Heterogeneous Computing: Promise and Challenge

Peter J. Zievers, CTO Xcelemor, Inc.

July 31, 2014

Abstract

The user community benefits from heterogeneous computing technologies today more than ever. These benefits include vastly improved per-function performance at a fraction of the power consumption compared to comparable software functions executing on a microprocessor. Microprocessor solutions today struggle to meet the demands of ever more complex feature expectations for both physical limitations and known architectural constraints. Additionally, creeping operational expense issues gradually compromise scaling prospects for microprocessors as a technology choice. Despite all contributions and considerable potential as an alternative, designers relegate heterogeneous computing to an assistant role. Scalable heterogeneous computing has ample range to satisfy generic application needs while slashing power consumption. Control plane performance considerations focus requirements for the successful solution.

1 The Problem

1.1 Introduction

Modern Field Programmable Gate Array (FPGA) technology configures the array of programmable logic elements by setting bits in a RAM array. FPGAs are available that permit sector reprogrammability, or the facility for altering the programming on selected subsections of an FPGA. Such fine functional granularity enables transition of instantiated hardware designs at arbitrary times and on a nearly arbitrary scale relative to the size of the FPGA. Theoretically, a sector programmable RAM based FPGA should be able to time-multiplex hardware functionality, permitting such functionality to be mixed with other hardware designs that are all delivered for use on a schedule.

Graphics Processing Units provide a high-density compute plant for general purpose application. They are partitionable and configurable, enabling high-throughput precision modest-to-high complexity computation in a compact form factor. Sophisticated high-bandwidth memory ports and a flexible internal interconnect support parallel computation threads, as well.

Digital signal processors (DSPs) have solved demanding numerical applications for a host of products, and continue to succeed today in that capacity. For example, the DSP is the workhorse technical choice for products as diverse as wireless personal electronics and synthetic aperture radar imaging. FPGAs

also synergistically contain DSP functionality, enhancing the traditional DSP application base while expanding the FPGA application base.

Microprocessors have grown into computational powerhouses in their own right. A vibrant comprehensive application development infrastructure enables a vast and deep application pool. Software solutions pervade our daily existence powered by the ubiquitous microprocessor. Computational density increases with the availability of additional cores in a microprocessor package.

The available technologies support an unimaginable range of application functionality. Combining these technologies in creative ways exponentially increases the total solution range, since the strengths of one technology match the weaknesses of others. All of these technologies share basic integrated circuit device technologies, such as packaging and drive electronics, enabling systems to be built with combinations of these technologies.

As of this writing, there is no comprehensive management scheme that fully realizes the potential of combining these individually high-potential technologies. The problem statement in section 1.2 formalizes this issue.

1.2 Technical Problem Statement

Dynamically synthesize and manage an arbitrary digital hardware circuit design employing any and all available processing technologies on an arbitrary scale.

2 The Technical Environment

The opportunity lies at the crossroads of the potential of the technological building materials and the needs, preferences, and expectations of the collective consuming user community. Prominent drivers of the digital systems design landscape are:

- User Demand
  - Information explosion
  - Microprocessor speed cap
  - Electric power cost
  - Battery capacity
  - Time to market
  - Acceptance of heterogeneous computing
- Technology Supply
  - Optical technologies
  - Languages and Build environments
  - Integrated circuit packaging
  - Integrated circuit device density
  - Commercially-available packaged application-level functionality

2.1 User Demand

Looking back over the last 35 years, or so, technical evolution drove the advancement of computing. When microprocessors first emerged, they were specialized devices that could be programmed for simple or moderately complex logical operations that were often hard-wired in TTL logic using discrete devices [Betker97]. Until about 5 to 10 years ago, advances in microprocessor design and fabrication techniques continued to drive computation capacity, a run lasting about 25 to 30 years [Patt10]. Users were happy, because just by waiting for a 9-12 month development cycle they would be rewarded with jumps in computation capacity. Then things changed.

Beginning about 5 to 10 years ago, clock speeds stopped improving [Mims10]. The industry has been stuck at about 3GHz CPU clock speed since. Users noticeed that the CPU was no longer pushing the industry faster and over time began to react by trying to apply more microprocessors to their former problems. Device manufactures tried (and continue to try) to help by making more and more cores available within devices, but get stuck at familiar bottlenecks due to software application design, clock speed, I/O limitations, and data access speed [Patt10]. The bulk of the application developer community is still fairly well wedded to the traditional core-per-application computation model for its programming simplicity even as Intel itself aggressively promotes a multicore programming model. Amdahl's Law reveals the well-established complication in the multicore plan, stated thus

$$Overall\ Speedup = \frac{1}{S + \frac{F}{n}}, \tag{1}$$

where

Overall Speedup	$\equiv$	Overall speedup of the parallelized application
F	$\equiv$	Parallelizable fraction of application
S	$\equiv$	Serial fraction of application
n	$\equiv$	Number of parallel threads of execution

In other words, the intrinsic nature of the application program limits the value of additional cores. For example, if most of a program must run serially then available additional processing facilities waste. Additional work traces performance limitations more deeply into the application [Gelenbe88]. Still, with microprocessor technology stalled in some very key ways, users continue to expect top performance even as their increasing sophistication drives features to ever higher complexity and surging demand stresses the infrastructure to the breaking point [Lohr12], [Equat11], [Venka13].

The information explosion was heralded as far back as the 60s and 70s, where the term was used to describe the information made available by mass media (television, radio, print). Then personal computers became commonplace in the home and the office when the internet happened. The implications reach far and wide [Sween03], [McIlr10], [Turner08]. With first thousands and then millions of people contributing content that was suddenly consumable by virtually anyone anywhere anytime, information growth escalated from what could be termed an "explosion" to something more resembling a chain reaction. Information theory tells us that as a system with N components grows, the number of connections grows $O(N^2)$ (or, more precisely $N * (N - 1)$). As each endpoint grows in content, it seems that the bandwidth required grows even more rapidly than

$O(N^2)$. If we look at publicly available data on computation growth, we often see such results [Borer13].

The microprocessor speed cap has already been mentioned briefly. It's no coincidence that multi-core solutions arrived on the heels of that roadblock to CPU performance growth. The very fact that multi-core did not evolve before this is strong evidence that the CPU manufactures knew that multi-core was a sub-optimal solution; otherwise it would have been pursued more vigorously in the previous 25 years. Back when CPU speeds were 33 or 66 MHz, one could count on upgrading the PC's CPU engine once or twice over the life of the box as long as your CPU was socket mounted [Little98]. Heat dissipation properties of the silicon substrate put an end to this growth path, and the vast preponderance of users have been stuck at roughly 3GHz since.

The tech industry is facing an energy crisis. The cost of power consumption by data centers doubled between 2000 and 2006, to $4.5 billion, and prospectively doubled again by 2011, according to the U.S. government. With energy prices spiking, the challenge of powering and cooling these SUVs of the tech world has become a major issue for corporations and utilities. "The digital economy is pervasive," said Andy Karsner, Assistant U.S. Energy Secretary for energy efficiency. "The demands for computing will grow exponentially, but electric consumption can't grow the same way." The $4.5 billion spent in the U.S. in 2006 is the equivalent of the electric bills for 5.8 million U.S. households. The cost to sustain the ocean of microprocessors in a data center grows ever more daunting. These days, for every $1 spent on computing equipment in data centers, an additional 50 cents is spent each year to power and cool them. About half of the energy is for air conditioning. Power consumption is an open problem for data centers [Hamm08]. Borer et al [Borer13] provide more detailed estimates that reach a similar conclusion. Borer et al estimate $0.35 is a reasonable operating expense associated with $1 of compute equipment spend, but points to another $0.20 as the amortized cost of the data center infrastructure to power and house the compute equipment.

Batteries may seem technologically mundane to some, but users feel strongly about their contribution. Multiple recent studies reveal that longer battery life is at the top or near the top of the typical mobile device user wish list [Versace13], [Knight13], [Saginor12]. Specific factors contributing to faster battery drain include the 3G to 4G transition [Saginor12], multiple antenna support and bigger and more colorful displays [Versace13]. This finding extends to tablet users, as well [Knight13]. The need for better batteries goes beyond convenience for users of mobile electronics into issues of national security and energy independence [Fitz13], ever more-strongly incenting battery vendors toward product improvement. Despite the promised rewards, battery technology development is notoriously slow. Narrowly held supply for the key material in today's workhorse batteries further complicates the future of batteries [Tahil07].

Users crave new features and vendors strive to meet demand. For example, Apple released major new feature packages for its iPhone product on a pace of one major release per year [Sacco13]. The unrelenting user community has an impressive wish list of fairly luxurious features for the release of iPhone6 [Mariano13]. A methodology for evaluating benefits of faster time to market for information technology applications shows that measurable benefits can be realized around overall business value, higher return on investment, and reduced risk [Glied13].

Heterogeneous computing techniques have accelerated computing since the beginnings of broad industrial use [Estrin07]. Early work horse heterogeneous applications centered around vector processors, which offloaded from the main CPU rigorous functions such as numerically intensive array calculations. These applications took the form of co-processors not unlike those available for personal computer systems up until the introduction of the 80486 class of microprocessors. Today, co-processors routinely populate computing systems in the form of ethernet line processors and video cards. As of 2013, 53 of the fastest 500 computers used co-processors, with 31 of these using NVIDIA's GPU [Courtland13]. The extended list of applications alluded to in section 4 demonstrates that development into more focused application domains has been active for the last 20 years at least.

2.2 Technology Drivers

More than 75 years of evolution delivers modern electronic design to us now. Designers routinely build complex and sophisticated components into still more complicated systems today. Today's complicated systems in turn are destined to be the building blocks of tomorrow's design. Thanks to deeper understanding in the development community, these complex components often deploy in a straightforward manner at no loss to their intrinsic complexity, tacitly certifying the maturity of the state of the art.

Inexpensive transport bandwidth has ushered in broad-based interconnection of electronics, and in so doing, transformed the landscape of electronic design. Rapidly developing dense wavelength division multiplexing (DWDM) implementations swamped legacy switching architectures [Bawab02]. Compact commercially available DWDM low-power 10G optical transceiver solutions commonly exist that support more than 100 channels [Broc14]. Vertical cavity side-emitting lasers (VCSELs) offer inexpensive, low-power, high-performance interconnect between modules generally available with 10G capacity per channel in a 12-channel package [Fini14].

The great diversity in languages parallels the dramatic range of individual application domains. Development organizations work hard to integrate mixed-language contributions and to streamline supply-chain issues. Hence, there are strong incentives to re-use proven field-tested software and/or hardware components. Simultaneously, the components and languages grow in feature complexity. OpenCL [Open14] takes a promising approach in that it respects the differences between diverse base technologies but couches availability in a common, familiar framework. OpenCL acceptance grows as it encompasses GPU, DSP, microprocessor, and now FPGA application development. The advanced state of the art in build environments enables the process of alloying myriad combinations of languages, components, and even entire products.

Integrated circuit packaging supports high-performance transceiver technology with impressive pin densities and counts. For instance, today one can purchase a Virtex7 device with 1200 usable input/output sites each of which is enabled for 1.866 Gbit/second single ended transfer. These devices can be equipped with SERDES transfer ports that support up to 28.8 GB/s. As well, the NVIDIA Fermi device [NVID14] supports six independent GDDR5 64-bit memory ports running at base speeds exceeding 1 GHz. Emerging technologies [Schow10] optically connect devices on a printed circuit board, boosting

device throughput while slashing crosstalk.

Advances in device physics produce functionally dense devices capable of operation in the gigahertz range. NVIDIA's Fermi device has 512 independently operating computational cores. Intel built an experimental microprocessor with 80 independently-operating computational cores [Zaz08] and markets a 72-core microprocessor [Anth12]. Altera and Xilinx each offer 20nm-range high-capacity system-on-a-chip products [Altr13] [Leib13]. Late model FPGAs contain substantially more programmable logic capacity than earlier models while offering diverse and capacious input/output facilities. Acronix samples a product that features > 1M look up table, > 100Mbits of embedded RAM, and a raft of I/O all in the same package [Achx13]. Texas Instruments Keystone product features up to 8 DSP cores and 4 Cortex processors on the same die [TI12], all capable of independent operation.

Complex product development requires modular pre-built functionality to achieve acceptable time-to-market. Deep function catalogs spanning a vast range of different functionalities enable fast delivery of boundlessly functional reliable application solutions and free the product designers to think more abstractly. These catalogs or libraries take the form of blocks of software or hardware and sometimes come bundled in software/hardware combinations. The variety and number of libraries is too enormous to enumerate all that's available, but the bibleography references some examples.

3 The Opportunity

Users demand that the march of technology [Netcraft13] continues to progress unabated. Yet, the tried-and-true means of feeding that appetite has ceased to be a design alternative [Kogge11]. The information technology industry has turned to massively parallel solutions with certain optimizations, but the fundamental conflict captured by Amdahl's Law places practical limits on this approach. Ultimately, these practical limits banish important applications to a low-performance ghetto if these valuable and helpful applications ever even see the light of day. As well, massively parallel solutions in their inefficiency dissipate ever more watts for diminishingly fewer added results as developers try to push out the range of these massively parallel solutions. Product development organizations attempting to extend the range of these massively parallel systems face a tough choice: risk sales by disappointing customers with unremarkable while improved products [Clark14] or risk a big development budget and reputation by going for truly differentiating features that may or may not be reachable at all within production constraints. Server virtualization has helped, but as organizations attempt to build on these successes they encounter still more roadblocks on this complicated and unpredictable path [WSJ13]. Meanwhile, the world moves on as climate change issues work their way to the forefront every day threatening a little more the supply of inexpensive electricity while the battery business finds itself increasingly subject to complications as it struggles to find the next step after lithium ion technology.

Development organizations large and small find refuge in heterogeneous computing as many have already seized this opportunity. Section 4 enumerates dozens of examples of hardware-accelerated features implemented across diverse application domains through the last 15 years. Intel not only partici-

pates but strategically integrates co-processor technology into their corporate vision [Courtland13]. Advancing technologies alluded to in section 2.2 enable these recent heterogeneous computing machines.

Despite the application performance advantages and dramatically lowered power consumption, heterogeneous computing deployment spreads slowly and selectively even as more and more organizations embrace the possibilities.

Challenges present stiff headwinds to broader acceptance. The language barrier has been addressed with OpenCL [Open14]. There's a growing collection of libraries enabling application developers to quickly get to market [Demler12], [Hiller13]. The lack of a platform standard places constraints on such libraries, compromising the value of these libraries and hindering a broader market that would encourage greater library development. Heterogeneous computing elements are either single-tasked or operate in a stateless mode, severely limiting the functional range and frequently ruining the operational efficiency of the applied heterogeneous computing components. Without the benefit of operational efficiency the power consumption savings dissipate. Restricted range limits the level of functionality packed into the application code in diametric opposition to the users' cravings. The lack of libraries pushes out the time to market and may decimate the performance advantages as functions are repeatedly re-invented, encouraging deployment of sub-optimal design. Together, these factors push the heterogeneous computing result back within reach of the legacy microprocessor platform, leaving insufficient reason for product development organizations to migrate from the microprocessor running software.

A standard platform catalyzes growth and the final realization of all of the promises made by heterogeneous computing.

There are high expectations for such a standard platform. To fully keep the promise, a heterogeneous platform must transparently scale meeting Amdahl's challenge along the way. Waste in the form of stalled pipelines that dissipate heat without producing a meaningful result dissolves the reason for committing to heterogeneous computing. Optimization efforts to resolve these conflicting forces result in a complex of more than one individually NP-complete problems (e.g., [Wang96]).

Heterogeneous computing systems routinely manage resources by software means at the sacrifice of versatility. While software control manages SIMD solutions, these dispatch in a highly structured manner dramatically simplifying scheduling via rigid constraints. Current software-managed OpenCL solutions generate code for a single FPGA on an attached PCI card. While multiple copies of these PCI cards may plug into the same motherboard, programming ignorant of additionally available resources can not take advantage for the application's sake, seriously crippling if not eliminating any possibility of scaling.

Software will struggle to meet this management challenge while keeping the promise of heterogeneous computing. There are numerous heuristics available for coping with an NP-complete problem, but depending on the specific optimization consume a great deal of compute time. Dependency-rich optimizations can thwart multi-core approaches via Amdahl's Law.

Well-understood array computing applications (e.g., applications written around LAPACK) with their relatively simple-to-route often rigid topologies set the heterogeneous computing scalability baseline, while representing a narrow cross section of the possible application space. As the dependencies complicate the task graph, the serial portion of the application grows accordingly dimin-

ishing the value of systems with static interconnect topologies. Time-to-market demands compromise efforts to efficiently map the application coding solution to a system with a static interconnect topology, realistically pushing the Amdahl's Law ratio heavier into the serial component. Still, inherently serial computations can be pipelined which is just another form of parallel computing, offering relief from performance constraints. Certainly, the components and materials exist to build an efficient high performance parallel pipelined computing system. However, continuing the upward performance march for the generic computer application while adequately managing the power consumption remains an open problem. An important issue is extracting the efficiency from an interworking collection of these components and materials.

An example illustrates this point. Assume a scalable system with five widely-varying parameters and five or six degrees of freedom contributing to the ultimate scoring or evaluation. Let the variable parameters of the model scalable system be:

- Free logic resources
- Access ports
- Schedule slots on data transport facility
- Buffers
- Database of application fragments

An application is a circuit that can be partitioned into different combinations of fragments. The object of the model is to fit the partitioned application into the available free logic resources and interconnect these fragments using access ports attached to the transport facility. Buffers decouple the application fragments on either end of a connection. Each of the items on the above list is limited in quantity. Assume that the application logic is fragmented in more than one way and that the fragment size is allowed to vary.

Let the measureable, interesting degrees of freedom for an assembled application be:

- Scheduling cycles
- Running power consumed
- System computation efficiency
- Minimum compute time
- Median compute time
- Maximum compute time

A scheduling cycle includes an active period and its subsequent suspend period, and helps determine the overhead incurred across the executing lifetime of the application. Running power consumed measures the amount of power used by the system under the conditions of a non-empty application schedule. System computation efficiency measures the contribution of available programmable logic resources to either control plane conditional decisions (e.g., if/else data

path routing) or data plane (e.g., intermediate or final values). Stability of the efficiency measure is especially important as the system scales for both power consumption and overall performance reasons. The compute time represents measurement of the amount of time it takes to calculate a set of results for an arbitrary application. Minimum and median compute times vary but maximum compute time is subject to a hard limit.

A dynamic mix of applications both constrain and complicate the optimization. As a heterogeneous system scales, a software resource management scheme soon struggles.

Assume the process of scheduling an allocation involves generating a series of configurations to an application, evaluating each configuration against the others, then accordingly committing the resources for an upcoming scheduling interval. Each configuration is an ordered and interconnected combination of application fragments summing to the whole application function. Assume that the amount of time required to generate a configuration can exceed 10 microseconds for a single state-of-the-art processor core running software. It would take such a state-of-the-art processor core at least a few milliseconds to compute an efficient placement for a modest system. A preferable scheduling solution considers an extended scheduling span, accumulating a score over multiple schedule slot instances stabilizing the solution against overall goals. Assume that a schedule slot is 10 to 20 milliseconds and the evaluation window is 300 to 700 milliseconds. Depending on the exact mix of applications heuristics can easily multiply the optimization time required by more than the factor of ten-or-less required to meet the time slot limit.

Stability requires that the scheduler produce a scheduling solution in less time than the duration of a schedule time slot to avoid stalling of the application. Stalling the application diminishes the efficacy of the parallel computing platform as it pushes the overall application compute time out. In order to acceptably meet real-time scheduling constraints, the resource allocation efficiency standard limits system scale. At the lower limit, the application consists of a single monolithic image requiring a single computing target (e.g., only one GPU or one FPGA) that's attached directly to the microprocessor driving the associated software plant. At any scale greater than the minimum, efficient use of resources involves optimizing the use of distributed compute resources that are interconnected with a constrained set of facilities. As alluded to earlier, such optimizations quickly become highly compute intensive.

Therefore, high-performance resource management is the key to liberating the potential of heterogeneous computing because the application range of a heterogeneous computing platform is proportional to the realistic scalability of such a heterogeneous computing platform. Appropriate resource management coupled with appropriate supporting system resources drives dynamic synthesis of a broader range of high-performance application pipelines that can satisfy the constraints of serial computation while providing generous throughput at an attractive power consumption. Combined with array computing techniques, high-performance resource management addresses both parts of the Amdahl's Law ratio, unlocking the real opportunity before us.

4 Applications

Demanding throughput- and computation-intensive applications suit heterogeneous computing architectures the best. It's shown that hardware assisted functionality has broad horizontal penetration. Designers have been implementing hardware acceleration functions throughout the history of modern electronics, and FPGAs and DSPs in particular have satisfied a gradually wider niche of functionality. Developers have created significant applications for the following product application areas:

- High-Performance Computing [Khasg13] [Reporter08] [Sanch11]
- Equipment [Li13] [Bend13] [Li11]
- Tools [Dufour12] [Rose11] [Kiran13]
- Transportation [Xico12] [Skup13]
- Consumer Applications [Lang09] [Niit10]
- Mobile Devices [Bsoul13] [Gudis12]
- Government [Ratsaby10] [Magis13] [Dilek13] [Pingree08]

5 Solution

AAXE^{TM} (Adaptable Architecture eXecution Environment) operating system for heterogeneous computing independently and autonomously manages generic collections of heterogeneous computing resources for highest efficiency and highest productivity. AAXE^{TM} is the registered trademark for the AAXE system. AAXE O/S seamlessly integrates arbitrary combinations of diverse implementation content, providing precision delivery of functionality most efficiently across a broad scaling range using commercially available off-the-shelf components and industry standard tool chains. In doing so, AAXE O/S facilitates optimum application performance while precipitously driving down total cost of ownership and development cycle time. The keys to the AAXE O/S value-add are these

- High-Performance hardware-realized resource management
- Modular standardized interfacing
- Fast-response communication infrastructure

Applications or libraries written for AAXE O/S execute on AAXE systems of any scale point, protecting development investment. AAXE applications seamlessly integrate with software applications via straightforward modular standard interfaces. In summary, AAXE O/S delivers the right functionality in the right amount at the right time to generic application needs.

5.1 Developing an AAXE Application

Xcelemor engineers the AAXE O/S to manage bit streams. At design time such a bit stream has the form of built proprietary applications. The developer uses standard, familiar development tools to create debugged applications. The developer enables these applications for the AAXE system with the inclusion of the AAXE infrastructure libraries at build time. The AAXE infrastructure libraries consist of a collection of wrappers that present a very limited overhead to the overall application under development. Interface to software plant takes standard form of hardware driver to attached AAXE equipment. Specific access to an arbitrary AAXE application depends on such an application's architecture defined in great part by the register definition. AAXE application interface design process is identical to common application development practice. AAXE reserves a relatively narrow range of the register address space standardizing the location of important application information. The AAXE User's Manual that ships with the AAXE development package describes these details.

AAXE scheduling encourages segmenting and modularization of developed applications. For instance, if an AAXE application exhibits dependency between certain functions, such an AAXE application might be partitioned into two sub-applications with the first processing vectors into a set of intermediate results which are then fed forward to the second sub-application. This approach echoes software shell scripting and offers similar benefits including easier verification/hardening, greater reuse prospects, simplified maintenance, and faster time to market. Since AAXE manages parallel hardware resources, breaking bigger applications into a collection of smaller applications presents a significant packaging advantage as well as enables AAXE more placement opportunity into systems that are inevitably fragmented.

5.2 Executing an AAXE Application

AAXE manages heterogeneous resources, greatly increasing efficiency and driving down the cost of ownership. Applications run at the whim of the users of the system, starting and stopping at arbitrary times. Heterogeneous applications take advantage of parallel computing techniques, such as deep pipelining and array processing, introducing complicating spatial aspects to resource management. As referenced in section 3, these and other constraining realities demand a potent approach to treatment as the bridge to general-purpose heterogeneous computing. The AAXE scheduler addresses the peculiar needs of generalized parallel applications by optimizing at any moment not just the instantaneous mix of parallel applications but also the placement of whatever currently active parallel applications with intrinsic and extrinsic consequences.

Intrinsically AAXE scheduling balances the needs of the users with the realities of the constituent heterogeneous technologies, as embodied by both the employed components and the AAXE supporting infrastructure. AAXE continuously optimizes to objectives via dedicated hardware. Prioritization extracts maximum leverage from dynamic scoping into both the instantaneous user needs and time/space scheduling possibilities, at the same time keying operations. The optimization plant scales with the AAXE deployment, ratcheting up management capacity in tandem with increasing application execution capacity. In an AAXE system, the resource management functions are actually AAXE applica-

tions though only system administrators have access to these AAXE resource management functions. A symbiotic feedback loop exists between the user space and the system space in that when instantaneous conditions dictate, system management functions can assume top priority devoting additional resources to retiring issues. In this relationship, AAXE system performance gracefully degrades under overload, offering the users the chance to intervene with their own remedial prescriptions, such as sending warning messages out to offending users that have overstepped their allocation.

Extrinsically AAXE scheduling takes maximum advantage of targeting opportunities. Applications that wait for input vectors suspend. Applications that suddenly surge can petition for momentarily available resources, dynamically flexing the system at no impact to the other users who may not at the moment require such spare resources. A study under way shows that the AAXE O/S facility for aggregating resources enables application designs that far exceed the capacity of even the most expensive ASICs. AAXE scheduling enables developers to realize multiple configurations of a function and then in real time activate the most opportune choice. For instance when an application employs iterative compute machinery to calculate results, as the quantity of resources increases unrolling these iterative functions increases application throughput beyond the sum of throughputs contributed by a collection of application instances done the iterative way. In a moment, based on conditions at the time AAXE can choose to devote more resources to this higher throughput aggregated solution supporting more users for that time or clearing the function from the schedule sooner by producing the results more promptly. In summary, flexible scheduling directly impacts system availability by addressing application needs in the context of the entire user load.

6 Biography

Peter. J. Zievers is the Chief Technical Officer at Xcelemor, Inc. His research interests include analysis and design of heterogeneous computing systems, resource management for computing systems, and high-performance digital communications for parallel distributed computing systems. He received his BS in electrical engineering from the University of Wisconsin at Madison and his master of science and PhD. degrees in electrical engineering from the University of Texas at Austin. At AT&T and Lucent Technologies, he participated in central office design, ATM and ethernet transport network switch design and wireless system development, with responsibilities including chip and card design, system cabling, ethernet subsystem software, and system diagnostics. Before starting up Xcelemor, he consulted on software projects spanning broadband wireless, secured radio, personal messaging, and broadband infrastructure switching systems, responsibilities including diagnostics, 3^{rd} party feature package integration, and system verification. This effort is dedicated to Mary Hoeffner, grandmother of the author.

References

[Bawab02] , El-Bawab, T.S. & Jong-Dug Shin *Optical packet switching in core networks: between vision and reality,* **IEEE Communications Magazine,** vol. 40, no. 9, September, 2002, pages 60-65

[Broc14] , *Brocade 10GBPS Tunable DWDM 80 KM Tunable SFP+ Optical Transceiver*, ordering part number 10G-SFPP-ZRD-T

[Fini14] , *Optical Components 850nm, 10 Gbps VCSEL Array*, Finisar part number V850-209x-002, **http://www.finisar.com/products/optical-components/VCSEL-and-Detectors/V850-209x-002**

[Open14] , *OpenCL: The open standard for parallel programming of heterogeneous systems,* **http://www.khronos.org/opencl/**

[NVID14] , *NVIDIA's Next Generation CUDA Compute Architecture: Fermi,* NVIDIA, 2009

[Schow10] , Schow, C.; Doany, F.; Kash, J. *Get on the Optical Bus* **IEEE Spectrum** September, 2010 page 33

[Zaz08] , Zazalan, M. *Intel: 80 cores by 2011* **Tech Freep: Daily News and Free Press** http://techfreep.com/intel-80-cores-by-2011.htm September 26, 2008

[Anth12] , Anthony, S. *Intel unveils 72-core x86 Knights Landing CPU for exascale supercomputing* **ExtremeTech** http://www.extremetech.com/extreme/171678-intel-unveils-72-core-x86-knights-landing-cpu-for-exascale-supercomputing November 26, 2012

[Altr13] , *The Breakthrough Advantage for FPGAs with Tri-Gate Technology* **Altera White Paper** WP-01201-1.0, June, 2013

[Leib13] , Leibson, S. and Mehta, N. *Xilinx UltraScale: The Next-Generation Architecture for your Next-Generation Architecture* WP435 version 1.0, July 8, 2013

[TI12] , *A better way to cloud: TI's new KeyStone multicore SoCs revitalize cloud applications, enabling new capabilities and a quantum leap in performance at a significantly reduced power consumption* http://www.multivu.com/mnr/54044-texas-instruments-keystone-multicore-socs-revitalize-cloud-applications November 13, 2012

[Achx13] , *Speedster22i HD FPGA Family Data Sheet,* Achronix Semiconductor Corporation, Document number DS004 revision 2.4, August 15, 2013

[Kogge11] , Kogge, P. *Next Generation Supercomputers*, **IEEE Spectrum** http://spectrum.ieee.org/computing/hardware/nextgeneration-supercomputers/0, January 26, 2011

[Winkler12] , Winkler, R. *Oracle's Little Problem with Big Data,* **The Wall Street Journal** http://online.wsj.com/news/articles/SB10001424052702304587704577333823771344922, April 9, 2012

[Netcraft13] , *January, 2014 Web Server Survey* **Netcraft** http://news.netcraft.com/archives/category/web-server-survey/

[WSJ13] , *Five New Virtualization Challenges Impacting IT Pros and Data Center Management,* **The Wall Street Journal** http://online.wsj.com/article/PR-CO-20130822-905204.html August 22, 2013

[Courtland13] , Courtland, R. *What Intel's Xeon Phi Coprocessor Means for the Future of Supercomputing,* **IEEE Spectrum** http://spectrum.ieee.org/semiconductors/processors/what-intels-xeon-phi-coprocessor-means-for-the-future-of-supercomputing July 24, 2013

[Reporter08] , Reporter, V. *The Value of a Millisecond: Finding the Optimal Speed of a Trading Infrastructure*, Tabb Group paper number V06:007, April, 2008

[Niit10] , Niitsuma, H and Maruyama, T.*Sum of Absolute Difference Implementations for Image Processing on FPGAs,* **Field Programmable Logic and Applications (FPL), 2010 International Conference on** pages 167-170 Print ISBN: 978-1-4244-7842-2, August 31, 2010

[Bend13] , Bendaoudi, H. and Khouas, A.*Stereo vision IP design for FPGA implementation of obstacle detection system,* **Systems, Signal Processing and their Applications (WoSSPA), 2013 8th International Workshop on** pages 145-150 May 12, 2013

[Lang09] , Lange, H.; Stock, F. ; Koch, A. ; Hildenbrand, D. *Acceleration and Energy Efficiency of a Geometric Algebra Computation using Reconfigurable Computers and GPUs,* **Field Programmable Custom Computing Machines, 2009. FCCM '09. 17th IEEE Symposium on** pages 255-258 Print ISBN: 978-0-7695-3716-0 April 7, 2009

[Bsoul13] , Bsoul, A.A.M.; Hoskinson, R. ; Ivanov, M. ; Mirabbasi, S. ; Abdollahi, H. *Implementation of an FPGA-based low-power video processing module for a head-mounted display system,* **Consumer Electronics (ICCE), 2013 IEEE International Conference on** pages 214-217 Print ISBN: 978-1-4673-1361-2 January 11, 2013

[Sanch11] Sanchez-Roman, D.; Sutter, G. ; Lopez-Buedo, S. ; Gonzalez, I. ; Gomez-Arribas, F.J. ; Aracil, J. *An Euler solver accelerator in FPGA for computational fluid dynamics applications,* **Programmable Logic (SPL), 2011 VII Southern Conference on** pages 149-154 Print ISBN: 978-1-4244-8847-6 April 13, 2011

[Li13] , Li, J.; Chen, Y.; Ho, C.; Lu, Z. *Binary-tree-based high speed packet classification system on FPGA,* **Information Networking (ICOIN), 2013 International Conference on** pages 517-522 Print ISBN: 978-1-4673-5740-1 January 20, 2013

[Li11] , Li, X.; Jing, X. *FPGA based mixture Gaussian background modeling and motion detection,* **Natural Computation (ICNC), 2011 Seventh International Conference on** pages 2078-2081 Print ISBN: 978-1-4244-9950-2 July 26, 2011

[Dufour12] , Dufour, C. ; Cense, S. ; Yamada, T. ; Imamura, R. ; Belanger, J. *FPGA permanent magnet synchronous motor floating-point models with variable-DQ and spatial harmonic Finite-Element Analysis solvers,* **Power Electronics and Motion Control Conference (EPE/PEMC), 2012 15th International** pages LS6b.2-1-LS6b.2-10 Print ISBN: 978-1-4673-1970-6 September 4, 2012

[Rose11] , Rose, A.V.V. ; Seshasayanan, R. ; Oviya, G. *FPGA implementation of low latency routing algorithm for 3D Network on Chip ,* **Recent Trends in Information Technology (ICRTIT), 2011 International Conference on** pages 385-388 Print ISBN: 978-1-4577-0588-5 June 3, 2011

[Kiran13] , Kiran, D.C. ; Misra, J.P. ; Yashas, D. ; Gurunarayanan, S. *Integrated scheduling and register allocation for multicore architecture,* **Parallel Computing Technologies (PARCOMPTECH), 2013 National Conference on** pages 1-7, Print ISBN: 978-1-4799-1589-7 February 21, 2013

[Xico12] , Xicotencatl-Perez, J.M. ; Lezama-Leon, A. ; Liceaga-Ortiz-De-La-Pena, J.M. ; Hernandez-Terrazas, R.O. *Real Time Stereo Vision with a modified Census transform in FPGA,* **Electronics, Robotics and Automotive Mechanics Conference (CERMA), 2012 IEEE Ninth** pages 89-94 Print ISBN: 978-1-4673-5096-9 November 19, 2012

[Skup13] , Skup, K.R.; Grudzinski, P. ; Orleanski, P. ; Nowosielski, W. *A digital controller for satellite medium power DC/DC converters,* **Methods and Models in Automation and Robotics (MMAR), 2013 18th International Conference on** pages 566-571 Print ISBN: 978-1-4673-5506-3

[Gudis12] , Gudis, E.; van der Wal, G. ; Kuthirummal, S. ; Chai, S. *Multi-Resolution Real-Time Dense Stereo Vision Processing in FPGA* **Field-Programmable Custom Computing Machines (FCCM), 2012 IEEE 20th Annual International Symposium on** pages 28-32 Print ISBN: 978-1-4673-1605-7 April 29, 2012

[Pingree08] , Pingree, P.J. ; Scharenbroich, L.J. ; Werne, T.A. ; Hartzell, C. *Implementing Legacy-C Algorithms in FPGA Co-Processors for Performance Accelerated Smart Payloads,* **Aerospace Conference, 2008 IEEE** pages 1-8 Print ISBN: 978-1-4244-1487-1 March 1, 2008

[Ratsaby10] , Ratsaby, J.; Zavielov, D. *An FPGA-based pattern classifier using data compression,* **Electrical and Electronics Engineers in Israel (IEEEI), 2010 IEEE 26th Convention of** pages 320-324 Print ISBN: 978-1-4244-8681-6 November 17, 2010

[Magis13] , Magistretti, E.; Gurewitz, O. ; Knightly, E. *802.11ec: Collision Avoidance Without Control Messages,*, **Networking, IEEE/ACM Transactions on** Volume PP, issue 99, page 1, November 13, 2013

[Dilek13] , Dilek, S.M; Ayranci, A. ; Ata, R. ; Ceylan, O. ; Bulent Yagci, H. *Radio trasmitter design including FPGA for micro/nano satellite,* **Signal Processing and Communications Applications Conference (SIU), 2013 21st**, pages 1-4 Print ISBN: 978-1-4673-5562-9 April 24, 2013

[Khasg13] , Khasgiwale, R.; Krnan, L. ; Perinkulam, A. ; Tessier, R. *Reconfigurable data acquisition system for weather radar applications,* **Circuits and Systems, 2005. 48th Midwest Symposium on** Volume 1, pages 822-825 Print ISBN: 0-7803-9197-7, August 7, 2005

[Betker97] , Betker, M.R.; Fernando, J.S.; Whalen, S.P. *The History of the Microprocessor,* **The Bell Labs Technical Journal** pages 29-56, Autumn, 1997

[Mims10] Mims, C. *Why CPU's Aren't Getting Any Faster,* **MIT Technology Review** http://www.technologyreview.com/view/421186/why-cpus-arent-getting-any-faster/ October 12, 2010

[Patt10] Patterson, D. *The Trouble with Multicore,* **IEEE Spectrum** pages 28-32, 52-53 July, 2010

[Gelenbe88] Genenbe, E. *Multiprocessor Speedup, Amdahl's Law, and the Activity Set Model of Parallel Program Behavior,* **RIACS Techncial Report** Report number 88.37 Research Institute for Advanced Computer Science, NASA Ames Research Center, December, 1988

[Lohr12] Lohr, S. *For Impatient Web Users, an Eye Blink Is Just Too Long to Wait,* **The New York Times** http://www.nytimes.com/2012/03/01/technology/impatient-web-users-flee-slow-loading-sites.html?pagewanted=all&_r=0 February 29, 2012

[Equat11] Equation Research *What Users Want from Mobile,* **Equation Research for Compuware** July, 2011

[Venka13] Venkatraman, A. *Users want reduced cloud complexity and more interoperability, study finds,* **ComputerWeekly.com** http://www.computerweekly.com/news/2240186451/Users-want-reduced-cloud-complexity-and-more-interoperability-study-finds June 19,2013 16:33

[Sween03] Sweeney, L. *Information Explosion,* Spring 2003

[McIlr10] McIlroy, T. *The Information Explosion (and it's implications to the Future of Publishing),* August 15, 2010

[Turner08] Turner, M.J. *IBM Information Infrastructure Aims to Tame the Information Explosion,* **Enterprise Strategy Group** September, 2008

[Little98] Little, J. *Evergreen 486 to 586 Upgrade Processor,* **Linux Journal**, Issue number 52, August 1, 1998

[Versace13] Versace, C. *What Do Consumers Want In A New Smartphone?,* **Forbes** http://www.forbes.com/sites/chrisversace/2013/08/21/what-do-consumers-want-in-a-new-smartphone/ August 21, 2013

[Knight13] Knight, K. *Report: For mobile, consumers want better batteries, processors* **BizReport** http://www.bizreport.com/2013/03/report-for-mobile-consumers-want-better-batteries-processors.html March 14, 2013

[Saginor12] Saginor, J. *J.D. Power: Consumers most dissatisfied with smartphone battery life,* **Digital Trends** http://www.digitaltrends.com/mobile/j-d-power-consumers-most-dissatisfied-with-smartphone-battery-life/ March 15, 2012

[Fitz13] Fitzgerald, M. *The Future Requires (Better) Batteries,* **The Wall Street Journal** http://online.wsj.com/news/articles/SB10001424052702304066404579125791002529378 November 11, 2013

[Tahil07] Tahil, W. *The Trouble with Lithium: Implications of Future PHEV Production for Lithium Demand,* **Meridian International Research** March, 2007

[Sacco13] Sacco, A. *iPhone evolution, timeline and notable moments,* **ComputerWorld** http://www.computerworld.com/slideshow/detail/119573/iPhone-evolution–timeline-and-notable-moments#slide1 September 12, 2013

[Mariano13] Mariano, K.D. *Top 5 Most Wanted iPhone 6, 5S Features; Consumers Demand More from Apple,* **International Business Times** http://au.ibtimes.com/articles/490645/20130716/iphone-6-5s-wanted-features-specs-rumors.htm#.UuFNyd3nZE5 July 16, 2013

[Glied13] Gliedman, C. *Calculating the Value of Faster Time-To-Market,* **CSO Data Protection** http://www.csoonline.com/article/219040/calculating-the-value-of-faster-time-to-market March 1, 2004

[Estrin07] Estrin, G. *Reconfigurable Computer Origins: The UCLA Fixed-Plus-Variable (F+V) Structure Computer,* **IEEE Annals of the History of Computing** Volume 24, Issue 4, pages 3-9, October-December, 2002

[Hamm08] Hamm, S. *It's Too Darn Hot,* **BusinessWeek** http://www.businessweek.com/stories/2008-03-19/its-too-darn-hot March 19, 2008

[Demler12] Demler, M. *MathWorks Introduces HDL Coder and Verifier For MATLAB,* http://tech.opensystemsmedia.com/eda/2012/03/mathworks-introduces-hdl-coder-and-verifier-for-matlab/ March 5, 2012

[Hiller13] Hiller, G. *Easy OpenCL with Python,* **Dr. Dobbs** http://www.drdobbs.com/open-source/easy-opencl-with-python/240162614 October 15, 2013

[Borer13] Borer, E.T.; Hillegas, C. W.; Kurtz, R., Mougey, A., Tatro, D.E.; Ziegler, J. *Building a Modern Computing Infrastructure at Princeton University,* **Educause Review Online** http://www.educause.edu/ero/article/building-modern-computing-infrastructure-princeton-university May 6, 2013

[Wang96] Wang, Z. and Crowcroft, J. *Quality-of-service routing for supporting multimedia applications*, **IEEE Journal on Selected Areas in Communications**, pages 1228-1234, Volume 14, Issue 7, September, 1996.

[Clark14] Clark, D. *NVIDIA Chip Offers Greater Performance,* **The Wall Street Journal** http://online.wsj.com/news/articles/SB10001424052702303949704579461440461821598 March 28, 2014