Holistic generational offsets: Fostering a primitive online abstraction for human vs. machine cognition

Shaun D'Souza* Trevor Mudge
Computer Science and Engineering
University of Michigan

Abstract

We propose a unified architecture for next generation cognitive, low cost, mobile internet. The end user platform is able to scale as per the application and network requirements. It takes computing out of the data center and into end user platform. Internet enables open standards, accessible computing and applications programmability on a commodity platform. The architecture is a super-set to present day infrastructure web computing. The Java virtual machine (JVM) derives from the stack architecture. Applications can be developed and deployed on a multitude of host platforms. O(1) $\leftrightarrow$ O(N). Computing and the internet today are more accessible and available to the larger community. Machine learning has made extensive advances with the availability of modern computing. It is used widely in NLP, Computer Vision, Deep learning and AI. A prototype device for mobile could contain N compute and N MB of memory.

Keywords— Mobile, AI, Cognitive, Server, Internet

1 Introduction

The web ecosystem is rapidly evolving with changing business and functional models. Cloud platforms are available in a SaaS, PaaS and IaaS model designed around commoditized Linux based servers. 10 billion users will be online and accessing the web and its various content. The mobile and internet are ubiquitous today. The industry has seen a convergence around IP based technology. Additionally, Linux based designs allow for a system wide profiling of application characteristics.

*Former affiliation

A virtualized architecture consists of Figure 1. The compiler is the glue logic for all the layers interfacing software to the underlying platform. Application performance was a primary determinant of system performance. Processor and Memory technology determine system performance [10]. With the advent of the Internet, computing performance is increasingly being utilized in the network. Applications are internet based and network connectivity is central to the platform. Network performance is a primary determinant of system. Existing internet connectivity are limited by technology capabilities Wi-Fi, 4G (Mbps).

Figure 1: Host architecture.

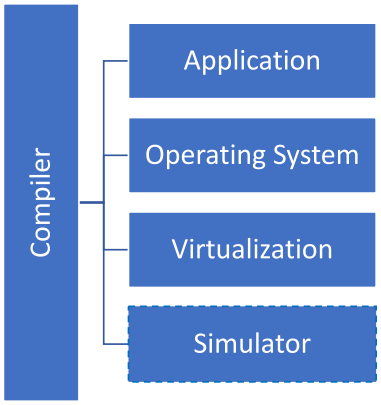
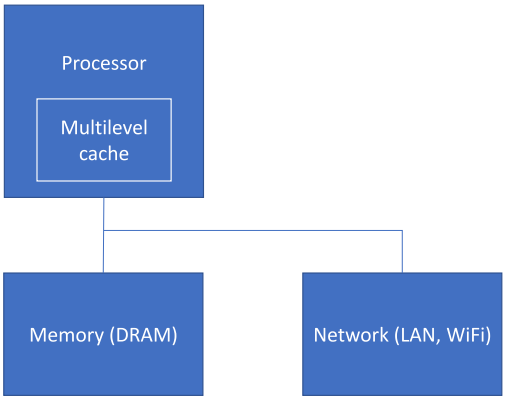


Figure 2: Conventional computer platform components. Compute coupled to memory and network.



Conventional Computer platform components

- Processor, Cache
- Memory (DRAM)
- Network (LAN, Wi-Fi)

Increasingly applications are internet based. Network performance is a primary determinant of system performance. Conventional computer was an in the box solution for desktop applications. The architecture was developed for high performance desktop applications. Processor, cache (GHz) coupled to high capacity Memory (DRAM). Figure 2 shows a conventional platform with Processor and cache coupled to Memory and Network.

Processor technology speeds are increasing faster than DRAM memory technology. Processors are designed to operate at a high frequency >2 Ghz. Caches are coupled to the processor to facilitate execution at high speed. DRAM memory technology is designed for high density >2 GB. As a result, the platform is not able to scale to meet the network performance and system performance requirements.

Network applications rely of moving data from network to memory and the processor - Figure 2. As a result, system performance is determined by bandwidth throughput in the memory. Internet applications are consuming increasing bandwidth and will use $>1-10$ Gbps bandwidth in a mobile platform.

Processor and memory (DRAM) technology have evolved independently to increase system performance. Processors were designed to run at high speeds (>2 GHz). Memory (DRAM) was designed for large capacity (>2 GB)

Until recently it was not feasible to integrate multi-Mb memory in a processor. Caches were used in the processor with application residing in main memory DRAM. However, with technological advances it is now possible to integrate multi-Mb memory (SRAM) in a processor. This enables us to re-evaluate the system hierarchy with processor, memory and network - Figure 9.

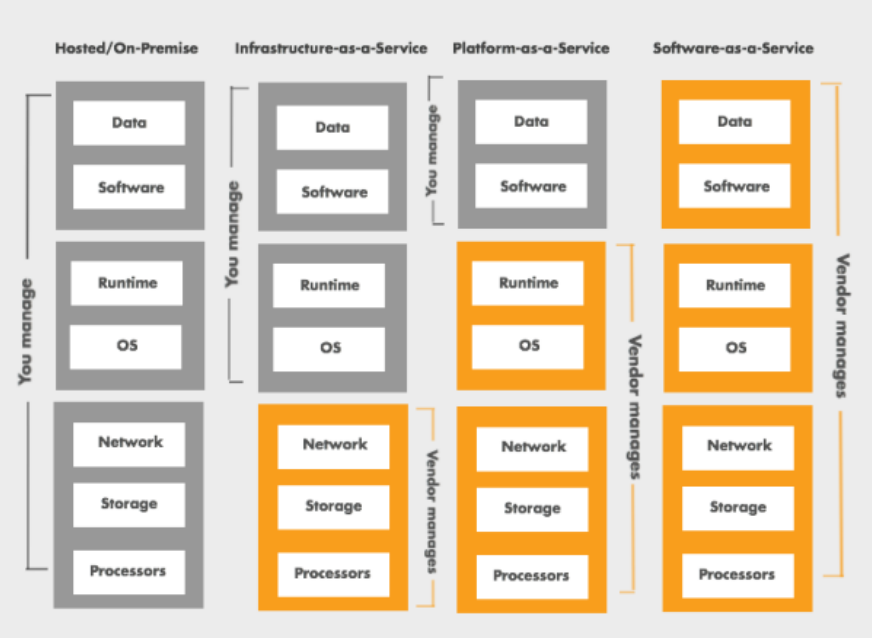
Integrating large memory in the processor allows us to eliminate caches in the design. Desktop applications were constrained by technology performance [13]. However recently we are seeing a saturation in application performance requirements. The Internet is today the platform for application enabling and the internet operational enablement is a driver of technology growth [4].

Web N. Open source has enabled the development of more efficient internet systems. Cloud computing combined with service-oriented architecture (SOA) and parallel computing have influenced application software development. Code is implemented in a variety of frameworks to ensure performance including OpenMP, MPI, MapReduce and Hadoop. Parallel programs present a variety of challenges including race conditions, synchronization and load balancing overhead.

Virtualization. This hosted infrastructure is complemented by a set of virtualization utilities that enable rapid provisioning and deployment of the web infrastructure in a distributed environment. Virtualization abstracts the underlying platform from the OS enabling a flexible infrastructure.

Open Source. Additionally, the cloud ecosystem is supported by the Open Source community enabling an accelerated scale of development and collaboration [14]. Simulators are used to benchmark application systems.

Figure 3: As-a-service cloud.



2 Differentiator

We propose an architecture for the next generation enterprise including an end to end solution for the web infrastructure. This highlights the challenges in bringing billions of users online on a commodity platform. There is a large opportunity in enabling technology consumption for more than a billion users.

- Web N, 10 Billion users, Intelligent machines, Turing test
- Social media, enterprise mobility, data analytics and cloud
- AI, Technology and enterprise, Virtualization, Open Source
- Machine learning, compilers, algorithms, systems

OOP and Java have enabled enterprise system architecture. Java is an algorithms, web and enterprise centric programming language. It allows for deployment of applications on a host of platforms running a virtual machine. Write once, run anywhere (WORA). 3 billion mobile devices run Java. Enterprise applications provide the business logic for an enterprise. Architectures have evolved from monolithic systems, to distributed tiered systems, to Internet connected cloud systems today.

Figure 4 shows the Virtual memory system on a host architecture and address translation to a physical address using the Page table in the Operating system.

Figure 4: Virtual memory.

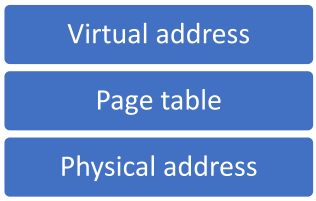
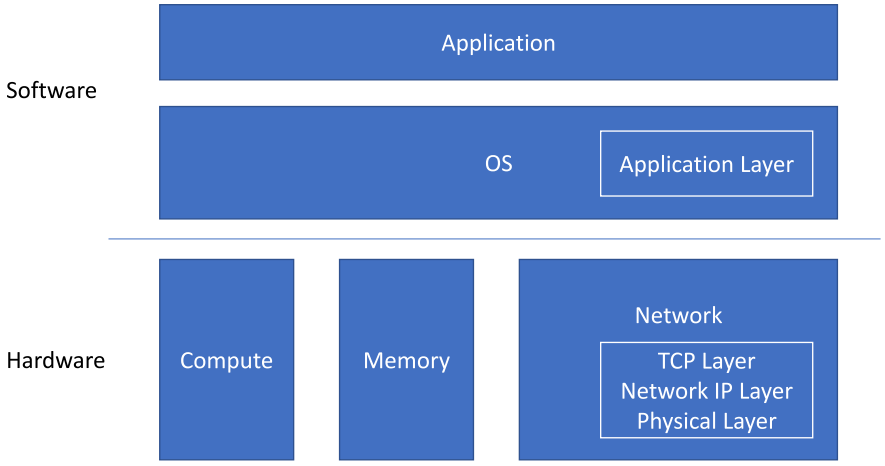


Figure 5: Conventional computer platform stack.



Technology assumptions -

- Architect and design cloud applications to support multi-tenancy, concurrency management, parallel processing and service-oriented architecture supporting rest services.
- Law of accelerating returns [12] and Moore's Law
- Prices and margins, competition, converging global supply and demand, evolving business models

A conventional computer platform consists of - Figure 5
Hardware:

- Processor and cache
- Memory (DRAM)
- Network TCP layer, IP layer, Physical layer – LAN, Wi-Fi, WiMAX, 5G, 4G

Software:

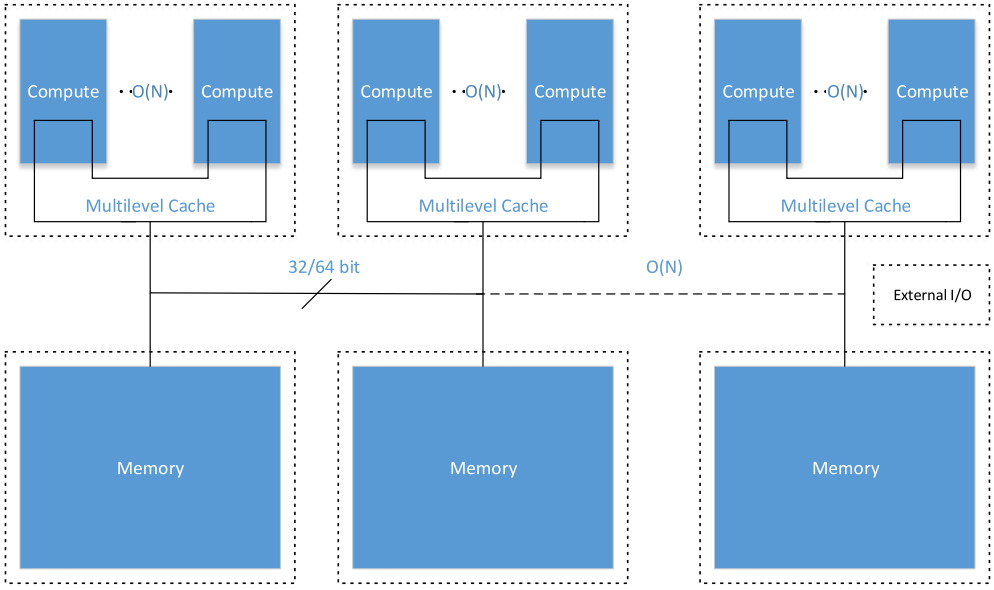
- Application

- OS Network Stack – Application layer

These use deep and wide multilevel architectures for compute and memory to accelerate performance. A cache is commonly used as a local store for memory. System memory is accessed on an External I/O interface. We are seeing a wall being met in single thread and single process performance. We have consequently moved to multi-threaded and multi-process designs. However, these are hitting a wall due to the overhead of maintaining coherence and synchronization in a multilevel cache and memory. The features are implemented in the constraints of the enterprise vendor or customer environment.

This is further exacerbated through latencies in accessing external interfaces whether in an external multilevel cache, memory or I/O (network). This highlights a wall in enhancing single threaded / process implementations using deep compute architectures. Additionally, multi-threaded / process implementations are hitting a fundamental design barrier and wall using a local cache memory store which has to be coherent and synchronized across internal and external I/O.

Figure 6: Conventional platform, multilevel architectures using wide, deep internal and external components.



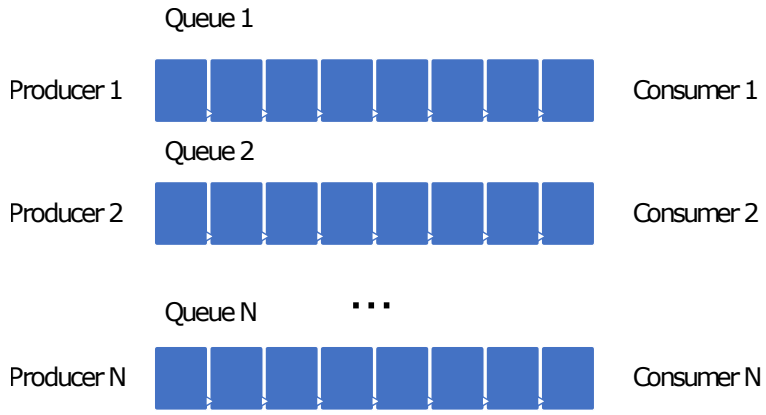
I/O access latency penalty (units)

- Internal 1 - 10's
- External 100 - 1000's

Figure 6 illustrates an architecture using a set of plug and play interfaces to scale performance. These are integrated in various topologies using a combination of internal and external I/O. Some

common topologies include Ring, Mesh, Star and Ad hoc. Trade-off considerations around performance, price and technology constraints partition the design across internal and external components. The topologies are constrained in the limitations of the interfaces. Increasingly architectures are consolidating these hierarchies in a single technology using compute and multilevel caches with wide and deep configurations.

Figure 7: Producer Consumer.



With the exponential growth (Moore's Law) in technology development we find that developers have increased access to commoditized compute (RISC, CISC) technology. Platforms based on commodity Linux solution are widely deployed in the enterprise. Application developers are concerned about application performance and scalability in the cloud. Application performance bottlenecks are constantly evolving in a tiered internet. They vary around system constraints limitations in the kernel functionality. However, application scalability is bounded in fundamental constraints of application development arising from a producer consumer model. The Producer Consumer or Bounded buffer problem is an example of a multi-process synchronization challenge. It forms a central role in any Operating system design that allows concurrent process activity.

As we have N producers, N consumers and N queues in the application Figure 7 we can see that there are opportunities for the synchronization through the use of semaphores, deadlock avoidance and starvation. If we imagine infinite resources, then the producer continues writing to the queue and the consumer has only to wait till there is data in the queue. The dining philosopher's problem is another demonstration of the challenges in concurrency and synchronization.

Figure 9 - The architecture addresses technology challenges in scaling the next generation internet including efficiency in the data center [6]. The architecture enables high performance commodity computing in the end user platform enabling technology of scale. Machine learning libraries like TensorFlow [3] can be programmed on a homogeneous architecture fabric. This facilitates an abstraction for the development of algorithms [2] and software on an open platform using a host of programming languages [8].

Figure 8: Proposed high-level architecture with compute (RISC, CISC) coupled to memory, no caches.

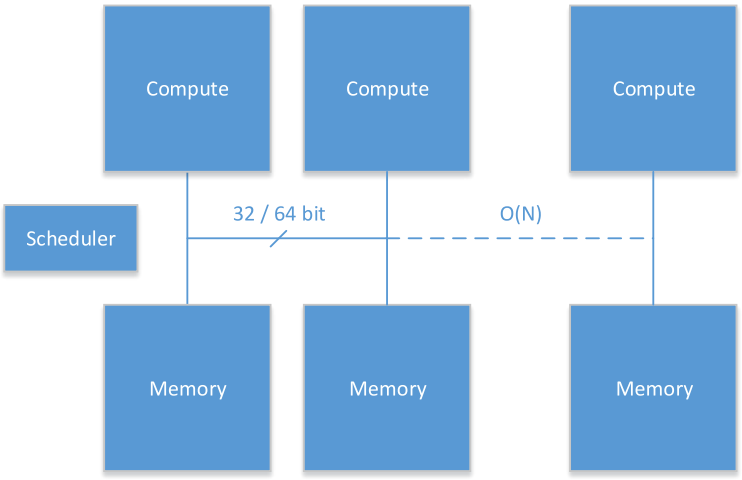
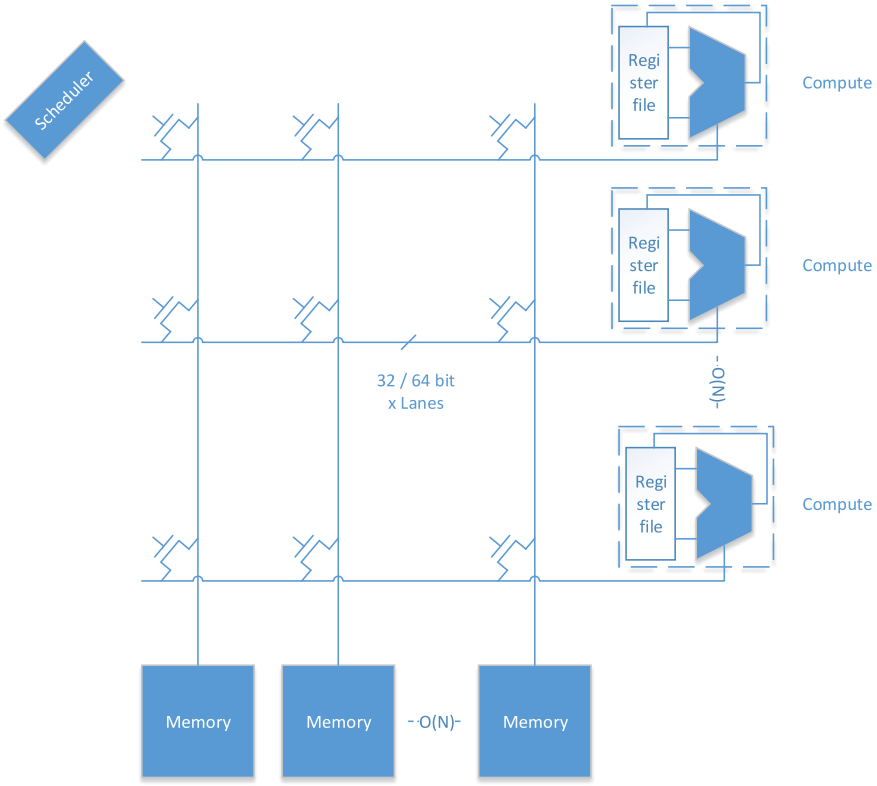


Figure 9: Proposed architecture with compute (RISC, CISC) coupled to memory using a crossbar switch, no caches. Architecture scales in underlying technology.



Conventional computing platforms:

- Processor >2GHz

- Memory >2GB

Proposed architecture (Figure 9):

- Network Bandwidth 1 – 10 Gbps

Shared bus implementation uses a Crossbar switch. These could use a buffer less design without any forwarding logic to access discrete memory banks using a bus select logic and multiple Lanes. A scheduler is used to access individual memory banks on the system bus. Scheduler could use a static round robin scheduler or a priority request response implementation. 3D Stacking technology enables integration of heterogeneous technology integrating DRAM (>1 GB) close to the Compute.

System bottlenecks are constantly evolving. As infrastructure is increasingly being commoditized with a growth of development around open source technologies. It is essential that adequate bandwidth is provisioned in the cloud to allow for application scalability. Virtualization technology enables efficient partitioning of additional resources. As a metric, it is key to replicate scale the infrastructure maintaining redundancy to ensure quality of service in the end-to-end internet.

3 Potential Benefits

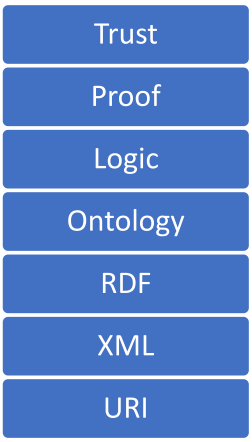
In the broader context of the internet it is always beneficial to host resources close to the client consumption including providing a larger bandwidth to the consumer. Additionally, open platforms and standards enable for a balanced distribution of available bandwidth resources allowing for a scalable platform for 10 billion consumers. Innovation and advancement are enabled through open source and open platforms around internet based wireless technology. The protocol stacks comprising the future semantic web data are as Figure 10.

- Web N. The internet is increasingly accessible to more than 10 billion users. It has been designed around Internet protocols and standards. The next generation of the web will use various Semantic web technologies.
- Cloud computing. Rapidly commoditized infrastructure and Linux servers
- Knowledge systems. Vast repositories of structured, unstructured data
- Efficient programming languages. Github

Proposed architecture:

- Multiprocessor and SRAM memory tightly coupled – no caches
- Generic configurable Network I/O – LAN, Wi-Fi, WiMAX, 5G, 4G

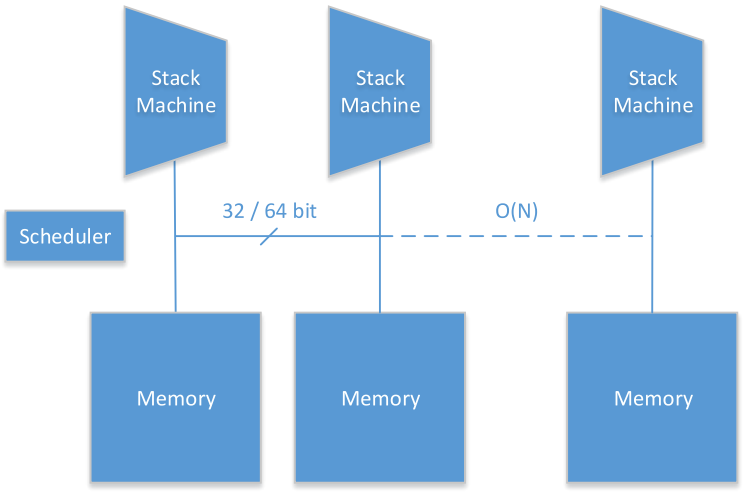
Figure 10: Semantic web stack.



- Programmable Network I/O - Network layers integrated in OS stack

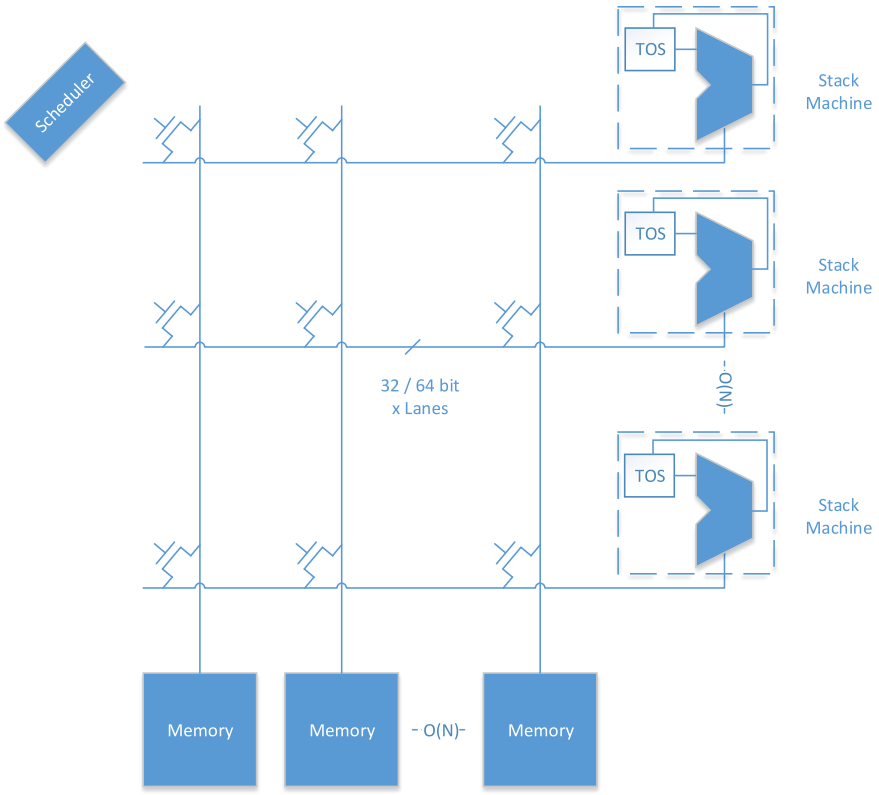
The Stack machine is a fundamental compute primitive. Processor and memory technology are now capable of integrating multi-Ghz and multi-Mb designs. There is a diminishing improvement for multi-Ghz processor designs as application memory is accessible in memory (DRAM).

Figure 11: Proposed high-level architecture with stack machine tightly coupled to memory, no caches.



We propose a high-performance general-purpose web computing platform using a tightly coupled processor (no caches L1, L2) and memory (SRAM) - Figure 12, Figure 9. A shared memory CMP architecture allows for a turn key, low cost solution to mobile connectivity allowing tight integration of processor technology and application specific software stack. A prototype device for mobile could contain 4 - 8 compute and 1 - 8 MB of SRAM memory [9]. O(1) $\leftrightarrow$ O(N). Machine learning

Figure 12: Proposed architecture with stack machine tightly coupled to memory using a crossbar switch, no caches. Architecture scales in underlying technology.



applications could be run efficiently on the platform [11].

As the device becomes accessible to more markets, we would see increasing accessibility to the internet [1]. Internet accessibility in a low-cost device enables scale in markets and applications. The platform uses general purpose processors in a shared memory environment to enable better programmability. Internet platforms enable open standards for technology development.

Figure 13: g++ compiler.

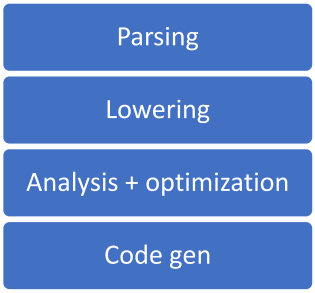
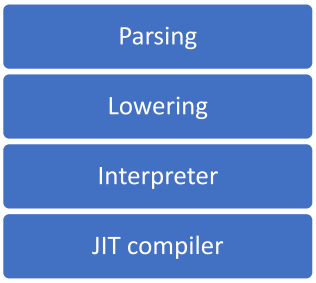


Figure 14: V8 JS.



Compilers and translators. Compiler is a Sequential batch architecture. Compilers translate

information from one representation to another - Figure 13, Figure 14. Most commonly, the information is a program. Compilers translate from high-level source code to low-level code. Translators transform representations at the same level of abstraction.

- Windows - 50 million LOC
- Google internet services - 2 billion LOC

Some designs use a virtual machine Eg. JVM that runs on the target architecture. Increasingly designs are converging around web ecosystems and the JavaScript Developer frameworks using a JIT compiler. The proposed solution supports both native implementation and those using a virtual machine.

4 Business impact

Programming languages are supported in a specific Software vendor stack Eg. C++ / C# (Microsoft), Java (Oracle), Python / JS (Google) to create a community developer ecosystem. These were conventionally developed around vendor specific platforms such as the PC Desktop, Mac or Mobile etc.

Figure 15: Converged memory stack.

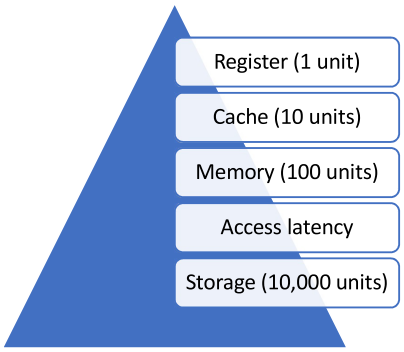


Figure 16: Converged compute stack.

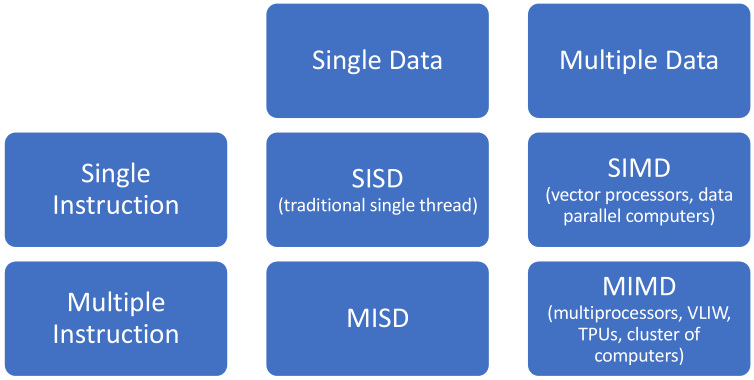
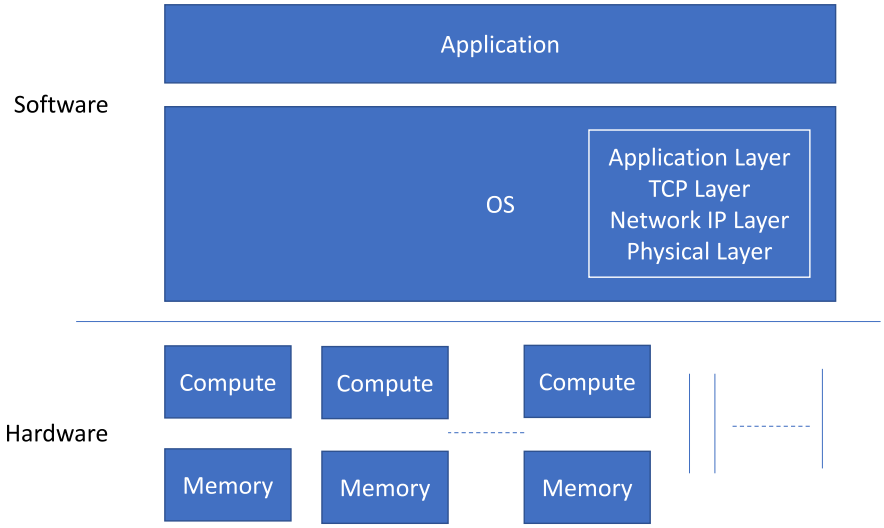


Figure 15, Figure 16 show the converged memory and compute stacks in the Enterprise vendor software ecosystem. A specific combination of these primitive's can be configured in the end-user application requirements such as networking or tensors [5]. These would use micro and macro consideration tradeoffs. Specific implementations of these designs are incentivized economically in the Software vendor ecosystems stack in a tiered technology industry leveraging a set of diversified business models.

Figure 17: Proposed architecture stack.



Defined Platform architecture specification solution is agnostic to the underlying technology. The Programming language is the layer of abstraction. We facilitate an abstraction layer for the development of Software and Algorithms.

- Layered architecture
- Pay as you go
- Open Source
- QOS requirement are guaranteed in the tiered Cloud Service provider
- Interoperability
- Scalability

We are seeing a broader industry wide convergence and disruption. The idea is a vital cog in the technology stack.

- Modus ponens, Conjecture

The architecture is a unified approach to bring next generation cognitive, low cost, mobile internet. The end user platform is able to scale as per the application requirements and network requirements in an efficient manner to improve cost, time to market, energy and accessibility - Figure 17. It takes computing out of the data center and into end user platform enabling an internet of scale for the next century. Internet enables open standards, accessible computing and applications programmability on a commodity platform.

- Faster time to market
- Increase quality and efficiency - Common Architecture Pattern
- Cost effective development of AI Solutions
- On premise or Managed deployment
- Lower cost of maintenance and support
- Pluggable support for multiple Vendors

5 Conclusion

Increasing number of devices are being connected to the internet. The internet is an open platform for next generation technologies [7]. Open platforms enable better collaboration and innovation. The future of the internet is mobile as >1 billion devices go online on IP. The presented architecture is a CMP design based on commodity processor and memory technology. We have an architecture with N (10's - 100's) compute connected to multi-MB SRAM memory using a shared memory system bus architecture. Number of compute and memory can scale in the power and performance requirements of the platform and the technology generation.

6 Highlights

- We propose an end to end architecture for the next generation web infrastructure.
- Multiprocessor and SRAM memory tightly coupled - no caches. Shared memory system bus (crossbar switch).
- Homogeneous architecture. The Stack machine is a fundamental compute primitive.
- Applications can be developed and deployed on a multitude of host platforms.
- A prototype device could contain N compute and N MB of memory. O(1) $\leftrightarrow$ O(N).
- Broader industry wide convergence and disruption. Idea is a vital cog in the technology stack.

References

[1] Android developer guide. `http://developer.android.com/guide/index.html`.

[2] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[3] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[4] T. J. Berners-Lee. Information management: A proposal. Technical report, 1989.

[5] F. Chollet et al. Keras, 2015.

[6] J. Corbet, A. Rubini, and G. Kroah-Hartman. *Linux Device Drivers, 3rd Edition.* O'Reilly Media, Inc., 2005.

[7] S. Faulkner, S. Moon, T. Leithead, A. Eicholz, and A. Danilo. HTML 5.2. W3C recommendation, W3C, Dec. 2017. https://www.w3.org/TR/2017/REC-html52-20171214/.

[8] A. Hejlsberg, M. Torgersen, S. Wiltamuth, and P. Golde. *C# Programming Language.* Addison-Wesley Professional, 4th edition, 2010.

[9] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[10] T. Kgil, S. D'Souza, A. Saidi, N. Binkert, R. Dreslinski, T. Mudge, S. Reinhardt, and K. Flautner. Picoserver: Using 3d stacking technology to enable a compact energy efficient chip multiprocessor. In *Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS XII, pages 117–128, New York, NY, USA, 2006. ACM.

[11] A. Kumar, S. Goyal, and M. Varma. Resource-efficient machine learning in 2 kb ram for the internet of things. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1935–1944. JMLR. org, 2017.

[12] R. Kurzweil. The law of accelerating returns. In *Alan Turing: Life and legacy of a great thinker*, pages 381–416. Springer, 2004.

[13] T. Lindholm, F. Yellin, G. Bracha, and A. Buckley. *The Java Virtual Machine Specification, Java SE 8 Edition*. Addison-Wesley Professional, 1st edition, 2014.

[14] Z. Mahmood and S. Saeed. *Software engineering frameworks for the cloud computing paradigm*. Springer, 2013.