My research focuses on increasing the efficiency of server systems for datacenters to support sustainable growth of important global services during the environmental crisis. I seek to minimize the energy footprint and maximize the compute and storage density of server systems through better system integration, the use of emerging technologies, specialization of various system components for prevalent applications, and disciplined approximation. I believe in holistic approaches to application-system co-design, which include tighter integration of applications, systems and technologies, and rethinking of the established abstraction layers.

Previous Research

Understanding modern server workloads. I was fortunate to be part of a truly collaborative and fruitful effort towards micro-architectural and system-level characterization of modern datacenter workloads, which are scale-out in nature due to the scale of the services they provide. I am one of the primary architects of *CloudSuite* [1], the first publicly available open-source benchmark suite whose purpose is to help with the design and evaluation of scale-out server systems. Through a rigorous evaluation of CloudSuite, we demonstrated significant differences in micro-architectural behavior between modern server workloads and commonly used benchmarks [2][3][4][5], and we pointed out the big mismatch between the capabilities of available server platforms and the modern application needs. Our work uncovered research directions in server processor designs, highlighting the opportunities for processor specialization to match the needs of the cloud. An example of such specialized architecture is Scale-Out Processor [6], another highly collaborative effort I was part of, which improves processor efficiency by almost an order of magnitude compared to conventional processors. CloudSuite has been extensively used by renowned academic and industrial (both research and product) groups to drive their research and design, and has been integrated into Google's PerfKit Benchmarker.

DRAM caches for servers. While working on CloudSuite and Scale-Out Processor, I realized that specializing computation for high throughput ultimately drives designs into a memory bandwidth wall and exposes the overwhelming energy cost of moving data between the processor and memory. My dissertation research was mostly focused on reducing the communication between the processor and memory by bringing computation and data closer to each other and leveraging the 3D die-stacking technology that allows for co-integration of logic and DRAM in the same chip. I showed that the amount of DRAM we can integrate per chip is insufficient to host the entire memory that servers need, and that the on-chip DRAM should be instead used as a hardware-manage cache [7][8][9], and I demonstrated the thermal feasibility of such designs [10].

My thesis provides a detailed characterization of real-world server software stacks with respect to on-chip DRAM caches [11]. I showed that the single most important design parameter in DRAM caches is the granularity they operate on, based on which the caches can be divided into block-based and page-based (e.g., 64B vs. 4KB). The two design classes optimize for temporal and spatial locality respectively. My work showed that due to the scale of the server memory systems and scale of the DRAM caches, the temporal locality in DRAM caches is scarce, while the spatial locality is abundant. I proposed *Footprint Cache* [7], a practical die-stacked cache design to exploit the best aspects of both block- and page-based cache designs while avoiding their respective drawbacks. Footprint Cache achieves high hit ratios, efficient use of energy and bandwidth, and low latency for caches of up to a few gigabytes. To enable scalability to arbitrary stacked-DRAM capacities and remove the metadata lookup latency from the critical path, I proposed *Unison Cache* [8], which incorporates tags directly into the stacked DRAM and performs tag and data lookups in parallel and efficiently. Unison Cache is the first scalable DRAM cache solution that allows for practical and efficient implementation of prefetching, associativity, and all standard cache optimizations that rely on associativity [11]. So far my work in this area has received over hundred citations and has already been built upon by several research groups.

Virtual memory support for near-data processing. Alternative approaches to reducing the traffic between the processor and memory include moving data closer to the execution via NUMA-aware data placement, and moving the execution closer to the memory by leveraging DRAM and logic co-integration. As my recent work shows [12], both of these approaches exacerbate already severe problems associated with the conventional virtual-to-physical address translation, which include high complexity, dramatic performance overhead, and poor scalability. While bringing data and execution closer to each other reduces data fetch latency, leaving the page tables randomly placed across the memory system shifts the bottleneck to address translation [12].

My work leverages the observation that modern server workloads experience page swapping infrequently and have less fragmented memory layouts, often mapping many subsequent virtual pages to subsequent physical frames. While the conventional fully associative page placement flexibility remains largely unexercised, it mandates a costly level of indirection prior to each data fetch. This work is the first to show that restricting the mapping between the virtual address space and physical memory from fully associative to set associative has practically no impact on the page fault rate in modern servers. Limiting the associativity means that a page cannot reside anywhere in the memory system but only in a fixed number of statically determined physical locations; if all possible locations for a given virtual address are physically adjacent – i.e., in the same memory device and same DRAM row – the location of any data can be determined solely based on its virtual address, allowing for data fetch to proceed without waiting for translation.

This novel observation has led to the Distributed Inverted Page Table (DIPTA) translation mechanism, in which each memory partition keeps the translation information for its share next to the data. DIPTA selectively restricts associativity to speed up the translation for arbitrary parts of the address space. It keeps all but a few bits invariant across the virtual-to-physical mapping, and uses highly accurate way prediction (i.e., predicting the unknown bits upon an access) to enable address translation to complete together with data fetch. When applied to the entire address space, DIPTA obviates the need for conventional translation hardware – such as TLBs, MMU caches, and page walkers – and completely eliminates the ever-growing performance overhead of translation, providing speedups up to 5x and 2.4x over conventional translation using 4KB and 1GB pages respectively. DIPTA can be fully embedded into DRAM and is fully compatible with conventional page tables, which execution units still can still use to access the parts of memory with unrestricted associativity.

Specialized secure video storage. The need for denser and cheaper storage has never been stronger. Yet, storage specialization aiming to increase density by leveraging application properties remains largely unexplored. My recent study shows how disciplined approximation can be used to specialize storage to densely store already compressed videos [13], which are the most voluminous data type today.

While storage density can be effectively increased by shrinking the storage cells or by storing more information per cell, both approaches expose high error rates that quickly grow with the achieved gains. Using strong error correction schemes can alleviate the problem, but results in increasingly high storage overheads. Finding the sweet spot requires balancing the error correction overhead against the density gains. In this work, I leverage the fact that encoded videos are inherently noisy; encoding significantly reduces the storage footprint at the expense of deterministic noise. The key finding is that by exposing the noise stemming from the dense storage substrate in a disciplined way, it is possible to strike a better balance between the total noise, which includes both the encoder-produced deterministic noise and the storage-related non-deterministic noise, and the overall density benefits provided by both. The holistic approach to the quality/density optimization process allows for achieving better quality/density points, which video compression alone could not otherwise achieve.

To control the noise coming from the storage substrate, I proposed a novel and efficient methodology to compute bit-level reliability requirements for encoded videos and protect each bit accordingly. The key insight is that by tracking visual and metadata dependencies, we can compute the visual quality loss that would be suffered if any bit becomes corrupted and hence determine what reliability class each bit belongs to. A video file is then split into multiple bit streams of different reliability needs, with each stream being protected with an appropriate error correction scheme. I further leverage the static encoding order within a video frame to compactly describe the bit-to-reliability mapping with only a couple of bytes per frame, and store the reliability mappings within strongly protected frame headers. When applied to a dense but error-prone phase-change memory (PCM) substrate with eight information levels per cell, the proposed approach eliminates ~50% of the error correction overhead in the worst case – i.e., under the most error-intolerant encoder settings. Thanks to the streaming nature of videos, the methodology can be independently applied to short video sequences between key frames, allowing for an efficient implementation and use in real-time systems. Importantly, the reliance of the proposed storage system on approximation does not interfere with the ability to encrypt and successfully decrypt videos. Although encryption significantly changes the entropy of the data, I show that it is possible to apply the state-of-the art encryption schemes to the proposed approximate video storage in a corruption-tolerant way, such that approximating an encrypted video stream produces output of the exact same quality as encrypting an already approximated stream. The ability to encrypt videos according to the highest security standards is crucial in modern times, both for privacy and digital right management reasons.

Future Plans

Near-data TLBs and page tables (short term). Limiting virtual memory associativity to a very small number allows us to build DIPTA as an inverted page table in hardware, but it is not required for achieving most of the benefits. Instead, reducing the associativity *by* a small factor is sufficient, because most of the translation latency can be eliminated if the associativity is reduced by only as much as it is needed to determine the memory chip that holds the requested data. In this case it is up to the remote memory chip or its controller to locally infer the precise data location. Within a memory chip I plan to support full associativity, whereas the effective associativity is reduced only between the chips. Because finding data in a fully associative memory partition is difficult to efficiently perform in hardware, I plan to explore software-based solutions using traditional virtual memory and “near-data TLBs”, whose task is to speed up translations within the local memory partition. Unlike traditional control-flow centric TLBs, near-data TLBs hold mappings associated with a small memory shard and use their capacity more effectively. An alternative direction I plan to explore is restricting TLB miss latency through NUMA-aware placement of conventional direct page tables.

Approximate video services (medium term). Video streaming services, such as YouTube and Netflix, allow for variable-quality content streaming to accommodate the user's needs and capabilities. However, storing multiple versions of each video results in a significant storage overhead, which could be avoided by storing only the largest version and transcode it into smaller versions on the fly. A better approach would be to encode videos in one of the recently standardized scalable (layered) formats, in which every subsequent layer refines the previous by encoding extra frames per second, extra pixels, or extra quality (bitrate) for a fixed resolution. While the resulting video is slightly larger than the largest conventionally encoded video, the transcoding time is significantly reduced. While my work on specialized video storage so far considers a single encoding layer, the methodology I developed perfectly fits the scalable formats as well. The reason is that every subsequent layer only refines the visual quality of the previous and thus inherently requires lower reliability, allowing approximation across another orthogonal dimension.

Beyond storage, I plan to explore the benefits of my methodology when applied to video streaming, where different bits can be transferred through network channels of different reliability or priority, and approximate video processing, where less important video bits may need less precise computing. This approach is a perfect fit for resource-demanding video applications such as virtual reality, in which the content outside of current focus has significantly lower reliability requirements.

Holistic approach to building future multimedia systems (long term). Storage, processing, and delivery of multimedia content will certainly remain indispensible in the 21st century and will only grow in importance. What these applications will look like in the future mostly depends on how the resources they use will relate to each other in terms of performance and cost. For example, assuming that storage is much more expensive than compute, the most economical way for Netflix to store videos would be to store the largest video only and transcode it into lower quality versions on demand. Under the opposite assumption, the most economical way would be to store each version of the video and avoid transcoding entirely, with countless intermediate solutions between the two extremes. Understanding these trade-offs and extrapolating the technology trends is key to optimal and timely designs. Unfortunately, while different system components have so far been improving predictably and at the rates that were necessary to keep the entire system in balance [14], the end of the conventional technology scaling, the vast differences in specialization opportunity for different system components, together with the emergence of new and radically different technologies (e.g., DNA-based storage) will likely change the half a century old balance of resources in unpredictable ways. As a result, we may soon be designing systems with quite unusual parameters and very differently proportioned resources, and I'm interested in understanding the capabilities of such machines.

My long-term goal is to design an optimization framework that would guide the system-application co-design for given technology parameters, or help explore the potential roles and benefits of new technologies in the context of multimedia systems. The important advantage of multimedia applications is that they have been well studied and feature a whole spectrum of alternative solutions, starting from the highest-level algorithm to the lowest-level entropy coder, which are designed under different resource constraints and show vastly different properties and trade-offs regarding the use of system resources. The already rich spectrum of solutions will rapidly expand with the emergence of ML-assisted multimedia compression/processing approaches and applications such as virtual/augmented reality, which bring entirely new trade-offs. On the systems side, there is variety of techniques that are often used to implicitly trade one resource for another. Examples include many forms of caching, which can, for instance, trade memory/storage for bandwidth/latency; memoization, which trades storage for compute; data distribution and replication, buffering, etc. The framework is supposed to include all known techniques that affect the balance of resources both at the application and at the system level. The system modeling must also include the user's capabilities and user behavior, such as load and popularity distributions, for which I will seek to establish strong partnerships with the relevant industry partners. Hopefully, the envisioned framework will be eventually generalized to other types of flexible applications, most notably machine learning. I believe that the design process for future systems will increasingly have to rely on such holistic frameworks that can leverage the flexibility of the application and the system and adapt to the rapidly evolving technology space.

References

[1] CloudSuite: A Benchmark Suite for Cloud Services. http://cloudsuite.ch

[2] M. Ferdman, A. Adileh, O. Kocberber, S. Volos, M. Alisafaee, **D. Jevdjic**, C. Kaynak, A. Popescu, A. Ailamaki, and B. Falsafi. *Clearing the Clouds: A Study of Emerging Scale-Out Workloads on Modern Hardware*. In International Conference on Architectural Support for Operating Systems and Programming Languages (ASPLOS), Mar 2012.

[3] P. Tozun, I. Pandis, C. Kaynak, **D. Jevdjic**, and A. Ailamaki. *From A to E: Analyzing TPC's OLTP Benchmarks - the Obsolete, the Ubiquitous, the Unexplored*. In International Conference on Extending Database Technology (EDBT), Mar 2013.

[4] M. Ferdman, A. Adileh, O. Kocberber, S. Volos, M. Alisafaee, **D. Jevdjic**, C. Kaynak, A. Popescu, A. Ailamaki, and B. Falsafi. *Clearing the Clouds: A Study of Emerging Scale-Out Workloads on Modern Hardware*. In ACM Transactions on Computer Systems (TOCS), Vol. 30, Issue 4, November 2012

[5] M. Ferdman, A. Adileh, O. Kocberber, S. Volos, M. Alisafaee, **D. Jevdjic**, C. Kaynak, A. Popescu, A. Ailamaki, and B. Falsafi. *A Case for Specialized Processors for Scale-Out Workloads.* In IEEE Micro Top Picks, Vol. 34, 2014.

[6] P. Lotfi-Kamran, B. Grot, M. Ferdman, S. Volos, O. Kocberber, J. Picorel, A. Adileh, **D. Jevdjic**, S. Idgunji, E. Ozer and B. Falsafi. *Scale-Out Processors*. In International Symposium on Computer Architecture (ISCA), Jun 2012.

[7] **D. Jevdjic**, S. Volos, and B. Falsafi. *Die-Stacked DRAM Caches for Servers: Hit Ratio, Latency, or Bandwidth? Have It All with Footprint Cache.* In International Symposium on Computer Architecture (ISCA), Jun 2013.

[8] **D. Jevdjic**, C. Kaynak, G. Loh, and B. Falsafi. *Unison Cache: A Scalable and Effective DRAM Cache*. In International Symposium on Microarchitecture (MICRO), Dec 2014.

[9] S. Volos, **D. Jevdjic**, B. Falsafi, and B. Grot. *Memory Systems for Scale-Out Servers*. In IEEE Micro, 2016.

[10] D. Milojevic, S. Idgunji, **D. Jevdjic**, E. Ozer, P. Lotfi-Kamran, A. Panteli, A. Prodromou, C. Nicopoulos, D. Hardy, B. Falsafi, and Y. Sazeides. *Thermal Characterization of Cloud Workloads on a Power-Efficient Server-on-Chip*. In International Conference on Computer Design (ICCD), Sep 2012.

[11] **D. Jevdjic**. Multi-Gigabyte On-Chip DRAM Caches for Servers. PhD thesis. September 2015.

[12] J. Picorel, **D. Jevdjic**, and B. Falsafi. *Near-Memory Address Translation*. Submitted for publication. https://arxiv.org/abs/1612.00445

[13] **D. Jevdjic**, K. Strauss, L. Ceze, and H. Malvar. *Approximate Storage for Encoded and Encrypted Videos*. International Conference on Architectural Support for Operating Systems and Programming Languages (ASPLOS), Apr 2017 (to appear).

[14] J. Gray, and P. Shenay. *Rules of Thumb in Data Engineering*. In International Conference on Data Engineering, March 2000.