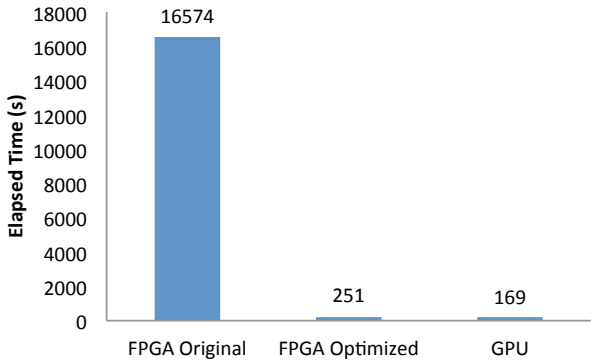


Fig. 8 Performance comparison of FPGA and GPU with Rodinia Needleman Wunsch benchmark.

for FPGA performance studies.

To realize performance portability on FPGAs, understanding its performance characteristics for a variety of HPC kernels is the most pressing issue. The availability of vendor-supported OpenCL compilers enables to quickly run existing OpenCL-based benchmarks, however, just running those kernels on FPGAs does not necessarily allow us to expose relevant performance characteristics. Rather, it is likely to be necessary to investigate how to exploit FPGA-specific features from OpenCL to better understand the potential performance of FPGAs.

As preliminary work, we have evaluated a few kernels in the Rodinia benchmark suite on an Altera Stratix V FPGA using Altera's OpenCL compiler. The benchmark suite provides a wide range of parallel computing motifs implemented in OpenMP for multi-core CPUs, and CUDA and OpenCL for GPUs. We first evaluated the performance of the OpenCL versions without minimal changes for using the Altera Compiler, and also developed a completely new kernel with FPGA-specific optimizations such as sliding-window based efficient pipelining. Figure 8 shows their performances of the Needleman-Wunsh benchmark with that on an NVIDIA Tesla K20C GPU. As shown in the figure, the original OpenCL version, which is primarily developed for GPUs and uses OpenCL local memory for efficient vector processing, performs significantly slower than the GPU. The reason is that while the multi-threaded execution model of OpenCL can be efficiently pipelined on FPGAs, using barriers within an OpenCL workgroup requires a complete pipeline flush, resulting in extremely inefficient pipeline usage. The optimized FPGA version solves this problem by reorganizing the computation using the compiler-based loop pipelining with a sliding window, achieving the competitive performance as the GPU. For more elaborated analyses and results, refer to [42].

The above experiment indicates that while FPGAs are indeed promising for the future post-Moore era, FPGA-specific optimizations are highly important to effectively exploit the potential capability of the reconfigurability. However, since expressing such optimizations in OpenCL requires substantially different programming than those for GPUs, no single code performs equally efficiently on both GPUs and FPGAs. To address this performance portability issue, our future work will explore higher levels of programming abstractions such as OpenMP as well as domain-specific languages.

3.3 Programming Models for Post-Moore Era and Their Implementation Challenges

How we program post-Moore era machines? As is the case in virtually all new generation machines, it is a question of utmost importance. It is also a question to which giving a single definitive answer is impractical or perhaps impossible. In the following subsections, we review how we program HPC machines today and look into the possibility of changing it, with emerging communication hardware currently on the horizon and hopefully matured in post-Moore era.

3.3.1 High performance programming as we practice today

Developing highly scalable programs on today's HPC machines is already a daunting task; it often requires so-called nested programming, in which intra-node and inter-node parallelism are expressed and managed in distinct programming interfaces—typically MPI for inter-node parallelism and OpenMP (+ CUDA) for intra-node—with different sets of performance considerations and pitfalls. In particular, inter-node programming typically puts many burdens directly on the programmer, including load balancing, data partitioning, communication aggregation, and overlapping communication with computation. They are of course concerns in a single node shared memory environment too, but only to a lesser extent, as the programmer typically does not have to specify placement of tasks and data in every detail to accomplish a reasonable performance. In shared memory environments, programmers frequently confront a distinct set of performance problems instead, stemming from implicit (or sometimes unintended) communication among processors and memory, such as false sharing and memory hot spots.

Nested programming has long been taken for granted; in particular, carefully aggregating inter-node communication and partitioning tasks/data are all considered essential in order to maintain communication performance to an acceptable level. This stems from the fact—or the perception—that inter-node communication, especially fine-grain communication, is inherently inefficient and thus should be minimized at all cost.

We believe emerging communication technologies can potentially make a sea change in this landscape, and this is where HPC machines can be made significantly "better" both in terms of programmability and performance delivered to applications (if not necessarily peak performance) in post-Moore era, when we can no longer rely on continued increases in transistor densities. Of course, this can happen only with co-design efforts between software and hardware.

To put it into perspective, Figure 9 shows typical latencies and bandwidths we observe in today's HPC machines. They obviously make a deep and continuous hierarchy. An important for our discussion here is that the latency gap within a compute node is already quite wide—two orders of magnitude increase from L1 cache (a few cycles) to off chip ($\approx$ a few hundreds cycles, depending on the coherent traffic it caused). In comparison, the gap between the longest intra-node latency and the latency to the next node ($\approx$ 1-2 μs) is not very significant. A similar observation can be made about bandwidth; the gap between a typical intra-node main memory bandwidth ($\approx$ a few hundreds GB/sec) and an inter-node bandwidth ($\approx$ 10+ GB/sec) is an order of magnitude, which

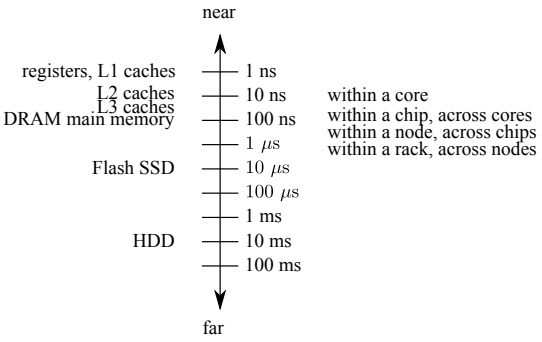


Fig. 9 A typical latency hierarchy in today's HPC systems

is in the same ballpark of the gap between caches and main memory. These continuities across a range of hierarchy may suggest that, even today, it is worth reconsidering nested programming styles we are practicing today.

3.3.2 A Programming Model in Our Sight

We do not anticipate—nor advocate—that a programming model for post-Moore era should be discontinuously different from what we see today, at least on the surface. We believe the key progress is a proliferation of programming models supporting global view of data and computation, as well as those supporting hierarchical decomposition of data and tasks. Programmers are asked to formulate their algorithms in such a manner that a problem is hierarchically decomposed into sub-problems, maintaining a good compute-over-data ratio; data are placed in a global address space and tasks automatically load balanced, both within and across nodes (global load balancing). Load balancing could be done either statically or dynamically, but in order to be general enough to support irregular and dynamic problems, dynamic load balancing will play a significant role. Tasks are also used for hiding latencies of remote data accesses. Data partitioning will be determined in part by the programmer (e.g., initial distribution), but the system will also migrate/replicate data dynamically according to the task placements resulting from dynamic load balancing.

In any event the resulting communication is not likely to be as optimized as manually optimized task/data partitioning and aggregation. Yet with emerging communication technologies, we have a good prospect for efficiently supporting such communication patterns.

3.3.3 Integrated Optics and Electronics:

A promising communication technology is silicon photonics, optical circuits tightly and densely integrated with electronics (CPU or GPU).

Some reasons why we cannot efficiently support global view programming across nodes in today's hardware are (1) large overheads for small messages and (2) an intra-node bottleneck when many threads access the network. As discussed earlier, the latency between a node to the next is nowadays in the range of 1-2 μs, which is only a couple times larger than the latency between chips in a single compute node. Yet, a significant difference between the two is that the former has a much larger *overhead*—host processor cycles devoted to initiating communication and thus cannot be overlapped by computation on the host processor. Even with recent network interface controllers supporting kernel-bypassing accesses, the overhead for each message, due to setting up addresses, sizes, and strides, posting the command to a command queue, and checking its completion status, is still significant for small messages, compared to that of intra-node communication, which is essentially a mere memory access instruction. A photonics tightly integrated with processors can potentially simplify the protocol to access network interface significantly and deeply integrate outstanding communications into processor pipeline.

3.3.3.1 Optical Circuit Switching (OCS) networks:

As discussed in Section 2.3, future optical networks are expected to deliver dozen Tbps on a single fiber with wavelength division multiplexing, two orders of magnitude larger bandwidth than what we have today. This gives a good prospect for implementing very large scale machines with high bisection bandwidths. A particularly relevant for global view programming models is that such high bisection bandwidth networks may be able to deliver a robust performance for traffic patterns of automatically and dynamically load-balanced workloads, that are not as localized as manually tuned local balancing and data partitioning.

It is unlikely that such extremely high bandwidth networks are packet-switched; instead, they are likely to be circuit-switched networks in which there are only a limited number of high bandwidth paths at a time. An optical path can be reconfigured dynamically, but such reconfiguration takes time; large switches having many (e.g., > 100) ports are typically implemented with 3D MEMS and their typical reconfiguration time is dozens of milliseconds, or about four orders of magnitude longer than a single hop latency; switches achieving a much smaller reconfiguration time ($\approx 10\mu$s) with 2D MEMS technology have been demonstrated [15], but they tend to allow only a small number of ports.

How to design large scale machines with such circuit-switched networks [6], [7], [37] and how to effectively use them for HPC applications are still unsettled questions [2], [31], [39], but in any conceivable design, extreme high bandwidths are available only to a limited number of node pairs at any given time.

Past researches [5] show that many MPI applications enjoy a low communication degree per node, so a majority of communication can be covered by optical circuits. It is yet to be demonstrated whether machines equipped with circuit switched networks can *transparently* improve diverse workloads, including ones employing less localized communication patterns. Still more challenging is how to effectively use OCS networks for application written in global view programming models using global address spaces and automatically and dynamically balanced tasks; such programs tend to generate communication patterns challenging for OCS networks—less regular and and more fine-grain traffic.

While challenging, we see this problem as a great research agenda that requires concerted efforts across layers. On the programming interface level, programming languages can support aggressive batch prefetching and replication for read mostly data, which can be mapped onto OCS in a relatively straightforward manner. For frequently update data, while randomized dynamic

load balancing tends to generate literally random traffic, load balancing strategy could itself be adapted to currently available optical circuits, combined with aggressive piggybacking strategy to move data likely to be accessed by migrating tasks.

4. Post-Moore Numerical Methods and Algorithms

4.1 Overview

4.1.1 Background

Although it is predicted that the Moore's law will end in 2025 - 2030, FLOPS of (super) computers will grow for about 10 years. Because the increase in FLOPS are mainly owing to the increase in parallelism provided by many cores and special instruction sets. Consequently, we have to take into consideration the fact that the numerical algorithm in post Moore era should also utilize the massive parallelism. At least $O(10^3)$ threads and $O(10^5)$ computational nodes are effectively utilized in the algorithm.

When the Moore's law ends, we, the researchers on algorithms in practical analyses, will face a turning point. Although it is apparent that FLOPS in a single chip is no longer improved, it is hard to forecast the major architecture for computers and processors. However, there is an observation that "BYTES" will still grow. For example, three dimensional stacking and silicon photonics technologies contribute the increase in bandwidth between memory and processors, and between computational nodes. Moreover, to reduce the power, non-volatile memory will be more utilized in computers. On the other hand, to effectively utilize the benefits of the growth of BYTES provided by these technologies in practical analyses, the computational paradigm and the numerical algorithm should be changed. Although in the last decade, the more flops and less bytes algorithms (*compute intensity*) have been preferable, we should pay attention to the algorithm with more bytes but less flops (*data movement intensity*). However, it is not straightforward to efficiently utilize the new technologies in the algorithm. For future algorithms we should consider complex and deep memory hierarchy, heterogeneity of memory latencies, and efficient use of logical units attached to memory modules etc. For example, we should intensively investigate the bandwidth and latency reducing algorithms, in which the lower layer with higher memory bandwidth is more utilized or the global synchronizations and communications are reduced.

In the real world, the flops per watt is very important for the development of various IT and related social activities rather than the flops itself. Even if the Moore's law ends and the number of transistors which can be operated by fixed power is no more increased, the flops per watt has spaces to grow. In some specified applications or computational kernels, special instructions like SIMD, accelerators, and reconfigurable hardware such as FPGA can be used to increase the (effective) flops per watt. We investigate novel implementation method for these hardware and associated algorithms in typical computational kernels demanded in real world applications.

4.1.2 Development in this Study

In this study, we focus on the following issues towards development of numerical algorithms and applications with *data movement intensity* on the supercomputer systems in the *Post Moore Era*:

- High bandwidth in memory and network, Large capacity of memory and cache
- Large and heterogeneous latency due to hierarchy in memory and network
- Utilization of FPGA
- High concurrency with $O(10^3)$ threads on each node

Finally, we will develop the following libraries and frameworks:

- Linear solver library for both of sparse and dense matrices
- FFT library
- Framework for automatic-tuning (AT)
- Framework for application development

Because *implicit schemes* are very feasible for supercomputer systems in the Post Moore Era, we need to design and develop efficient and robust implicit solvers for both of sparse and dense matrices, and FFT algorithms.

Generally, parallelization and domain decomposition of FEM and FDM have been done in *space* direction. In order to keep scalability on future supercomputer systems with large number of nodes with hierarchical network, parallelization in *time* direction for time-dependent problems is essential. This type of method is called *parallel-in-space/time (PiST)*, and it can be effective if more than several hundred MPI processes are applied [13]. PiST approach is suitable for supercomputer systems with large latency and with network hierarchy in the Post Moore Era.

Since 2011, we have been developing *ppOpen-HPC* [29]. ppOpen-HPC is an open source infrastructure for development and execution of optimized and reliable simulation code on post-peta-scale (pp) parallel computers based on many-core architectures, and it consists of various types of libraries, which cover general procedures for scientific computation. ppOpen-HPC includes the four components, ppOpen-APPL (application development framework), ppOpen-MATH (math library), ppOpen-AT (automatic-tuning framework), and ppOpen-SYS (system software).

Automatic tuning (AT) is one of the critical technologies for performance portability of scientific applications on future supercomputer systems in the Post Moore Era. ppOpen-AT is an framework for AT with directive-based special language, and automatically and adaptively generates optimum implementation for efficient memory accesses in the processes of methods for scientific computing in each component of ppOpen-HPC. Although ppOpen-AT provides *compute intensity*, we will develop new strategy for AT towards *data movement intensity* in this study.

Finally, we integrate these libraries (linear solvers (dense and sparse matrices) and FFT), and AT capabilities into a new framework for application development in the Post Moore Era. In the following three sections, details of the development are described.

4.2 Numerical Algorithms, Libraries and Frameworks for Post-Moore Era

In our research, we focus on following four computational kernels: (1) iterative stencil computations (2) transient analyses (3) approximated matrix computations (4) sparse matrix computa-

tions. These kernels are often used in various simulations such as plasma simulations, CFD, earthquake simulations, social system analyses, information sciences, electromagnetic field analyses etc.

4.2.1 Iterative stencil computations

In the area of iterative stencil computations, we focus on the three dimensional FDTD method which is the most popular method for the electromagnetic field wave simulations. While the 3D FDTD has more complex stencil structure than 7 point finite difference method, we have successfully applied the temporal tiling to the method and also developed the auto-tunning method for the tile shape [26]. We are currently developing the implementation method of tiled 3D FDTD method for many core processors and GPU, on which more than hundreds of threads should be effectively utilized.

4.2.2 High performance parallel multigrid solver in space and time for transient analyses

Communication overhead generally increases, as number of nodes increases on large-scale supercomputer systems. Therefore, efficient communication is always the most critical issue in parallel algorithms, such as parallel preconditioned iterative solvers. The algorithm of preconditioned iterative methods in the Post Moore Era must be carefully designed and developed.

Parallel *multigrid* method is widely-used scalable method for large scale problems, and is expected to be one of the main algorithms in the Post Moore Era. It is well-known that multigrid method suffers from significant communication overhead, if number of MPI processes is large. We proposed the *h*CGA (Hierarchical Coarse Grid Aggregation) [28] for reducing the communication overhead in parallel multigrid method. In *h*CGA (Fig. 10), number of MPI processes is reduced and processes are repartitioned in an intermediate level before the final coarse grid solver on a single MPI process. Figure 11 shows the performance of weak scaling by 8 to 4,096 nodes of the Fujitsu FX10 at the University of Tokyo. Matrices derived from 3D Poisson's equations for groundwater flow problems through heterogeneous porous media are solved by CG method preconditioned by multigrid (*MGCG*). Improvement of performance by *h*CGA at 4,096 nodes (65,536 cores) with 17.2×10^9 unknowns reaches 60%. The approach, such as *h*CGA, is suitable for for supercomputer systems with network hierarchy in the Post Moore Era.

As shown in the previous section, *PiST (Parallel in Space and Time)* is suitable for the supercomputer systems with large latency and with network hierarchy in the Post Moore Era. PiST accelerates simulations for the time-dependent problems.

In [35], we developed parallel multigrid method in time for non-linear finite element analyses for electric motors. The method provides more degree of parallelism than the conventional parallel multigrid method in space. We are conducting the research on "algebraic" type of parallel multigrid method in time.

4.2.3 Approximated matrix computation

We also pay special attention to the H-matrices. The H-matrix is the approximation technique for a dense matrix. It reduces both the computational cost and the memory footprint for the dense matrix. There are many reports that describe the successful applications of the technique to various problems such as boundary element analyses or N-body problems. The H-matrix has a similar function to the Fast Multipole Method (FMM). While the FMM is a low B/F method, the H-matrix is a relatively high B/F method which is worth to be invetigated for the post Moore's era. We have already developed a distributed parallel H-matrix library called "HACApK" with a software framework for BEM analyses [18] [19]. While the HACApK library supports the hybrid multi-process and thread parallelism, we will improve it for applications run in huge number of processes and threads (Fig. 12). Moreover, the library will be tuned for accelerators and future machines.

4.2.4 Sparse matrix computations

Sparse matrix computations are used for not only conventional PDE analyses but also various new types of simulations such as big data analyses. We carry out research on sparse matrix vector multiplication kernels, the linear iterative solvers, and eigenvalue solvers. We try to develop new algorithms and implementation methods for these kernels on accelerators, FPGAs and the computing unit associated with memory.

Furthermore, we pay special attention to the communication overhead involved in the applications with sparse matrix computations. We will investigate the communication hiding and avoiding algorithms such as the *pipelined CG method* [16] more intensively.

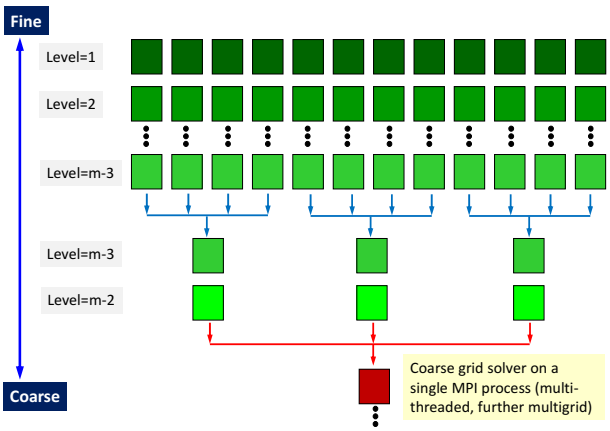


Fig. 10 Procedures of hierarchical CGA (*h*CGA), where number of MPI processes is reduced before the final coarse grid solver of CGA on a single MPI process [28]

4.3 Algorithmic Breakthrough for Dense Matrix Computations, FFT, Accuracy Assurance, and Auto-tuning Technologies

4.3.1 Overview

With nature of heterogeneous, hierarchical and extremely high ability of bandwidth of memories towards to Post Moore era, we research the following topics with respect to the estimated change from FLOPS to BYTES: (1) A new algorithm for dense matrix computations and FFT; (2) high performance implementations of accuracy assurance algorithm; (3) AT framework to apply (1) and (2) with respect to current AT framework; (4) A new circuit implementations of FPGA for key computations from (1) and (2).

Overview of the topics is shown in Fig. 13.

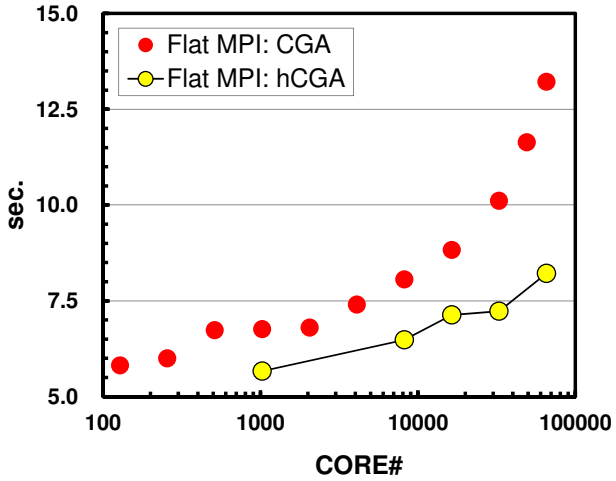


Fig. 11 Performance of MGCG solver on Fujitsu FX10 using up to 4,096 nodes (65,536 cores), weak scaling (elapsed time for MGCG): 262,144 (=64^3) meshes/core, max. total problem size: 17,179,869,184 meshes [28]

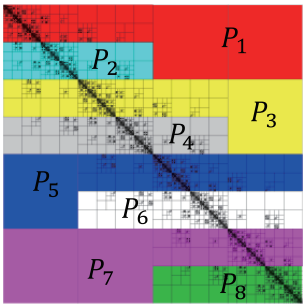


Fig. 12 Division of H-matrix for multiple processes in HACApK library

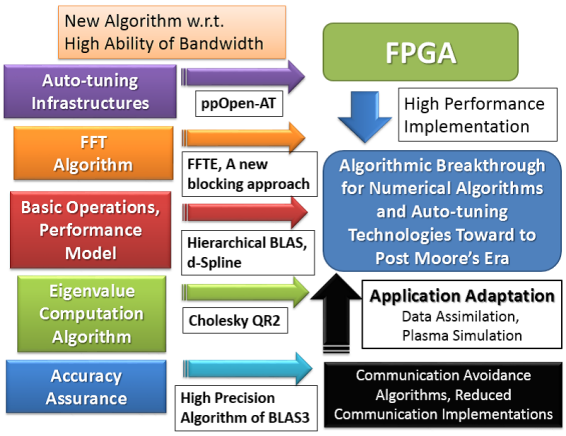


Fig. 13 Overview of research topics.

4.3.2 Dense Matrix Computations and FFT

Current numerical algorithms are based on high ability of computations (high FLOPS ratios), hence it is not enough taken care of high ability of memory bandwidth (BYTES). There are several blocking approaches with caches by utilizing localization of memory accesses while several blocking approaches require additional FLOPs with compared to algorithms by non-blocking approaches. This is merit for increasing FLOPS by Moore's law.

In addition we also need to follow advanced hardware to Post Moore era. FPFA is one of the advanced technologies to be considered to increase FLOPS per watt. Hence developing novel implementations on FPGA for kernel computations from matrix computations is crucial.

Key topics in Fig. 13 for dense matrix computations and FFT are summarized as follows:

- **Basic Operation**: Libraries of basic numerical operations, such as BLAS, are also key building blocks to develop high performance numerical library. We research a new high performance implementation of BLAS based on hierarchical memories.
- **Eigenvalue Computation Algorithm**: A blocking approach for eigenvalue computation increases FLOPS [22]. We investigate feasibility of non-blocking algorithm in viewpoint of increasing BYTES. On the other hand, communication avoiding (CA) is crucial approach to attain high performance in massively parallel execution. We also research novel CA algorithms for eigenvalue computation, such as Cholesky QR2 algorithm [40].
- **FFT Algorithm**: FFT (Fast Fourier Transform) requires abilities of data movement rather than that of FLOPS. Conventional algorithm is based on blocking of caches [34]. In this project, a new blocking approach based on BYTES will be investigated.

4.3.3 Accuracy Assurance, Auto-tuning Technologies, and Application Adaptations

Assuring accuracy of computed results is also one of principal issues for numerical computations. Although novel framework of accuracy assurance is utilizing level 3 BLAS (BLAS3) operations [30], but no framework by using memory hierarchy and extremely high ability bandwidth is proposed.

Auto-tuning (AT) technology is spotlighted to attain sustained performance between different computer architectures. AT technology is also expected to be one of key technologies toward to Post Moore era. Urgent topics for AT is to establish framework of code optimizations for hierarchical memories. Making performance model for principal numerical computations on upcoming architectures is also important issue to reduce search space by AT.

Key topics in Fig. 13 for accuracy assurance and AT technologies are summarized as follows:

- **Accuracy Assurance**: Conventional approaches of accuracy assurance are based on blocking and BLAS3 implementations. In particular, a high precision algorithm of matrix-matrix multiplication with BLAS3 is proposed to do perfect blocking of caches [30]. In this project, we make a new methodology to adapt hierarchical memories and by utilizing extremely high BYTES for the algorithms of accuracy assurance.
- **AT Infrastructure**: Specifying AT functions into arbitrary user programs is fundamental issue. In previous research, we have developed an AT language, named ppOpen-AT [23]. ppOpen-AT is designed to process codes of actual simulation by ppOpen-HPC [29]. Utilizing basic functions of ppOpen-AT, in particular, for loop transformations and algorithm selection, we extend functions of AT by new requirements of optimization for hierarchical memories and increasing BYTES with respect to newly developed technology by the group of Post-Moore Programming and System Software.

- **Performance Model**: Modeling computations on advanced architectures contributes effective code optimizations. In this project, we develop methodology of general performance model to adapt AT technology. We start it with a previous research, named d-Spline model [38]. To make a model, we collaborate with the group of Post-Moore High-performance Architecture for hardware trends and basics.
- **Application Adaptation**: The developed new algorithms including CA algorithms, and new implementations with reduced communications in the project are adapted to real applications. For example, data assimilation for big data processing and numerical simulations such as plasma are one of targets. Other applications inside the group of Post-Moore Numerical Methods and Algorithm also will be evaluated.

4.4 Framework for Development of Data-Movement Centric Applications in Post Moore Era

In this study, we develop an extended version ppOpen-HPC (ppOpen-APPL and part of ppOpen-MATH) [29] by *parallel-in-space/time (PiST)* method [13] for supercomputer systems in the Post Moore Era. ppOpen-APPL is a set of libraries covering various types of procedures for five methods (FEM, FDM, FVM, BEM, and DEM), such as parallel I/O of data-sets, assembling of coefficient matrix, linear-solvers with robust and scalable preconditioners, adaptive mesh refinement (AMR), and dynamic load-balancing. ppOpen-MATH is a set of libraries for multigrid, visualization, loose coupling, etc.

In this study, we develop the following components of the application framework with *parallel-in-space/time (PiST)* method:

- Nonlinear Algorithm
- Adaptive Mesh Refinement (AMR)
- Visualization
- Coupler for Multiphysics

Although the PiST method was originally implemented to linear time-dependent problems, it is also feasible for nonlinear problems [14]. We develop robust and efficient non-linear algorithm by PiST method and apply the developed method to integrated earthquake simulation code by FEM [17] developed by our collaborators.

ppOpen-MATH/MP is a coupling software applicable to the models employing various discretization (Fig. 11). It was originally developed for coupled simulation of NICAM (atmospheric model, semi-unstructured FVM) and COCO (ocean model, structured FDM) on K computer [3]. The developed tool has been extended for coupling of general scientific applications developed in the ppOpen-HPC project, and applied to coupled earthquake simulation, where seismic wave propagation (Seism3D on ppOpen-APPL/FDM) and building vibration (FrontISTR+ on ppOpen-APPL/FEM) have been coupled [24]. Performance of the coupled code has been evaluated using 4,560 nodes (72,960 cores) of Fujitsu PRIMEHPC FX10 at the University of Tokyo [24]. We develop a coupling tool with PiST. Moreover, we develop parallel AMR and visualization tools with PiST. Coupling, AMR and visualization are data-movement centric processes in scientific computing, and implementation of PiST to such processes is very challenging.

Finally, developed methods are validated through real simulations for atmosphere/ocean science, earthquake science, material science, fluid mechanics and structural engineering. Utilization of FPGA to complicated procedures of real applications are also evaluated under collaboration with other groups. Furthermore, developed components in the following two chapters (numerical libraries for dense and sparse matrices, automatic tuning (AT), and performance model) are implemented to this application framework.

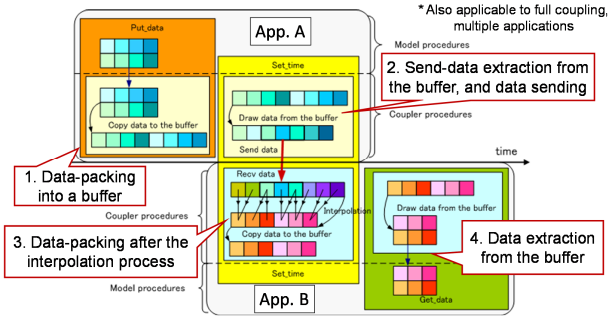


Fig. 14 Data Flow in ppOpen-MATH/MP [24]

5. Discussion and Future Work

Post-Moore research is still in its infancy, as the real value of the entire research portfolio to convert from FLOPS to BYTES may not be fully utilized until the mid-2020s. However, even for various exascale projects, we are already observing slowdowns in the lithography improvement since 28nm scaling, and this is expected to worsen as optical lithography is reaching its limits around 7nm, and beyond that UV lithography is an expensive endeavor. As such, many of the early results would be applicable as we approach the end of Moore's law, including the numerical algorithms to utilize the memory hierarchy, techniques such as autotuning, software frameworks for memory hierarchy and FPGAs, as well as new hardware designs and optics. Also, there would be many other research elements that could be utilized for the benefit of Post-Moore era.

We hope to be launching a comprehensive program on Post-Moore soon, to prepare for the inevitable transition from FLOPS to BYTES.

References

[1] 2013 Exascale Operating and Runtime Systems. Technical report, Advanced Science Computing Research (ASCR), February 2013. `http://science.doe.gov/grants/pdf/LAB13-02.pdf`.

[2] D. Ajwani, S. Ali, and J.P. Morrison. Graph partitioning for reconfigurable topology. In *Parallel Distributed Processing Symposium (IPDPS), 2012 IEEE 26th International*, pages 836–847, 2012.

[3] T. Arakawa, T. Inoue, and M. Satoh. Performance evaluation and case study of a coupling software ppopen-math/mp. In *Procedia Computer Science 29*, pages 924–935. Elsevier, 2014.

[4] Steve Ashby, Pete Beckman, Jackie Chen, Phil Colella, Bill Collins, Dona Crawford, Jack Dongarra, Doug Kothe, Rusty Lusk, Paul Messina, Tony Mezzacappa, Parviz Moin, Mike Norman, Robert Rosner, Vivek Sarkar, Andrew Siegel, Fred Streitz, Andy White, and Margaret Wright. The Opportunities and Challenges of Exascale Computing. 2010.

[5] K.J. Barker, A. Benner, R. Hoare, A. Hoisie, A.K. Jones, D.K. Kerbyson, D. Li, R. Melhem, R. Rajamony, E. Schenfeld, S. Shao,

C. Stunkel, and P. Walker. On the feasibility of optical circuit switching for high performance computing systems. In *Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference*, pages 16–16, 2005.

[6] A. Chakraborty, E. Schenfeld, and M. Silva. Switching optically-connected memories in a large-scale system. In *Parallel Distributed Processing Symposium (IPDPS), 2012 IEEE 26th International*, pages 727–738, 2012.

[7] K. Christodoulopoulos, K. Katrinis, M. Ruffini, and D. O'Mahony. Accelerating hpc workloads with dynamic adaptation of a software-defined hybrid electronic/optical interconnect. In *Optical Fiber Communications Conference and Exhibition (OFC), 2014*, pages 1–3, 2014.

[8] Jens Domke, Torsten Hoefler, and Satoshi Matsuoka. Fail-in-place network design: interaction between topology, routing algorithm and failures. In *Proceedings of IEEE/ACM International Conference for High Performance Computing, Networking, Storage and Analysis (SC14)*, pages 597–608. IEEE/ACM, Nov 2014.

[9] Toshio Endo and Guanghao Jin. Software technologies coping with memory hierarchy of gpgpu clusters for stencil computations. In *Proceedings of IEEE Cluster Computing (CLUSTER2014)*, pages 132–139. IEEE, Sep 2014.

[10] Toshio Endo, Yuki Takasaki, and Satoshi Matsuoka. Realizing extremely large-scale stencil applications on gpu supercomputers. In *Proceedings of IEEE International Conference on Parallel and Distributed Systems (ICPADS 2015)*, pages 625–632. IEEE, Dec 2015.

[11] H. Esmaeilzadeh, A. Sampson, L. Ceze, and D. Burger. Neural acceleration for general-purpose approximate programs. In *International Symposium on Microarchitectur*, pages 449–460, Dec 2012.

[12] Hadi Esmaeilzadeh, Emily Blem, Renee St. Amant, Karthikeyan Sankaralingam, and Doug Burger. Dark silicon and the end of multicore scaling. *IEEE Micro*, 32(3):122–134, 2012.

[13] R.D. Falgout, S. Friedhoff, Tz.V. Kolev, S.P. MacLachlan, and J.B. Schroder. Parallel time integration with multigrid. In *SIAM Journal on Scientific Computing 36-6*, pages 635–661, 2014.

[14] R.D. Falgout, A. Katz, Tz.V. Kolev, J.B. Schroder, A.M. Wissink, and U.M. Yang. Parallel time integration with multigrid reduction for a compressible fluid dynamics application. In *LLNL-JRNL-663416*, pages 1–15, 2015.

[15] N. Farrington, A. Forencich, G. Porter, P.-C. Sun, J.E. Ford, Y. Fainman, G.C. Papen, and A. Vahdat. A multiport microsecond optical circuit switch for data center networking. *Photonics Technology Letters, IEEE*, 25(16):1589–1592, 2013.

[16] P. Ghysels and W. Vanroose. Hiding global synchronization latency in the preconditioned conjugate gradient algorithm. In *Parallel Computing 40-7*, pages 224–238. Elsevier, 2014.

[17] T. Ichimura, K. Fujita, P.E.B. Quinay, L. Wijerathne, M. Hori, S. Tanaka, Y. Shizawa, H. Kobayashi, and K. Minami. Implicit nonlinear wave simulation with 1.08t dof and 0.270t unstructured finite elements to enhance comprehensive earthquake simulation. In *IEEE/ACM Proceedings of SC15*, 2015.

[18] A. Ida, T. Iwashita, T. Mifune, and Y. Takahashi. Parallel hierarchical matrices with adaptive cross approxima-tion on symmetric multiprocessing clusters. *Journal of Information Processing*, 22:642–650, 2014.

[19] A. Ida, T. Iwashita, M. Ohtani, and K. Hirahara. Improvement of hierarchical matrices with adaptive cross approximation for large-scale simulation. *Journal of Information Processing*, 23:366–372, 2015.

[20] Y. Inadomi, T. Patki, K. Inoue, M. Aoyagi, R. Rountree, M. Schulz, D. Lowenthal, Y. Wada, K. Fukazawa, M. Ueda, M. Kondo, and I. Miyoshi. Analyzing and mitigating the impact of manufacturing variability in power-constrained supercomputing. In *Proc. of the International Conference for High Performance Computing, Networking, Storage and Analysis*, Nov 2015.

[21] T. Kagami, H. Matsutani, M. Koibuchi, Y. Take, T. Kuroda, and H.Amano. Efficient 3-D bus architectures for inductive-coupling ThruChip Interfaces. In *IEEE Trans. on VLSI systems*, pages 493–506. Vol.24, No.2, Feb. 2016.

[22] Takahiro Katagiri, Jun'ichi Iwata, and Kazuyuki Uchida. A communication avoiding and reducing algorithm for symmetric eigenproblem for very small matrices. In *IPSJ SIG Notes, 2015-HPC-148 (2)*, pages 1–17. IPSJ, February 2015.

[23] Takahiro Katagiri, Satoshi Ohshima, and Masaharu Matsumoto. Directive-based auto-tuning for the finite difference method on the xeon phi. In *IPDPSW2015 (iWAPT2015)*, pages 1221–1230. IEEE, May 2015.

[24] M. Matsumoto, T. Arakawa, T. Kitayama, F. Mori, H. Okuda, T. Furumura, and K. Nakajima. Multi-scale coupling simulation of seismic waves and building vibrations using ppopen-hpc. In *Procedia Computer Science 51*, pages 1514–1523. Elsevier, 2015.

[25] S. Matsuoka, H. Amano, K. Nakajima, K. Inoue, T. Kudoh, N. Maruyama, K. Taura, T. Iwashita, T. Katagiri, T. Hanawa, and T. Endo. From flops to bytes: Disruptive change in high-performance computing towards the post-moore era. In *proceedings of the ACM International Conference on Computing Frontiers (CF'16)*, pages 274–281. ACM, 2016.

[26] T. Minami, M. Hibino, T. Hiraishi, T. Iwashita, and H. Nakashima. Automatic parameter tuning of three-dimensional tiled fdtd kernel. In *Proceedings of the Ninth International Workshop on Automatic Performance Tuning (iWAPT2014)*, 2014.

[27] N. Miura, Y. Kohama, Y. Sugimori, H. Ishikuro, T. Sakurai, and T. Kuroda. A high-speed inductive-coupling link with burst transmission. *IEEE Journal of Solid-State Circuits*, 44(3):947–955, 2009.

[28] K. Nakajima. Optimization of serial and parallel communications for parallel geometric multigrid method. In *Proceedings of the IEEE 20th International Conference for Parallel and Distributed Systems (ICPADS 2014)*, pages 25–32, 2014.

[29] K. Nakajima, M. Satoh, T. Furumura, H. Okuda, T. Iwashita, H. Sakaguchi, T. Katagiri, M. Matsumoto, S. Ohshima, H. Jitsumoto, T. Arakawa, F. Mori, T. Kitayama, A. Ida, and M.Y. Matsuo. ppopen-hpc: Open source infrastructure for development and execution of large-scale scientific applications on post-peta-scale supercomputers with automatic tuning (at). In *Optimization in the Real World -Towards Solving Real-Worlds Optimization Problems, Mathematics for Industry 13*, pages 15–35. Springer, 2015.

[30] Katsuhisa Ozaki, Takeshi Ogita, and Shin'ichi Oishi. Improvement of error-free splitting for accurate matrix multiplication. *Journal of Computational and Applied Mathematics*, 288:127–140, November 2015.

[31] H. Rodrigues, R. Strong, A. Akyurek, and T.S. Rosing. Dynamic optical switching for latency sensitive applications. In *Architectures for Networking and Communications Systems (ANCS), 2015 ACM/IEEE Symposium on*, pages 75–86, 2015.

[32] Kento Sato, Kathryn Mohror, Adam Moody, Todd Gamblin, Bronis R. de Supinski, Naoya Maruyama, and Satoshi Matsuoka. A user-level infiniband-based file system and checkpoint strategy for burst buffers. In *Proceedings of IEEE Cluster Computing (CLUSTER2014)*, pages 21–30. IEEE, Sep 2014.

[33] Vilas Sridharan, Nathan DeBardeleben, Sean Blanchard, Kurt B. Ferreira, Jon Stearley, John Shalf, and Sudhanva Gurumurthi. Memory errors in modern systems: The good, the bad, and the ugly. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pages 297–310. ACM, 2015.

[34] Daisuke Takahashi. An implementation of parallel 2-d fft using intel avx instructions on multi-core processors. In *12th International Conference, ICA3PP*, pages 197–205. Springer Lecture Notes in Computer Science, September 2012.

[35] Y. Takahashi, T. Tokumasu, K. Fujiwara, T. Iwashita, and H. Nakashima. Parallel tp-eec method based on phase conversion for time-periodic nonlinear magnetic field problems. *IEEE Transactions on Magnetics*, 51, 2015.

[36] Y. Take, H. Matsutani, H. Sasaki, M. Koibuchi, T. Kuroda, and H. Amano. 3D noc with inductive-couplings for building-block SiPs. In *IEEE Trans. on Computers*, pages 748–763. Vol.63, No.3, March 2014.

[37] S. Takizawa, T. Endo, and S. Matsuoka. Locality aware mpi communication on a commodity opto-electronic hybrid network. In *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, pages 1–8, 2008.

[38] Teruo Tanaka, Ryo Otsuka, Akihiro Fujii, Takahiro Katagiri, and Toshiyuki Imamura. Implementation of d-spline-based incremental performance parameter estimation method with ppopen-at. *Scientific Programming*, 22:299–307, 2014.

[39] Ke Wen, D. Calhoun, S. Rumley, Xiaoliang Zhu, Yang Liu, Lian Wee Luo, Ran Ding, T.B. Jones, M. Hochberg, M. Lipson, and K. Bergman. Reuse distance based circuit replacement in silicon photonic interconnection networks for hpc. In *High-Performance Interconnects (HOTI), 2014 IEEE 22nd Annual Symposium on*, pages 49–56, 2014.

[40] Yusaku Yamamoto, Yuji Nakatsukasa, Yuka Yanagisawa, and Takeshi Fukaya. Roundoff error analysis of the choleskyqr2 algorithm. *Electronic Transactions on Numerical Analysis*, 44:306–326, January 2015.

[41] Yuetsu Kodama and Toshihiro Hanawa and Taisuke Boku and Mitsuhisa Sato. PEACH2: An FPGA-based PCIe network device for Tightly Coupled Accelerators. In *HEART2014*, June 2014.

[42] Hamid Reza Zohouri, Naoya Maruyama, Aaron Smith, Motohiko Matsuda, and Satoshi Matsuoka. Evaluating and optimizing opencl kernels for high performance computing with fpgas. In *Proceedings of IEEE/ACM International Conference for High Performance Computing, Networking, Storage and Analysis (SC16)*. IEEE/ACM, Nov 2016. To appear.