Project 38: Accelerating Architecture Innovation into Fieldable Extreme-scale Systems (A cross-agency effort)

John Shalf[1], Dilip Vasudevan[1], David Donofrio[1], Anastasia Butko [1], Andrew A. Chien[23], Yuanwei Fang[23], Arjun Rawal[23], Chen Zou[23], Ray Bair[3], Kris Keipert[3], Arun Rodriguez[4], Maya Gokhale[4], Scott Lloyd[5], Xiaochen Guo[6], Yuan Zeng[6]

Lawrence Berkeley National Laboratory[1], University of Chicago[2], Argonne National Laboratory[3], Sandia National Laboratory[4], Lawrence Livermore National Laboratory[5], Lehigh University[6]

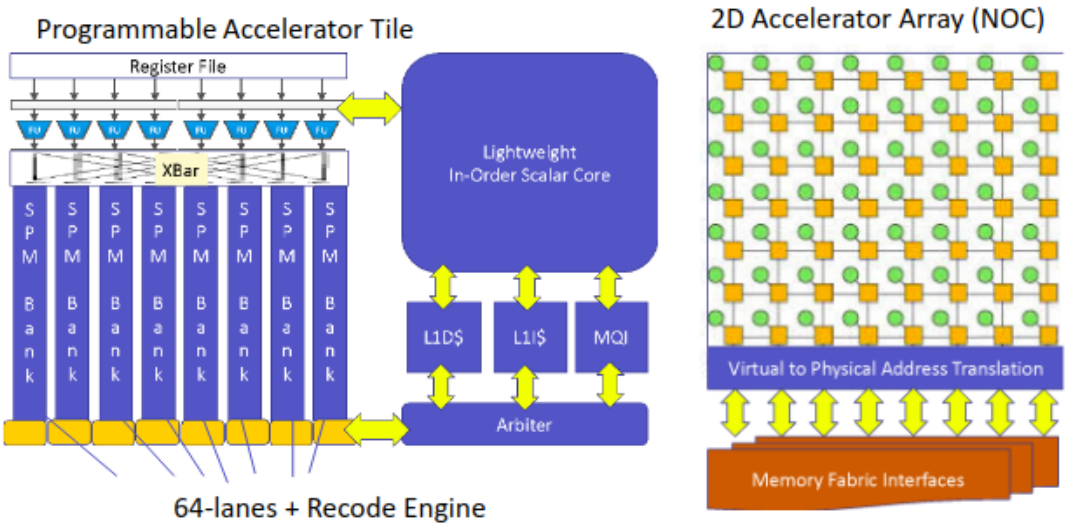

Figure 1: P38 Architecture (Message queues and DRE not pictured)

ABSTRACT

Accelerating technology disruptions and architectural change create growing opportunities and urgency to reduce the latency in for new architectural innovations to be deployed in extreme scale systems. We are exploring new architectural features that improve memory system performance including word-wise scratchpad memory, a flexible Recode engine, hardware message queues, and the data rearrangement engine (DRE). Performance results are promising yielding as much as 20x benefit. Project 38 is a cross-agency effort undertaken by the US Department of Energy (DOE) and Department of Defense (DoD).

ACM Reference Format:

John Shalf[1], Dilip Vasudevan[1], David Donofrio[1], Anastasia Butko [1], Andrew A. Chien[23], Yuanwei Fang[23], Arjun Rawal[23], Chen Zou[23], Ray Bair[3], Kris Keipert[3], Arun Rodriguez[4], Maya Gokhale[4], Scott Lloyd[5], Xiaochen Guo[6], Yuan Zeng[6] . 2019. Project 38: Accelerating Architecture Innovation into Fieldable Extreme-scale Systems (A cross-agency effort) . In *Proceedings of SC '19: The international conference for high performance computing networking storage and analysis (SC '19).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/1122445.1122456

SC '19, November 03–05, 2019, Denver, CO

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00

https://doi.org/10.1145/1122445.1122456

1 MOTIVATION AND GOALS

The short term goals of P38 are to quantify the performance value and identify the potential costs of specific architectural concepts against a limited set of applications of interest to both the DOE and DoD. Longer term goals include the development an enduring capability for DOE and DoD to jointly explore architectural innovations and quantify their value. Finally, a stretch goal for the effort is the specification of a shared, purpose-built architecture to drive future DOE-DoD collaborations and investments.

2 KEY ARCHITECTURE FEATURES

The team, including computer architecture, software, and applications researchers from numerous DOE laboratories and DoD innovation organizations have identified and are exploring the following key architectural features to address critical challenges in memory system performance. They include:

- **Customized Energy Efficient off-the-shelf IP:** such as existing RISC-V cores, or Fixed function units such as FFT

(Generated by SPIRAL), transpose, packet processing, encryption, and more.

- **Word Granularity Scratchpad Memory (SPM) and Flexible Gather-scatter:** both within a processor tile and across the chip. This both increases effective address/memory generation and enhances the effectiveness of SIMD execution.
- **Recoding Engine (aka Unstructured Data Processor or UDP):** provides programmable high speed data recoding, dealing with sub-word, variable-size symbol codes such as huffman effiiciently and at memory speeds. The UDP can handle branch-heavy code with average 20x improvement over processor core.
- **Atomic Message Queues:** that provide fast, directed synchronization and data sharing. Can be used to accelerate gather-scatter between processor tiles, and enable flexible parallelism across tiles without barrier overheads.
- **Data Rearrangement Engine (DRE):** provides efficient memory-side gather and presents the resulting data as a dense vector for memory interface and hierarchy

Collectively these features all focus on efficient data movement and information storage – the critical power and performance issues in modern computing systems.

3 EVALUATION

The key architectural features are being explored using a range of important applications, including: Sparse Matrix Trisolve, 1D Fast Fourier Transform, MerBench/HipMer (bioinformatics/genomics), Kripke - particle transport, HPGMG - geometric multigrid, XSBench - monte carlo neutronics, and graph pattern-matching. Each of the architectural features has been implemented at the level of RTL (or lower), and that design used to estimate power, area, and speed to calibrate the simulations. These studies show the potential benefits of the architectural features. Selected highlights include:

- Scratchpad memory and flexible gather scatter was found to yield significant performance and power benefits, improving the efficiency of data movement over cache-coherent memory hierarchies and enabling larger usable on-chip memories.
- Atomic message queues can reduce synchronization overhead by 4-8x compared to OpenMP/Atomics, and has particular benefits in applications such as sparse-matrix trisolve to increase the amount of exploitable parallelism.
- The Recode engine has been demonstrated to deliver 1.4-3.3x o the P38 selected representative sparse matrices and 3.4-6.7x for the full TAMU sparse matrix collection on reduction in memory bandwidth or increase in performance. Power benefits are also significant.
- Together message queues and Recode can deliver 20x improved scaling for SuperLU Trisolve
- DRE (data rearrangement engine) accelerated sparse vector gather, demonstrating benefits from 1.29 to 4.15x for a range of graph, image, and random-access computations.
- Fixed function accelerators, FFT demonstrated both performance and power benefits of 25-125x compared to a traditional CPU, showing the promise of a range of fixed function accelerators for future systems.

4 SOFTWARE PLAN

All of Project 38's novel architecture mechanisms can be accessed through intrinsics and libraries - enabling direct access by sophisticated applications. We plan to undertake a variety of efforts that exploit libraries for linear algebra, matrix operations, FFT's and more to enable applications. We also expect that enabling via high level frameworks (e.g. Kokkos, Raja, more) will be important.

Several require additional compiler support - UDP/Recode for data transformation libraries, DRE for memory side operation, and we are exploring prototype compilers (LLVM) and tools to enable large-scale application use. Further, we plan to build runtime and bootstrap tools to enable large-scale application use.

5 SUMMARY AND FUTURE WORK

The Project 38 effort has built a cross-agency architecture team exploring and validating the promise of key new architectural features for DOE and DoD workloads. These most promising of those features include:

- Scratchpad Memory (word-level)
- Recode/Unstructured Data Processor
- Gather-scatter hardware support (DRE)
- Atomic Message-queues

A variety of architectural evaluation, hardware design prototyping, software prototyping, application studies and technology transfer paths that will lead to integrated designs and prototypes are underway. In addition, ongoing innovative architecture feature refinement for greater performance and flexibility is also being explored.

We are exploring several SoC and vendor integration scenarios to make these architecture innovations fieldable. In addition, there is an ongoing study exploring Recode and DRE complementarity, on the compute and memory sides of the system and optimizing the bit-level and word-level arrangements of information in the system.

ACKNOWLEDGMENTS

Supported in part by Exascale Computing Project (ECP), Project Number: 17-SC-20-SC, a collaborative effort of the DOE Office of Science and the National Nuclear Security Administration. Support from the CERES Center for Unstoppable Computing and the Texas Advanced Computing Center (TACC) is also gratefully acknowledged.

REFERENCES

[1] Henry Cook, Miquel Moreto, Sarah Bird, Khanh Dao, David A Patterson, and Krste Asanovic. 2013. A hardware evaluation of cache partitioning to improve utilization and energy-efficiency while preserving responsiveness. In *ACM SIGARCH Computer Architecture News*, Vol. 41. ACM, 308–319.

[2] Yuanwei Fang, Chen Zou, and Andrew A Chien. 2019. Accelerating Raw Data Analysis with the ACCORDA Software and Hardware Architecture. *Proceedings of the VLDB Endowment* (2019).

[3] Yuanwei Fang, Chen Zou, Aaron J Elmore, and Andrew A Chien. 2017. UDP: a programmable accelerator for extract-transform-load workloads and more. In *2017 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 55–68.

[4] Farzad Fatollahi-Fard, David Donofrio, George Michelogiannakis, and John Shalf. 2016. OpenSoC Fabric: On-chip network generator. In *2016 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, 194–203.

[5] Daichi Fujiki, Aran Subramaniyan, Tianjun Zhang, Yu Zeng, Reetuparna Das, David Blaauw, and Satish Narayanasamy. 2018. Genax: a genome sequencing accelerator. In *Proceedings of the 45th Annual International Symposium on Computer Architecture*. IEEE Press, 69–82.

[6] Evangelos Georganas, Aydin Buluç, Jarrod Chapman, Leonid Oliker, Daniel Rokhsar, and Katherine Yelick. 2015. meraligner: A fully parallel sequence aligner. In *2015 IEEE International Parallel and Distributed Processing Symposium*. IEEE, 561–570.

[7] Maya Gokhale, Scott Lloyd, and Chris Hajas. 2015. Near memory data structure rearrangement. In *Proceedings of the 2015 International Symposium on Memory Systems*. ACM, 283–290.

[8] Scott Lloyd and Maya Gokhale. 2015. In-memory data rearrangement for irregular, data-intensive computing. *Computer* 48, 8 (2015), 18–25.

[9] Arjun Rawal, Yuanwei Fang, and Andrew Chien. 2019. Programmable Acceleration for Sparse Matrices in a Data-Movement Limited World. In *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 47–56.

[10] John Shalf, David Donofrio, Chris Rowen, Leonid Oliker, and Michael Wehner. 2011. Green flash: Climate machine (lbnl). *Encyclopedia of Parallel Computing* (2011), 809–819.