Specialized Microservers for the Data Center

Zsolt István (presenter)
Systems Group
Dept. of Computer Science, ETH Zürich
zsolt.istvan@inf.ethz.ch

David Sidler
Systems Group
Dept. of Computer Science, ETH Zürich
dasidler@inf.ethz.ch

Gustavo Alonso
Systems Group
Dept. of Computer Science, ETH Zürich
alonso@inf.ethz.ch

1. The Case for Microservers

Our work is motivated by two trends: the continuous growth of data centers that prompts more energy-conscious thinking, and the prevalence of scale-out workloads in the modern data center. Key-value stores are a good example of such workloads: services such as memcached are often used to ease the demand on the hard to scale relational databases, but since these key-value stores are more network heavy than computationally intensive, they do not utilize modern server hardware efficiently.

The root of this inefficiency comes from the fact that while regular servers have high computational capacity, they consume power on various levels of caches, multiple cores, and external devices. When the workload is mostly I/O bound, the large cores and additional devices will waste power without providing more performance [1]. A solution proposed both by academia and industry is to tailor servers to specific applications with the goal of higher efficiency. In the case of key-value stores, for instance, this has been achieved by using an ARM-based CPU with less cores and smaller caches. Unfortunately, the simpler CPUs (both low-frequency x86 and ARM architectures) result in lower per-machine throughput, even if the servers are overall more efficiently utilized.

Our goal is to dramatically reduce power consumption without compromising performance, so we took specialization a step further: we present a **microserver that implements the target application directly in hardware**, fusing network handling and application-specific processing on a Xilinx FPGA. We based our implementation on our previous fully pipelined, high performance, hash table design [2]. The specialized microserver not only **1) consumes less power than regular servers** (25W vs. 200W), it also **2) surpasses them in performance** (we are able to saturate 10Gbps line-rate for most request sizes and workload mixes), and **3) delivers very low latency** (a 5X improvement over software, with very low variation). We made our server compliant with the memcached protocol so that it can be evaluated as a drop-in replacement for the software based version, which has been extensively benchmarked and optimized in the community.

2. Building a Cluster

In key-value store deployments replication is commonly used for high availability and data consistency. Our goal is to provide a layer of replication on top of our FPGA-based servers, something that to our knowledge has not been explored yet in the literature. We explore the benefits of **carrying out replication over a secondary, "backbone" network**. We build and evaluate a ring interconnect, but the possibilities are much wider, and our implementation is flexible enough to accommodate other setups as well.

The FPGA boards that we use to build the microservers have four 10Gbps network interfaces each (one of which is configured to act as the main interface to clients); the remaining three interfaces can be used to build secondary networks. Our work is inspired by the recent Catapult project [3] at Microsoft Research, where a set of FPGAs work together, and communicate on a secondary network, in order to carry out a larger task.

Our "backbone" network is built on top of raw Ethernet frames on 10Gbps bidirectional links. The main benefit of an all-hardware solution is that the latency of accessing the network is minimal. We measured communication between two application pipelines on two different microservers to be below 2μs. This means that replication can be performed without incurring any penalties from the client's perspective, and as we will show in the poster, that it is **possible to serve close to full 10Gbps load on each instance while replicating requests to its neighbor in the ring**.

To sum up, our main goal in this work is to show the feasibility of building large distributed systems with specialized microservers, and to evaluate the benefits in performance and efficiency. Additionally, the lessons learned from this work might be useful for future generations of microservers, that are a combination of specialized hardware and general purpose processing elements.

3. Future Directions

Our future research on this topic mainly focuses on comparing different replication schemes and "backbone" network options. We are currently evaluating a replication scheme in which one microserver is used exclusively to replicate write requests to a set of other servers, and where clients read directly from a subset of said servers.

In addition, we are exploring the idea of consensus protocols running on our microservers. The motivation behind this idea is that the high determinism of the hardware pipelines should allow for simplified consensus logic in the common case. As a result, the overhead of achieving strong consistency should also be much lower than in regular, software-based systems.

References

[1] 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. *Micro, IEEE*, 34(3):31–42, 2014.

[2] Zsolt István, Gustavo Alonso, Michaela Blott, and Kees Vissers. A flexible hash table design for 10gbps key-value stores on fpgas. In *Field Programmable Logic and Applications (FPL), 2013 23rd International Conference on*, pages 1–8. IEEE, 2013.

[3] A. Putnam, A. Caulfield, E. Chung, D. Chiou, K. Constantinides, J. Demme, H.i Esmaeilzadeh, J. Fowers, G. Gopal, J. Gray, M. Haselman, S. Hauck, S. Heil, A. Hormati, J. Kim, S. Lanka, J. Larus, E. Peterson, S. Pope, A. Smith, J. Thong, P. Xiao, and D. Burger. A reconfigurable fabric for accelerating large-scale datacenter services. In *41st Annual International Symposium on Computer Architecture (ISCA)*, June 2014.

Systems@ETH zürich

zsolt.istvan@inf.ethz.ch

Zsolt István, David Sidler, Gustavo Alonso

Department of Computer Science, ETH Zürich

ETH zürich

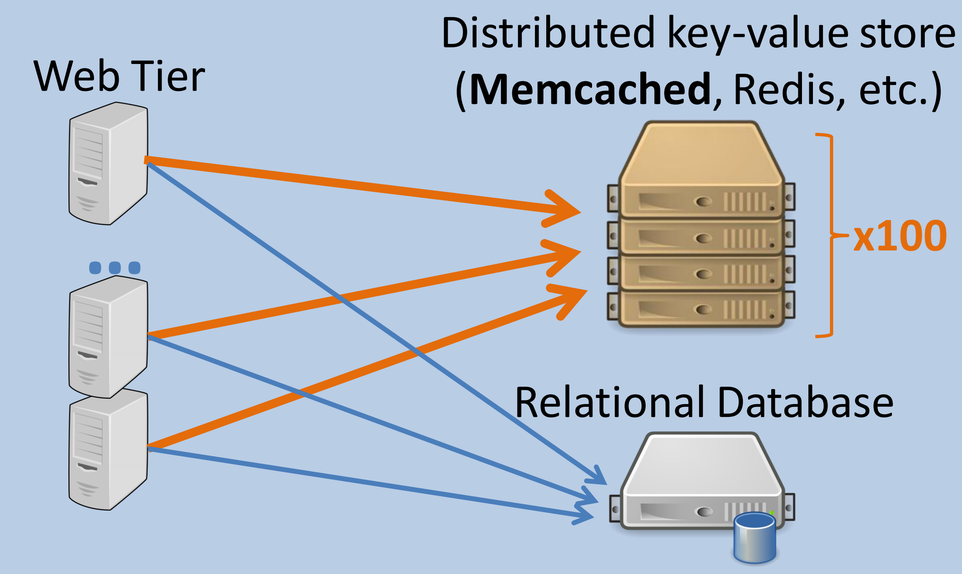
Specialized μservers for the data center

MICROSERVER *n.* An application-specific server implemented as a small appliance

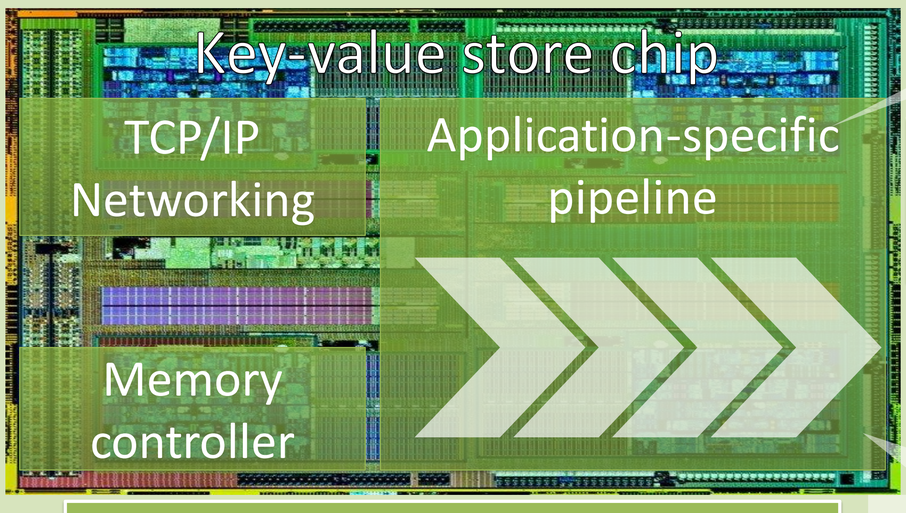
1 some widely-deployed applications are not a good fit for current servers

Main-memory key-value stores deployed in most datacenters
- Network-heavy: little computation, high concurrency
- Random access: multi-level cache poorly utilized

Modern CPUs and architectures are optimized for the opposite!



2 specialize: design and build a system for the application



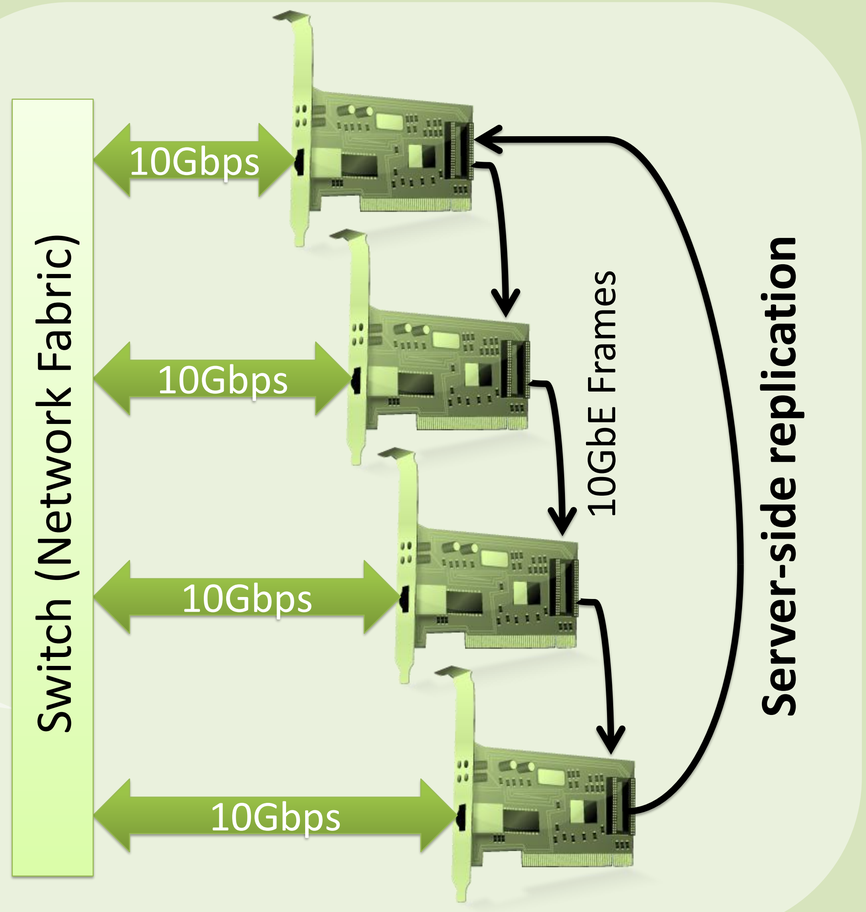
Main Memory

Memcached-equivalent server with state-of-the-art components

Distribute & Replicate
Performance & Fault-tolerance

Clients see regular application interface

We use FPGAs for prototyping:
4x 10Gbps interfaces,
8GB memory

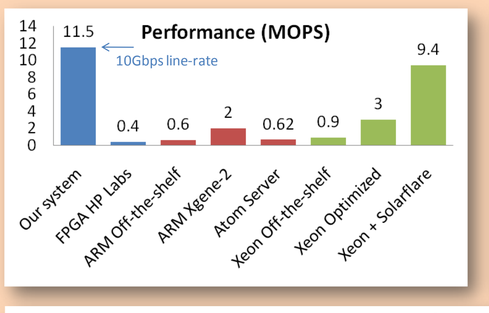


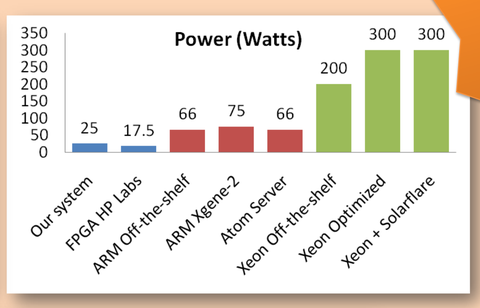
3 evaluate how much better the specialized server is

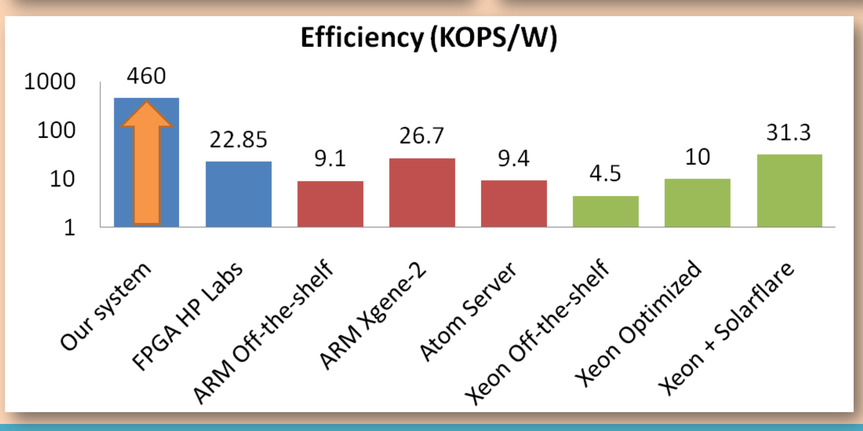
Highest efficiency, without sacrificing performance

Predictable response time, even with replication

Single node



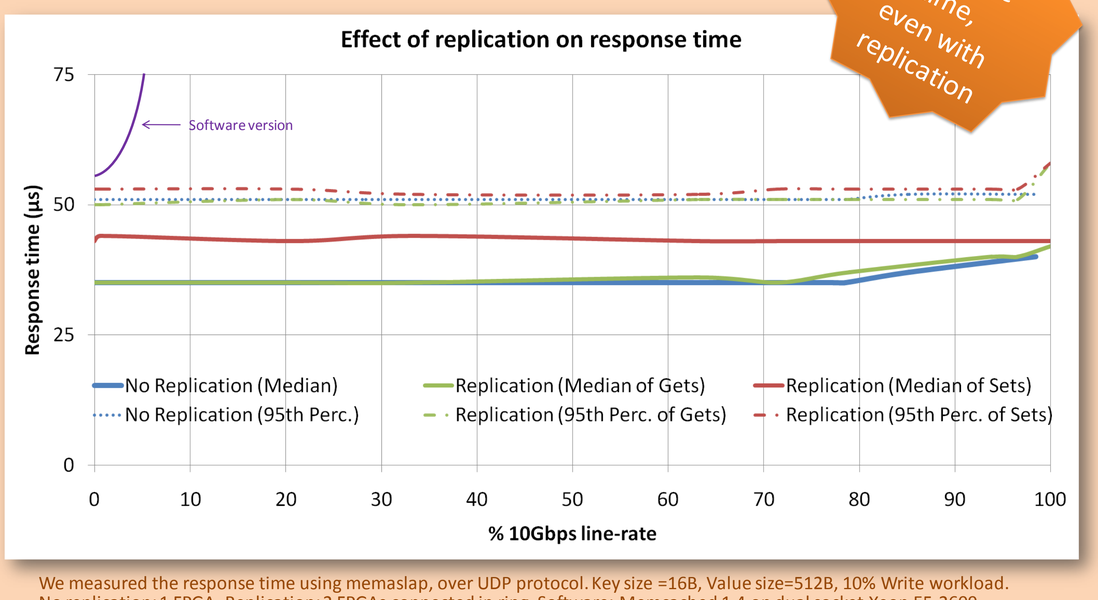




FPGA 10Gbps line-rate maximum is for UDP and 16B key and 16B value, a common ground of other related work.

Related work & systems:
- FPGA HP Labs – Appliance optimized for 1Gbps networking (FPGA'13)
- ARM Off-the-shelf – 8x ARMV8 @2.4GHz
- ARM Xgene-2 – From promo material
- Atom server – 8x Atom C2750 @2GHz (Intel promo)
- Xeon Off-the-shelf – 8x Xeon E5 @2.4GHz
- Xeon Optimized – 16x Xeon E5 @2.2GHz (Intel whitepaper on optimizing memcached)
- Xeon + Solarflare – 20x Xeon E5 @2.2GHz (Solarflare whitepaper on optimizing memcached for its NICs)

Replication



We measured the response time using memaslap, over UDP protocol. Key size =16B, Value size=512B, 10% Write workload. No replication: 1 FPGA. Replication: 2 FPGAs connected in ring. Software: Memcached 1.4 on dual socket Xeon E5-2609.

```
Response time = Client + Network + Internal
  36-42µs     =      33-35µs     +  2-6µs
```

4 speed up the datacenter

Using our current key-value store as a building block in larger systems:
- ZooKeeper-like metadata store
- Speeding up consensus protocols

Building more complex micro-servers with FPGAs:
- Persistent storage: FPGA + 64GBs of memory + SSDs
- Flexibility in management: FPGA + Atom/ARM SoCs

(with large reprogrammable fabric and small CPU)

Acknowledgements: This work is funded in part through the Microsoft Joint Research Center MSR-ETHZ-EPFL and an equipment donation from Xilinx. It builds upon an earlier collaboration with Xilinx.