Extreme-Speed Scalable Direct Sparse Solvers for Heterogeneous Supercomputing –An Enhancement to the LAPACKrc Library

Juan Gonzalez[1] and Rafael C. Núñez

[1]Accelogic, 1830 Main Street, Suite 204, Weston, FL, U.S.A.

E-mail: juan.gonzalez@accelogic.com

Abstract. At the core of DOE's scientific priorities is a broad need for more computing power and supercomputing infrastructure. An estimated 70% of computing cycles spent globally in the HPC ecosystem are used to solve large-scale linear algebra computational problems. Linear algebra is at the core and constitutes the primary bottleneck of important DOE research problems in fusion energy, nuclear accelerator modelling, circuit simulation, and weather modelling, among other research challenges. As part of an ambitious research program co-funded by DOE, NASA, and the Department of Defense, Accelogic is spearheading the development of LAPACKrc, a groundbreaking family of FPGA-based linear algebra solvers able to achieve speedups larger than 100x with a single chip ("rc" stands for "reconfigurable computing," a radically new computing paradigm that is changing the way high performance computing is done today). Recent efforts on the LAPACKrc research program have focused on producing FPGA-based direct sparse solvers—a key functionality still missing in the current LAPACKrc solver suite. Our latest direct sparse solver prototype demonstrates a speedup of up to 125x (compared against state-of-the-art CPU direct sparse solvers), which provides support for broad-based science/engineering breakthroughs.

1. Introduction

Improvements in High-Performance Computing (HPC) have become one of the primary means of addressing the many critical needs the scientific community continuously faces. These improvements are more needed every day, as HPC supports the discipline of "modeling and simulation," now considered to be a fundamental part of the scientific method, replacing experimentation when the latter becomes prohibitive in terms of cost and/or difficulty. Energy, climate modeling, oil & gas exploration, health, and astrophysics, are among the numerous scientific areas always avid for increased computational power.

One approach to boosting computational capacity in large-scale codes while reducing costs is using heterogeneous architectures —*i.e.*, programming environments that involve different types of processor architectures operating simultaneously (*e.g.*, CPUs, FPGAs, GPUs, Cell Broadband engines). The authors in [1] consider "the emergence of multicore/manycore and heterogeneous architectures the single biggest change in scalable computing in the past decade, with effective use of these new node architectures critical to reducing the performance gap between the peak performance of the hardware and the realized performance of the applications."

Exploiting heterogeneous architectures for scientific applications is becoming mainstream. Supercomputing technologies such as Cray's XT5h [2] and SGI's RASC [3] have been in the market for several years now, and it is common to find commercial supercomputer components that allow the seamless addition of FPGA or GPU computing nodes to an existing supercomputing network - transparently. On the software side, powerful libraries and programming tools are either available or at an advanced development stage for von-Neumann heterogeneous systems (*e.g.*, multicore, GPUs, etc), as it is evidenced by the CUBLAS [4], PLASMA, and MAGMA [5] libraries.

For heterogeneous non-von-Neumann architectures (*e.g.*, systems enriched with FPGA accelerators), the development of software and numerical libraries is more challenging – yet potentially much more rewarding – and has been happening at a slower pace. FPGAs are specialized "programmable logic" devices able to execute specific routines several orders of magnitude faster than a traditional von Neumann CPU – while maintaining their entire (software based) programmability. In addition to quantum increases in speed, FPGA-systems are also "green," rendering between ten and hundreds of times more energy-efficiency than conventional CPUs.

FPGA computing technology provides a flexible framework that allows tackling **bottlenecks** not efficiently handled by von Neumann solutions. For example, simultaneously adapting arithmetic units for each of the bottlenecks in the computational chemistry code NWChem [5], is a task that can be executed better by flexible/adaptable supercomputing architectures. One of the methods used in NWChem is the Hartree-Fock (HF) method. The computational speed in this *ab initio* electronic structure algorithm is conditioned by two core functions, namely, Fock matrix formation and eigenproblem solution. The main bottleneck of the algorithm can be either the Fock matrix formation or the eigenproblem solution. When the code is executed on a small number of processors, matrix formation is the primary bottleneck, whereas, when the code is executed on thousands of processors, matrix formation is efficiently scaled thus shifting the primary bottleneck to the eigensolver (which is scalable up to about 300 processors today.) These two steps are inherently sequential, creating an opportunity for an FPGA-enhanced accelerated solution. In FPGA computing, instead of having a single computing architecture that scales very well and provides computing performance for only one of the two bottlenecks, we can have two different architectures alternating operations in the same FPGA chip. These two architectures can be optimized for the solution of each particular problem. In a different flavor of FPGA enhancement, it is possible to connect a small group of FPGA computing units into a traditional CPU-based supercomputing network, so that we take advantage of the scalability of the matrix formation in thousands of CPUs (currently achieved by NWChem), and of accelerated eigensolvers in a small number of FPGAs (a solution that is possible through the use of specialized FPGA numerical libraries), to achieve groundbreaking end-to-end acceleration. This is a perfect example of a real heterogeneous supercomputing architecture in action, enabled through adaptive computing software.

Aimed at providing the needed software substrate for the effective exploitation of FPGA accelerators in current supercomputers, Accelogic is developing LAPACKrc, a groundbreaking family of linear algebra solvers that increase computational speed 100+ times using hybrid CPU/FPGA computing [13], [14], [15]. In the remainder of this paper, we introduce recent advances in the development of this library, particularly focused on incorporating FPGA-accelerated direct sparse solvers into LAPACKrc.

2. The challenges of heterogeneous CPU/FPGA supercomputing

In order to achieve consolidation as a viable alternative that provides very-low-cost and energy-efficient HPC—at both the supercomputer and the workstation levels, FPGA computing has faced three fundamental challenges:

(a) Making available FPGA hardware platforms well-suited to scientific computing;

(b) Making available advanced programming tools and compilers for FPGA computing design, comparable to those currently used in traditional von Neumann design; and,

(c) Providing software and numerical libraries that are fast, accurate, portable, and easy to use, thus allowing for widespread adoption of FPGA acceleration by the numerical computing community, and by domain scientists.

Fortunately, with the evolution of FPGA computing during the last 10 years, most of these challenges are being successfully addressed. With respect to (a), supercomputer vendors already offer an impressive array of FPGA-based computer systems—most notably the SGI RASC [3], Cray's XT5h [1], the SRC-7 system [7], DRC Accelium Platform [8], and Convey HC-1 system [9]. As for (b), a significant number of programming tools and C-like compilers for FPGAs are already available both within the open-source (*e.g.*, SPARK [10]) and commercial (*e.g.*, Mitrion platform [11], Impulse C [12]) communities.

Most success in (c) has been achieved and is ubiquitous for integer-arithmetic applications such as encryption and bioinformatics. In the floating-point arena, LAPACKrc offers a powerful array of numerical libraries that exploit the presence of FPGAs in a supercomputer network, under a paradigm that provides maximal speed performance, but is yet portable and easy to use for the domain scientist.

3. Enabling heterogeneous supercomputing for scientific applications: The LAPACKrc numerical library

Sca/LAPACKrc (or simply LAPACKrc) is a family of FPGA-based linear algebra solvers able to achieve speedups larger than 100x per FPGA chip. The goals of the LAPACKrc program are to 1) develop the fundamental FPGA computing algorithmic technology for high-speed FPGA-enhanced dense and sparse linear algebra; 2) produce solvers and their underlying math components as portable and easy-to-use utilities; and 3) aggressively integrate the LAPACKrc technology into higher-level solutions for DOE and its other program sponsors. LAPACKrc incorporates the traditional linear algebra functionality of libraries like LAPACK, ScaLAPACK, PETSc, and MUMPS. The library is compatible with traditional CPU-based High-Performance Computing (HPC) hardware, and can be used to accelerate current HPC codes via plug-in enhancements with hybrid accelerators such as the SRC-7 or the Cray XT5h [13], [14], [15]. Figure 1 illustrates the core components of the LAPACKrc library. It is worth noting that, although initially focused on FPGA acceleration, LAPACKrc is designed to integrate and "adaptively" squeeze performance from all the components of a heterogeneous supercomputer (*e.g.*, CPUs, GPUs, FPGAs) when working together to solve a problem.

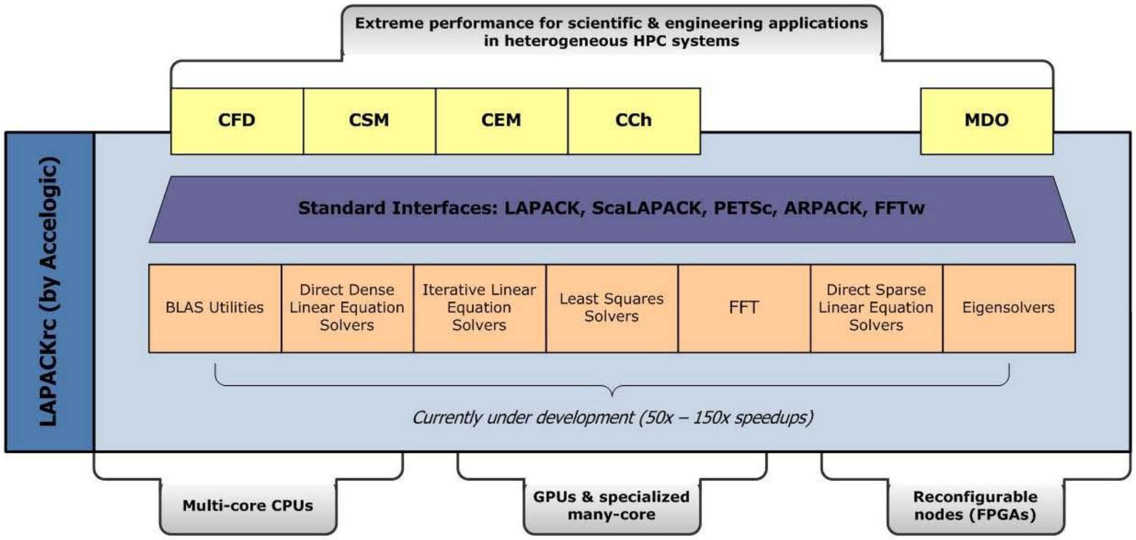
Extreme performance for scientific & engineering applications in heterogeneous HPC systems
CFD
CSM
CEM
CCh
MDO
LAPACKrc (by Accelogic)
Standard Interfaces: LAPACK, ScaLAPACK, PETSc, ARPACK, FFTw
BLAS Utilities
Direct Dense Linear Equation Solvers
Iterative Linear Equation Solvers
Least Squares Solvers
FFT
Direct Sparse Linear Equation Solvers
Eigensolvers
Currently under development (50x – 150x speedups)
Multi-core CPUs
GPUs & specialized many-core
Reconfigurable nodes (FPGAs)

Figure 1. Core components of the LAPACKrc library. LAPACKrc prototypes have demonstrated speedup factors of up to 150x for a variety of large-scale sparse and dense linear algebra problems.

4. Reducing time to solution in direct sparse computations

One of our most recent efforts in enhancing LAPACKrc has focused on incorporating direct sparse solver functionality. Our design is a collaborative architecture, with selected operations of the sparse direct solver executed in the CPU, and other numerically intensive kernels executed in the FPGA. Our sparse Cholesky solver is based on CHOLMOD [16]. CHOLMOD is used by MATLAB's 'A\b' operator when A is sparse symmetric positive definite. Our direct sparse computational engine is based on a many-core arithmetic unit (up to 1,640 cores per FPGA in our first prototype) for which operations are controlled by a dynamic scheduler.

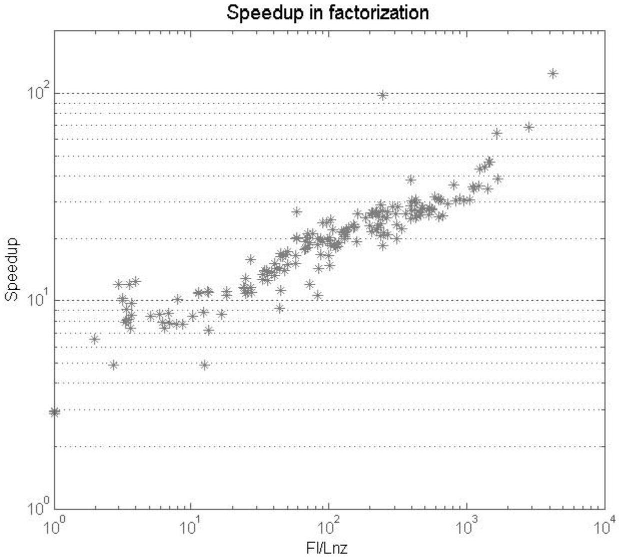

Figure 2. Speedup of LAPACKrc's sparse Cholesky factorization method, compared against CHOLMOD. The LAPACKrc solver runs on an Intel Xeon Woodcrest / Xilinx Virtex 6 system. CHOLMOD runs on an Intel Xeon Woodcrest @3GHz, on Ubuntu Linux.

We benchmarked our solver with all of the symmetric positive definite matrices in the University of Florida Matrix Collection [17], and attained speedups ranging from 3x for small matrices, up to 125x for large matrices. Figure 2 summarizes our results. We compared the performance of our solver against the fastest CPU solver (CHOLMOD [18]) running on the fastest CPU core (Intel Xeon Woodcrest [19]). The processing performance of our Cholesky solver (and of sparse Cholesky solvers in general) depends on the characteristics of the problem being solved. In general, the problem becomes more complex (*i.e.*, more time to solution) as the number of floating-point operations per nonzero entry in the factor L (of $A=LL^T$) increases. An important capability of this technology lies in its ability to provide more speedup as this ratio of flops per nonzeros increases.

Conclusions and future work

One approach to boosting computational capacity in large-scale codes while reducing costs is using heterogeneous architectures enhanced with accelerators such as FPGAs and GPUs. These types of systems have been in the market for several years now. Traditional supercomputers can be enhanced transparently by adding a number of accelerators able to boost performance "adaptively" for targeted numerical routines. The challenge in the proper exploitation of accelerator hardware is in well-designed software and numerical libraries that allow maximal speed and compatibility with the legacy CPU system, while maintaining portability and ease of use. Numerous efforts are starting to provide important software components, libraries, and development tools for accelerator-enhanced supercomputers. The development of FPGA-enhanced software and numerical libraries is more challenging than that of other accelerators – yet potentially much more rewarding in terms of overall speedup, cost, and power consumption. The LAPACKrc library represents the first industrial-quality family of linear algebra solvers intended for heterogeneous FPGA-enhanced HPC architectures, capable of speedups exceeding 100x per FPGA processor. The LAPACKrc library is currently integrating a large-scale sparse Cholesky factorization module that has reached up to 120x acceleration for symmetric/positive definite matrices in the sparse matrix collection. Future work includes, in addition to productization of the current prototype, the creation of additional methods for linear algebra.

Acknowledgments
This material is based upon work supported by the Department of Energy, U.S. Air Force, and NASA, under award numbers DE-SC-0001191, DE-FG02-07ER84728, DE-FG02-08ER85136, FA9550-09-C-0052, NNX07CA19P, and NNX10RA04C. We also wish to thank Tim Davis from the University of Florida and Jack Dongarra from the University of Tennessee for their consultation on this project.

Disclaimer
This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

References

[1] Geist A, Colella P, Heroux M 2008 Final Report *Workshop on CS/Math Institutes and High Risk/High Payoff Technologies for applications (2008)*

[2] Cray XT5h Supercomputer. http://www.cray.com/downloads/CrayXT5hBrochure.pdf (2008)

[3] SGI RASC Technology. http://www.sgi.com/products/rasc/ *(Sep 2006)*

[4] NVIDIA Corporation. CUBLAS Library Manual. Santa Clara, CA. *(Mar 2008)*

[5] Agullo E, et al 2009 Numerical linear algebra on emerging architectures: The PLASMA and MAGMA projects *Journal of Physics: Conference Series (Vol. 180, 2009)*

[6] Kendall R 2000 High performance computational chemistry: an overview of NWChem a distributed parallel application *(Computer Physics Communications 128: 260–283)*

[7] SRC-7 Overview. http://www.srccomputers.com/products/src7.asp. *(Feb 2008)*

[8] DRC Accelium. http://www.drccomputer.com/pdfs/DRC_Accelium_Overview.pdf *(2010)*

[9] The Convey HC-1: The World's First Hybrid-Core Computer. http://www.conveycomputer.com/Resources/HC-1%20Data%20Sheet.pdf *(2010)*

[10] Gupta R, Dutt N, Nicolau A 2004 SPARK: A Parallelizing Approach to the High-Level Synthesis of Digital Circuits. *Kluwer Academic Publishers*

[11] The Mitrion Platform. http://www.mitrionics.com, *(2010)*

[12] Impulse CoDeveloper C-to-FPGA Tools. http://www.impulsec.com/products_universal.htm.

[13] Veley D, Camberos J 2007 Computational Uncertainty in Reconfigurable Computing *NATO RTO Applied Vehicle Technology Panel (AVT) Symposium (Athens, Greece, Dec 2007)*

[14] Gonzalez J, Nunez R C 2008 LAPACKrc: A portable linear algebra library for extreme-speed acceleration of HPC codes using FPGAs *DoD High Performance Computing Modernization Program, 2008 Users Group Conference (Seattle, WA, July 14-18, 2008)*

[15] Gonzalez J, Nunez R C 2009 LAPACKrc: Fast linear algebra kernels/solvers for FPGA accelerators *Journal of Physics: Conference Series (Vol. 180, 2009)*

[16] Chen Y, et al 2009 Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate *ACM Trans. Math. Software (Vol 35, No. 3, 2009)*

[17] Davis T 1997 The University of Florida Sparse Matrix Collection, http://www.cise.ufl.edu/research/sparse/matrices *(NA Digest, vol. 97, no. 23, June 7, 1997)*

[18] Gould N, Hu Y, Scott J A. 2006 Complete results from a numerical evaluation of sparse direct solvers for the solution of large, sparse, symmetric liner systems of equations *NAG Internal Report 2005-1 (revision 2). Rutherford Appleton Laboratory (March 15, 2006)*

[19] Dongarra J. 2010 Performance of Various Computers Using Standard Linear Equations Software *University of Tennessee Computer Science Technical Report, CS-89-85, 2010*