EE Times

Addressing data growth and the impact on data center sprawl and cost in terms of performance/power per sq. ft.

In the June 15, 2017 article by Paul Master, co-founder and CTO at **Cornami**, discussed the high-performance computing needs of artificial intelligence (AI) markets, such as machine learning (ML), robotics, autonomous driving, analytics, and financial markets, which are all reliant on Big Data (BD) that requires real-time computing on a massive scale. To meet these needs, Cornami has developed and patented a new computing architecture, the TruStream Compute Fabric (TSCF), which is comprised of a sea of small cores (silicon); employs a unique, easy-to-use programming model that supports all forms of parallelism; and fabric scalability for processing massive data workloads concurrently. The TSCF significantly increases processing efficiency while reducing power usage, latency, and platform footprint, resulting in dramatically increased performance by orders of magnitude.

Another factor worthy of serious consideration by all corporations, large and small, is the ever-increasing data sprawl that is a direct result of these compute-intensive applications, the number of employees/users in multiple locations, and their myriad associated electronic-communication devices, such as desktops, laptops, tablets, and smartphones. Within a single corporation, the data can double within a year, which year after year becomes a staggering amount of data to process, store, manage, and protect. Additionally, for each of these functions there are associated costs that will continue to increase over time, for example: More high-speed computers for processing data; data storage facilities with racks of energy-hungry servers; complex networks and management software; and fail-safe security.

The business case for efficiently and effectively managing data sprawl is really matter of smart computing practices:

- Reduce the need for additional data centers which are costly to build and require long-lead times.
- Massive increases in data computing and storage becomes costlier with the demand for more power usage and cooling, and expanded footprints for data centers.
- The carbon footprint for these processes multiplies rapidly, increasing power grid requirements, which exacerbates nature's global warming.
- The ability to quickly access well-managed data allows companies to swiftly adapt and respond to new business requirements.

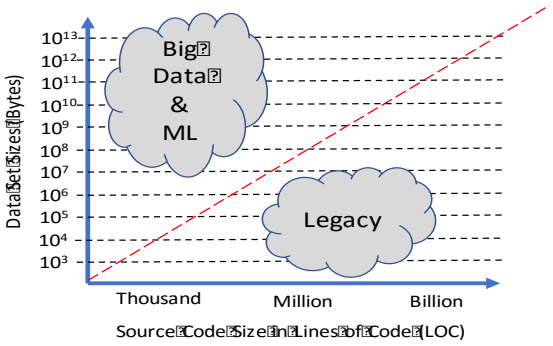
Any new computer architecture must be evaluated by an analysis of its effects on the datacenter.

Cornami's TruStream Compute Fabric (TSCF) addresses the issues associated with managing data sprawl in a highly efficient manner. TSCF is a massively parallel architecture comprised of a scalable sea-of-cores with

independent decision-making capabilities at each processing core and interspersed with high-speed memory. The TSCF is extensible across multiple chips, boards, and racks, with each core being independently programmable.

As we discussed in the previous article in this series, traditional architectures were optimized for characteristics that no longer apply to those of modern Big Data and Machine Learning workloads. This has major implications for a large, modern data center, specifically power consumption.

Figure 1 - Data Sets vs. Source Code Size



To recap the earlier article, as shown in figure 1, traditional legacy workloads can be described as having large source-code bases (million plus lines of code) that process relatively small data sets. Contrast that with Big Data and Machine Learning applications in which the source-code bases are measured in thousands of lines of code, but process data-sets that can be petabytes in size. Scaling for these applications is currently performed by replicating these small code-base apps on endless racks of traditional servers in a datacenter. Typical processor architectures have been optimized over decades to process these traditional legacy workloads – large source-code bases with smallish data sets (think large office applications, large operating systems, etc.). The key architectural super-specialization that has occurred to optimize the performance of these legacy applications is the multi-level cache. Processor architects discovered that large software code exhibits what is termed locality. Locality is a term (https://en.wikipedia.org/wiki/Locality_of_reference) that describes the "phenomenon in which the same values, or related storage locations, are frequently accessed." Or in other words, source code clusters together. There is a high probability that the next line of code your application will execute is located near the last line of code that was executed.

By placing a fast, local, cache memory that keeps a copy of this code cluster near the processor core, the processor core does not need to go to off-chip memory as frequently. Since access to any off-chip memory takes orders of magnitude more time and power than accessing on-chip memory, the system runs fasters. This is a widely successful multi-level caching model for source-code counts of large applications, and when coupled with underlying libraries and state of the art operating systems, can be measured in hundreds of millions of lines of source code.

But what happens if the underlying assumption is no longer correct, i.e. the assumption the processor and cache memory are efficiently running source code that is measured in hundreds of millions of lines of code is no longer valid? What does this multi-level caching subsystem cost in terms of power consumption when running todays Big Data and Machine Learning applications? What does this multi-level caching subsystem cost in terms of silicon area? Let's quantify this cost.

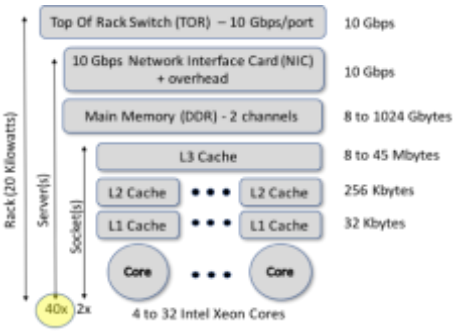


Figure 2 - The building block of a data center. A rack built up of a Top of Rack Switch, servers, and the servers built up of processor cores. The cache subsystems and their sizes are highlighted. Note that a high power 20 Kilowatt rack can have 40 1U servers in addition to the Top of Rack Switch.

Figure 2 shows the building block of a modern data center that consists of commodity servers (1U in height) that typically have two processor slots, with multi-core Intel Xeon processors installed. Multiple servers (up to 40 1U servers) are placed in racks with a top-of-rack Ethernet switch, which interconnects all the servers in the rack with a larger switch that interconnects racks together. Note the cache subsystem – main memory in the form of DDR resides on the motherboard of the server, which feeds an L3 cache, which feeds individual L2 caches, which feed L1 caches, which finally feed each individual processor core. (Note that bulk storage, spinning platters, SSDs, exotics, etc., are not shown here since we are highlighting the processing of streaming data for big data/machine learning.) A high power 20 kilowatt rack in a datacenter can have up to 40 1U of these servers installed.

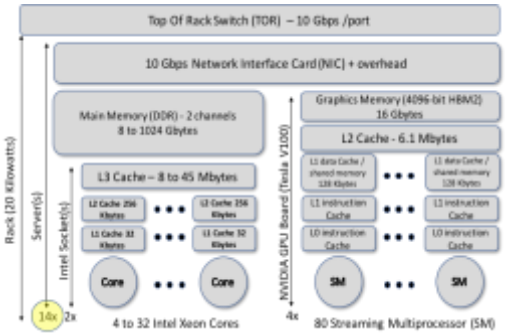


Figure 3 - The GPU server version. This high power 20 Kilowatt RACK can only have 14 1U servers installed since the GPU cards consume the total power budget of the rack. The GPU's caches and sizes are highlighted. See [8] & [9] for more architecture details on the NVIDIA Tesla V100.

In contrast to Figure 2, Figure 3 shows the same rack of 1U servers, but this time filled with the latest NVIDIA Tesla V100 cards. Up to 4 GPU cards can be installed in each 1U server, with each GPU having 80 streaming multiprocessors (SM). Note the familiar cache sub-systems, where each SM has a L0 instruction cache, a L1 instruction cache, a L1 Data Cache that can be configured as a shared memory, and a L2 cache, backed by off-die graphics memory chips (HBM2). Each NVIDIA Tesla V100 card can consume a maximum 250 Watts each.

So, what's the cost of this legacy architecture? What does it cost in energy to do work across the different levels of caches and the top-of-rack switch? The key metric we will be using to quantify the cost is the joule, https://en.wikipedia.org/wiki/Joule; think of this as heat produced over a specific time, in which one joule is equivalent to one watt-second. Pulling the relative energy data from references [1], [2], [3], [4] & [5] allows us to generate figure 3. This shows the relative energy requirements of the different architectural elements that make up a rack of servers.

Here are some notes we considered before analyzing the data.

1) The graph's y-axis is a log scale (each step size is 16 times more power) – this means as you perform operations higher on the graph, the energy cost is exponentially larger.
2) As a typical operation, we show the relative energy to perform a 32-bit floating point multiplication; since we are looking at big data and machine learning problems, 32-bit FP multiplication is an intermediate energy operation placeholder.
3) Some but not all ML algorithms require 32-bit FP, with 16-bit FP and Integer 8 also being used, which require lower energy.
4) Some but not all big data applications require 32-bit FP, with a 32-bit integer also used, which requires lower energy.

5) Note that what is not shown is the energy required for the processor to decode instruction, branch prediction, etc. (see [3] for more details).
6) Also note the energy values are for a modest Intel XEON chip with only 8 Mbytes of L3 cache. Be aware that energy consumption grows with the size of the cache memory and depending on your budget you can get L3 cache sizes of up to 45 Mbytes, so the L3 cache bubble will rise higher on the graph, but it will stay below the DRAM energy.
7) The DRAM uses 2 channels, which are using 2 DIMM modules (most Intel parts support at least 2 channels of DRAM). Energy for the 10 Gbps Network Interface Card (NIC) also includes the energy for the processor to process, package, and send the TCP/IP packets. Not surprisingly, the TCP/IP software processing is the largest component; see [1] for more details.
8) The top-of-rack switch power is shown for one 10 Gbps link. For this analysis, we are making the assumption that the traffic between servers is staying inside the rack.

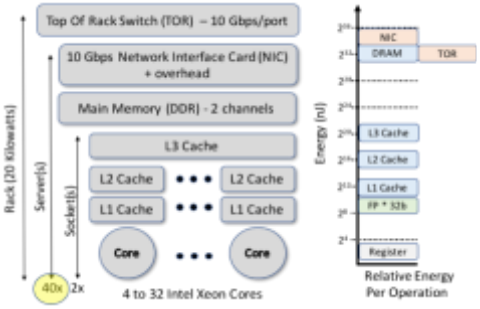


Figure 4 - Relative Energy per Operation – Note the chart on the right is logarithmic scale. Also note that a high power 20 Kilowatt Rack can support 40 1U servers in addition to the Top of Rack Switch.

Here are some observations about this coarse data that we are using this to calculate relative energy costs.

1) The cheapest possible operation in terms of energy, not surprisingly, is moving data on the die (register, ~64x less energy than a FP multiply).
2) Accessing the L1 Cache costs roughly ~5x times the energy required to perform a 32-bit Floating Point operation.
3) Getting data from the L2 cache costs ~16x more energy than a L1 access.
4) Hitting the L3 is another ~16x more energy than the L2.
5) The DRAM is a ~4000 time more energy than the L3 cache.
6) Moving data to the NIC card costs ~8x the cost of DRAM access.
7) Finally, sending data through the top-of-rack switch costs roughly as much as a DRAM access.

What does this tell us about existing architectures?

- Workloads must remain in the L1 cache for lowest-power usage and highest performance. Anything that can overwrite the cache will result in a penalty. What can overwrite the cache? Running any code of any reasonable size, which includes interrupts, operating system kernel, device drivers, background daemons, and networking protocols, risk flushing the code that is performing work within the caches.

  For example, Linux 4.12.4 is approaching 20 Million lines of code (see [7] for the size of the Linux code bases). Trying to manually tune what can execute on a traditional core is incredibly difficult. Existing architectures are designed to execute large code-base applications and not small code-base applications.
- Moving data into the server, through the DRAM, through all the cache levels is an incredibly high-power activity. And since the hallmark of big data and machine earning applications is the streaming of data in petabytes through these servers, we can draw the conclusion that existing architectures are not optimized for these types of workloads.

This highlights several desirable characteristics of a new architecture, such as the TSCF, that is specifically targeting big data applications, such as AI and machine learning:

(1) Moving data directly is cheap. Using on-die networks to move data from core to core is much more efficient and exponentially lower cost than using a multi-level cache.
(2) Computation is cheap. On workloads that can consume vast amounts of computation, having more computation on die is better than less. You can do 5 floating point multiplications for the energy of a single L1 cache access. ~80 for the energy used by a single L2 cache access. ~1280 for a single L3 access. ~5,242,880 for a single DRAM access. ~31 million for a single NIC access.
(3) There is a significant power advantage to an architecture that does not have the huge L1/L2/L3 caches. The caches consume exponentially more power as data traverses through them. Small memories that are tuned to the size of the workloads exhibit a power advantage, even against identically sized cache as the circuitry needed for cache functionality is not required.
(4) Transistors, which are currently used for caches, would be better used to add more computational cores.
(5) Dense computational fabrics, such as Cornami's TruStream Compute Fabric (TSCF), exhibit a power advantage. Adding many, many cores inside a single die reduces the number of transactions that need to go off server through the top-of-rack switch to another server, and then through all the cache layers. A dense computational fabric of small cores per die increases datacenter efficiency and lowers energy costs.

Using this "lens of energy" dissipation to identify the characteristics of a new architecture that is designed to run Big Data and Machine Learning applications not only identifies energy optimizations, but how an energy-optimized design has the advantages of higher performance and lower costs.

The conclusion is clear: The key to efficiently managing massive data growth and its subsequent data sprawl is using a new, massively parallel-processing architecture that is comprised of a scalable sea-of-cores with a dense computational fabric.

References:

[1] "Characterizing 10 Gbps Network Interface Energy Consumption" by Sohan, Rice, Moore & Mansley

[2] "Beyond the Roofline: Cache-aware Power and Energy-Efficiency Modeling for Multi-cores" by Ilic, Pratas, & Sousa in IEEE Transactions On Computers, January 2017

[3] https://www.futurearchs.org/sites/default/files/horowitz-ComputingEnergyISSCC.pdf

[4] http://www.mellanox.com/related-docs/prod_eth_switches/PB_SN2410.pdf

[5] "Multicore Technology: Architecture, Reconfiguration, and Modeling"
edited by Muhammad Yasir Qadri, Stephen J. Sangwine

[6] http://www.embedded.com/electronics-blogs/say-what-/4458517/A-new-computing-architecture-for-AI-applications

[7] https://www.linuxcounter.net/statistics/kernel

[8] https://devblogs.nvidia.com/parallelforall/inside-volta/

[9] https://images.nvidia.com/content/volta-architecture/pdf/Volta-Architecture-Whitepaper-v1.0.pdf