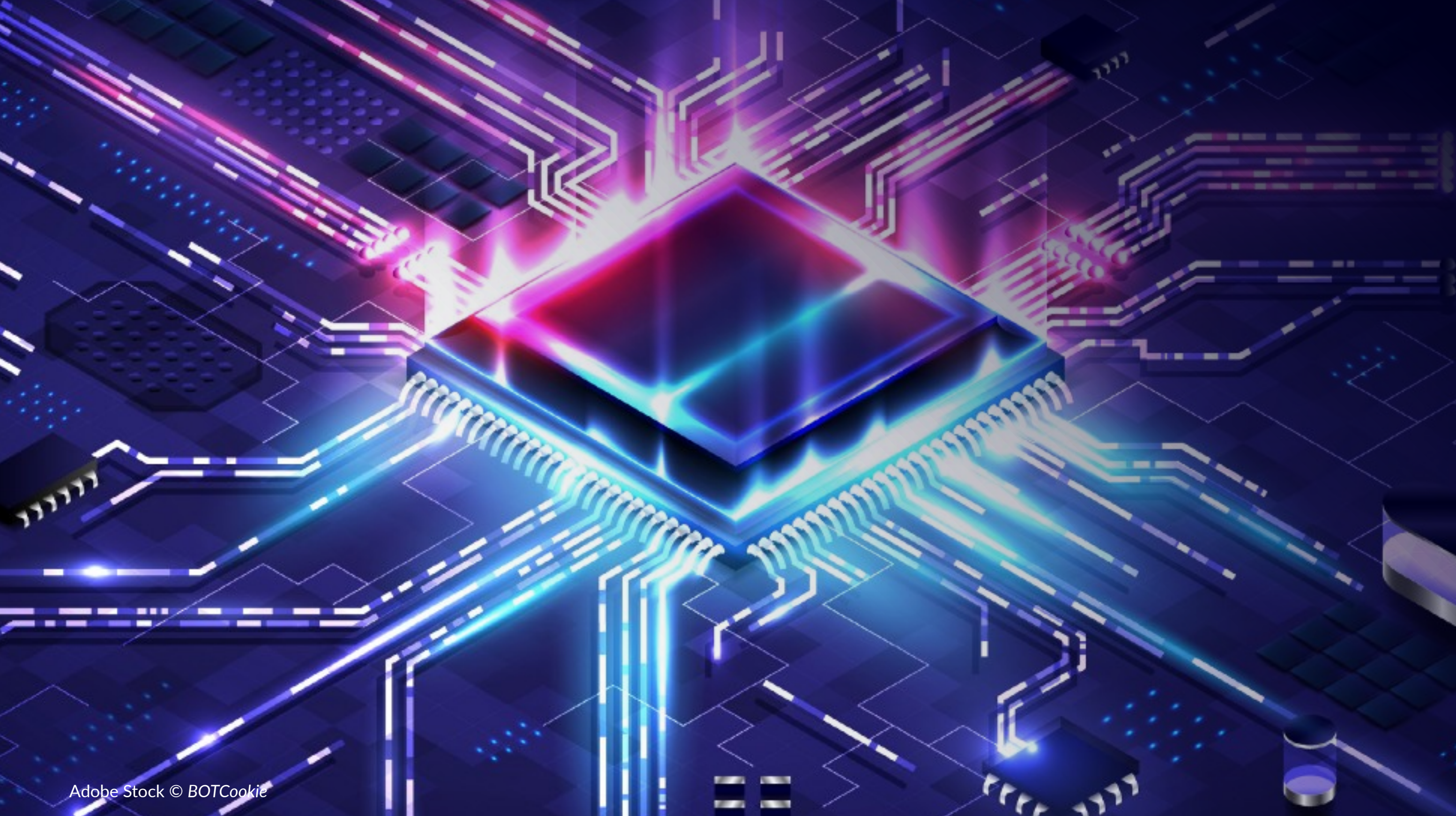
Towards faster, greener and easier to program computers

Alberto Ros
University of Murcia, Spain



Adobe Stock © *BOTCookie*

The increasing computational power of computers is essential for enabling many important emerging application domains such as big data, media, medical or scientific modelling. Today, the computers used in most digital devices, from smartphones to high-performance servers, include several processing units, providing more performance per watt than a large monolithic processing unit. These multi-core processors enforce a particular memory consistency model that defines how intuitive they are to program.

The ERC Consolidator Grant project ECHO—Extending Coherence for Hardware-Driven Optimizations in Multicore Architectures—aims to boost the performance and energy efficiency of future multi-core computers while keeping them easy to program or even making programming easier (Ros, 2020).

A fundamental technique to improve computer performance is *speculation*. This technique consists of executing work before it is known if it is actually needed or even correct. If speculation is right, important performance gains are obtained. However, a large penalty can be paid if the work executed speculatively is not useful. ECHO aims to remove the inefficiencies of speculation.

ECHO started in September 2019, and since then, its team of researchers has made substantial progress towards its goal. This article digs into key progress within the project, namely, improving the performance and efficiency of memory operations, improving the execution of atomic operations, and strengthening the memory consistency model while maintaining performance gains. Next, we elaborate on these aspects.

Fast and efficient memory operations with prefetch and fusion

Memory operations, e.g. loads and stores, are responsible for transferring data from memory to the processing unit and vice versa. These operations incur a long latency, mostly when the communication happens with the main memory. Prefetching mechanisms are commonly employed to reduce the latency of memory operations. These mechanisms bring in advance the data predicted to be used by the processor to the fast memory levels closer to the processor (e.g. the first-level cache).

The first contribution of ECHO in this regard is store-prefetch burst (Cebrian, Kaxiras, and Ros, 2020), a highly selective and very aggressive prefetching strategy for store operations. The following new insights motivated the proposal: (*a*) only a few stores in the application cause the majority of the performance penalties; (*b*) the access pattern of such stores is easily predictable, as most times they write to nearby memory locations; and (*c*) the latency of those stores is not commonly hidden by standard prefetching mechanisms, as it would require tremendous prefetch aggressiveness. Our prefetcher accurately detects contiguous store-access patterns and, in a single request to the first-level cache controller, it issues prefetches for the remaining data in a memory page. The impact of this proposal, which requires just 67 bits of storage, on future computers is twofold. On the one hand, with a limited structure, it holds 56 stores, a similar performance to an unlimited structure. On the other hand, the structure that holds the stores could be reduced by almost three times (from 56 to 20 stores) when implementing this technique while maintaining performance advantages. Thus, the energy consumption of one of the most consuming structures in the processor, the store buffer, is reduced.

But ECHO also made important progress for general data prefetching mechanisms. Berti (Navarro-Torres *et al.*, 2022) is a novel first-level data cache prefetcher that brings many advantages with respect to state-of-the-art prefetching mechanisms. Berti accounts for timeliness; prefetch requests are issued long ahead of the processor's access to that memory location, so when the access happens, the data is already present in the cache. Berti can prefetch complex access patterns thanks to the 'local delta' concept (distance between two arbitrary accesses made by the same instruction). It also tolerates a certain degree of out-of-orderliness in memory accesses, by keeping an access history and checking the previous accesses to detect deltas. Finally, it features a novel mechanism for precisely computing the accuracy of each discovered delta. The impact of Berti comes both in terms of performance and energy efficiency. With a storage of just 2.55 KB, Berti improves performance by 3.5 per cent and reduces the energy consumption of the memory hierarchy by 33.6 per cent compared to the prefetcher winner of the last prefetching championship, IPCP.

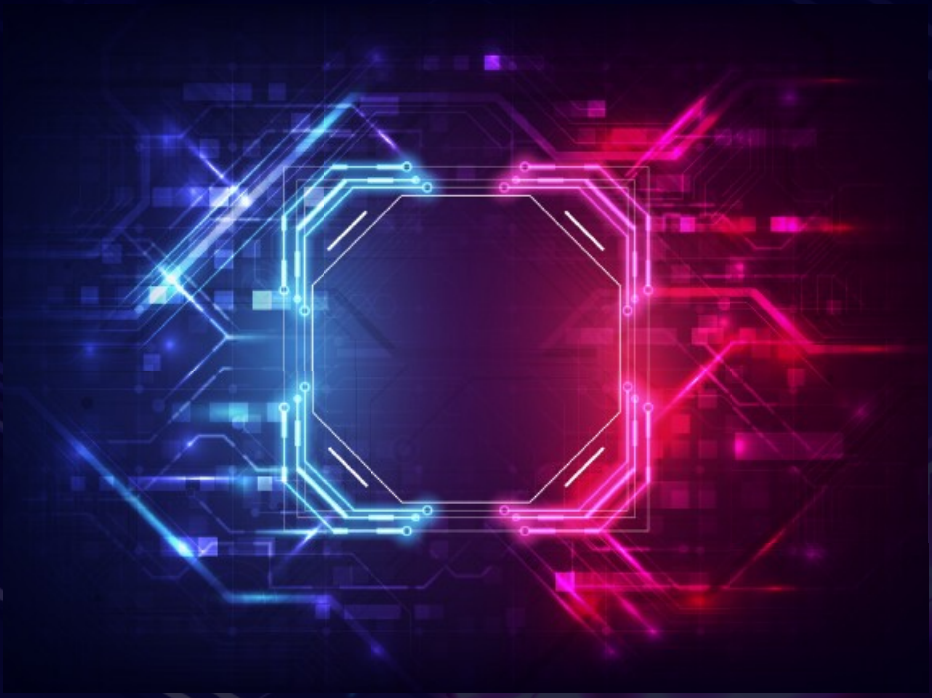
But apart from hiding the latency of memory operations, reducing the resources dedicated to the processor for memory operations is also important. To this end, a novel fusion mechanism for non-contiguous memory instructions, Helios (Singh *et al.*, 2022), has been recently proposed in the context of ECHO. Fusion allows two instructions to consume the resources of a single instruction, thus using those freed resources for extra computation. For the first time, Helios does fusion for non-consecutive instructions, using a precise prediction mechanism that detects potential pairs of memory instructions to fuse. The impact of Helios is mainly on performance, as it improves performance by 8.2 per cent over state-of-the-art consecutive fusion, yielding a total of 14.2 per cent performance uplift over not performing fusion at all.

Improving atomic operations by allowing concurrency

Atomic operations are indispensable to building correct parallel (multi-thread) programs. They are used, for example, to implement locks, which are software constructs that guarantee a thread of a parallel program to execute some instructions in isolation, i.e. atomically. Still, both atomic operations and lock constructs incur large performance penalties due to a lack of concurrency. ECHO has addressed these issues in two recently published works.

The first proposal is **multi-address atomic operations**, or in short, MAD atomics (Gómez-Hernández *et al.*, 2021). MAD atomics can replace lock constructs when they protect a small number of addresses. They perform fine-grained locking for multiple addresses, relying on a predetermined locking order. This way, concurrency is achieved with atomic operations in other threads when they target different memory addresses. Unlike other fine-grain solutions like hardware transactional memory, MAD atomics are non-speculative, bringing important performance and energy efficiency advantages. One of the main contributions of this work is the deep analysis of deadlock scenarios that come from being non-speculative and the solutions to prevent them. MAD atomics, requiring just 68 bytes of storage, can impact future computers by outperforming lock implementations by 3.4 times (and 2.7 times over hardware transactional memory), on average, when running concurrent data structures.

The second proposal, **free atomics** (Asgharzadeh *et al.*, 2022), involves removing the fences surrounding atomic operations. Fences are instructions that serialize memory operations, thus preventing their concurrent execution. This work makes the observation that for fences to be removed, the processor should: (*a*) add support for speculative atomic operation; (*b*) add support for concurrent atomic operations; and (*c*) resolve potential deadlocks. In addition, allowing the concurrent execution of atomic operations allows the forwarding of the data from one atomic operation to another in the processor pipeline without requiring communication through memory, thus speeding up even more atomic operations. With simple modifications and an additional small storage overhead of 15 bytes, free atomics improves performance by an average of 12.5 per cent for a large range of parallel workloads over a fenced atomic implementation and reduces energy consumption by a similar percentage. Free atomics was presented in the best paper session of the ISCA 2022 conference.



Adobe Stock © *BOTCookie*

Strong and fast consistency models

Memory consistency models are the contract between the programmer and the hardware designer. Thanks to that contract, the programmer knows the behaviour of a program, and the designer knows the tricks the hardware can do without breaking that contract. Stronger models mean more intuitive programming, but at the same time, it makes it harder for the hardware designer to achieve high performance.

Guaranteeing store atomicity is key for intuitive programming. However, most computers nowadays do not provide such a guarantee. A key step in ECHO towards programmability while keeping high performance is a speculative solution to enforce store atomicity (Ros and Kaxiras, 2020). This solution allows loads to read values from earlier stores even if the value has not been written to memory yet, but it makes it impossible for the programmer to notice it. The key insight of this work is that loads getting the data from earlier stores are not speculative, but younger loads are, until the value is stored in memory. The effective and cheap speculative proposal to dynamically enforce store atomicity provides the best of both worlds: (*a*) a more intuitive store-atomic memory model, as the one provided by IBM z-series processors; and (*b*) performance approaching (at an average of just 2.6 per cent) that of a non-store-atomic model, as the x86 model provided by Intel and AMD.

A follow-up of this work, ITSLF (Feliu *et al.*, 2021), enabled the communication of values from stores to loads across threads running in the same core in simultaneous multi-threading processors. The key insight of ITSLF is that communication across threads can be accelerated without affecting the memory consistency model. That is, the programmer is not able to detect such communication. The impact of ITSLF is to fasten data communication between threads, resulting in a 12 per cent performance improvement for communication-intensive applications. ITSLF was awarded an honourable mention at Micro TopPicks 2022, i.e. selected among the 24 more relevant computer architecture papers in 2021.

References

Asgharzadeh, A., Cebrian, J.M., Perais, A., Kaxiras, S. and Ros, A. (2022) 'Free Atomics: Hardware Atomic Operations without Fences', *49th International Symposium on Computer Architecture (ISCA)*, pp. 14–26. doi: 10.1145/3470496.3527385.

Cebrian, J.M., Kaxiras, S., and Ros, A. (2020) 'Boosting Store Buffer Efficiency with Store-Prefetch Bursts', *53rd International Symposium on Microarchitecture (MICRO)*, pp. 568–580. doi: 10.1109/MICRO50266.2020.00054.

Feliu, J., Ros, A., Acacio, M.E. and Kaxiras, S. (2021) 'ITSLF: Inter-Thread Store-to-Load Forwarding in Simultaneous Multithreading', *54th International Symposium on Microarchitecture (MICRO)*, pp. 1296–1308. doi: 10.1145/3466752.3480086.

Gómez-Hernández, E.J., Cebrian, J.M., Titos-Gill, R., Kaxiras, S. and Ros, A. (2021) 'Efficient, Distributed, and Non-Speculative Multi-Address Atomic Operations', *54th International Symposium on Microarchitecture (MICRO)*, pp. 337–349. doi: 10.1145/3466752.3480073.

Navarro-Torres, A., Panda, B., Alastruey-Benedé, J., Ibáñez, P., Viñals-Yúfera, V. and Ros, A. (2022) 'Berti: An Accurate Local-Delta Data Prefetcher', *55th International Symposium on Microarchitecture (MICRO)*, pp. 975–991. doi: 10.1109/MICRO56248.2022.00072.

Ros, A. (2020) 'Changing the Future in Computers', *The Project Repository Journal*, 6, pp. 126–128.

Ros, A. and Kaxiras. S. (2020) 'Speculative Enforcement of Store Atomicity', *53rd International Symposium on Microarchitecture (MICRO)*, pp. 555–567. doi: 10.1109/MICRO50266.2020.00053.

Singh, S., Perais, A., Jimborean, A. and Ros, A. (2022) 'Exploring Instruction Fusion Opportunities in General Purpose Processors', *55th International Symposium on Microarchitecture (MICRO)*, pp. 199–212. doi: 10.1109/MICRO56248.2022.00026.

PROJECT NAME

ECHO: Extending Coherence for Hardware-Driven Optimizations in Multicore Architectures

PROJECT SUMMARY

The ERC Consolidator Grant project ECHO (Extending Coherence for Hardware-Driven Optimizations in Multicore Architectures) aims to change the events that occur in multiprocessors such that predictions or decisions being made by the processing units will find the best possible outcome in the future. This way, large performance and energy improvements are expected in future computers leveraging ECHO concepts.

PROJECT PARTNERS

The ECHO project is based at the Faculty of Computer Science of the University of Murcia. The Computer Engineering department has a long tradition in computer architecture research and collaborates with several renowned research teams from Europe, America and Asia.

PROJECT LEAD PROFILE

Alberto Ros is a full professor in the Computer Engineering Department at the University of Murcia, Spain. Funded by the Spanish government to conduct his PhD studies, Ros received a PhD in computer science from the University of Murcia in 2009. He held postdoctoral positions at the Universitat Politècnica de València and Uppsala University. Ros received a European Research Council Consolidator Grant in 2018 to improve the performance of multi-core architectures. Working on cache coherence, memory hierarchy designs, memory consistency and processor micro-architecture, he has co-authored more than 100 peer-reviewed articles. Ros has been inducted into the ISCA Hall of Fame and MICRO Hall of Fame and is an IEEE senior member.

CONTACT

Alberto Ros
Facultad de Informática, Campus de Espinardo, 30100

☎ +34 868 888518

✉ aros@um.es

🌐 http://webs.um.es/aros

FUNDING

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No. 819134.