NSF CSR PI Meeting June 02, 2017
Report on storage systems breakout group discussion

Ali R. Butt
butta@cs.vt.edu

Introduction

The big data revolution along with the dawn of the age of Internet of Things (IoT) is driving the need for novel and innovative storage systems to store, manage, retrieve, and efficiently utilize unprecedented volumes of data. The Storage Systems breakout session participants explored open challenges and research issues that need to be addressed both in the short and long term to ensure sustained storage systems efficacy and performance.

Emerging Research Problems in Storage Systems

The group identified a number of emerging research problems. The wide variety of applications and use cases entail that the fundamental design of storage systems is revisited to support application-specific and application-defined semantics. Such tailored design will address many of the shortcomings of using the extant one-size-for-all generic approach that is plagued with inefficiencies both in performance and storage capacity overhead. Management of metadata, indexing, and high-speed transactions for small data items are key to next generation storage systems. Another aspect is to reevaluate the use of standards and APIs such as POSIX, and design new sustainable data representations and APIs to support emerging applications such as IoT. The group recognized that such redesign is already in progress. For example, systems such as key-value stores are being used to design application-specific solutions. However, new methods and techniques are needed to support a wider range of emerging applications.

Hardware advances are further driving the way storage systems are realized. Storage hybridization and heterogeneity is now a part of most scalable storage deployments. However, the software subsystems for supporting and using such systems are lagging, and new management systems need to be designed and invented. Similarly, in-memory storage systems and persistent memory systems are giving rise to a new class of memory-only storage solutions. While such systems can be thought of as simply a tier of a traditional storage hierarchy, there is clearly a need for innovation to leverage the unique opportunities offered by new hardware advancements and designing the storage layer to maximize efficiency and performance.

At scale, the concept of resource de-segregation is being adopted in data centers and high performance computing resources. The impact of such approaches on storage provisioning and utilization also needs to be explored.

Mature Research Problems

The group also identified two mature research problems in storage systems domain. One is to revisit system-level design, e.g., context switches, for emerging fast storage technology, e.g., NVMe, etc., where system-level overheads can have significant opportunity cost. Thus, such operating systems details need to be examined to handle new storage architectures. On the flip side, the group agreed that issues surrounding hard disk drives management, Flash Translations layers, etc. are now mature. Solutions in this domain are robust and sufficiently sustained by industry. These aspects may not need further direct NSF investment, however partnership with storage manufacturing and design industry is highly desirable to further reap the benefits of such mature technologies.

Need for Larger Collaborations

The group identified several grand problems in the storage systems, which demand larger collaborative teams from different research domains to come together. Such problems include a clean slate storage system design to support features of scale, efficiency, security, and ultra-high-speed transactions. Another problem is how to address, design, and implement solutions for data management for IoT and other massive-scale system deployments. Furthermore, there is a strong need for collaborating with industry partners, e.g., in-memory based storage developers, to design sustainable and scalable hybrid and emerging storage support.

Potential Future Directions

Finally, the future of storage systems hold approaches such as active storage devices where computation is moved much closer to or within the storage layer. Similarly, intelligent storage systems driven by large-scale data analysis and machine learning approaches are foreseen. Here, the system dynamically fine-tunes storage parameters and semantics to create a fluid storage solution that adapts to the application needs not only at the high-level management layer, but throughout the storage stack from on-media data-layout optimization to API customization and representation.

Summary of PROGRAMMING LANGUAGES,COMPILER, RUNTIME SYSTEMS BREAKOUT SESSION

Programming languages, compilers and run-time environments address the expression and the optimization of computation and data which are fundamental for all application domains. Past research has developed high-level programming languages and parallel program languages and advanced theories and techniques to improve program performance and correctness. During the breakout session at the NSF CSR PI meeting, the attendees discussed new research challenges, which are summarized as follows:

To continue the development of new languages and compiler techniques, e.g. parallel languages for shared and distributed memory machines, general-purpose and domain-specific languages, and optimizing compilers, to support modern computing systems and applications. Modern applications include irregular computations such as sparse and graph problems. Modern systems include massively parallel, specialized and heterogeneous systems in processing, memory, storage and communication technologies. One of the most important common challenges is exploiting locality and reducing data movement. A promising approach is to develop a symbiotic relationship with the fields of data analytics, with programming support improving the speed and scale of data analytics on the one end and the improved data analytics serving program analysis and optimization on the other end.

To create new theories, abstractions, algorithms, techniques, platforms and pedagogy to address problems not just program functions but also performance for both traditional problems of speed, scalability, memory usage, power, and response time and emerging problems of resource management, approximation, adaptation, resilience, reliability, location, connectivity and other aspects of a complex system.

To identify and address significant challenges in new application areas such as mobile, wearable, virtual reality and edge computing.

Summary of discussion on Operating Systems and Virtual Machine

The participants of our breakout section had an engaging discussion that focused on identifying the emerging research problems in the area of operating systems and virtual machines.

One of our observations is that emerging hardware often drives research in this area. We identified several important hardware advances that are likely to necessitate innovation in systems software in the near future in order to fully realize the benefit of new technology. The looming impact of persistent memory (NVRAM, etc.) offers exciting opportunities because this technology challenges us to rethink the entire storage stack. Researchers must find methods to preserve important storage abstractions such as atomicity, consistent recovery, and a clean separation of transient and persistent data while imposing only minimal performance overhead so as to maintain the speed of persistent memory. GPUs and other heterogeneous compute options are becoming increasingly essential for modern applications; systems researchers must provide abstractions that better integrate these resources into general-purpose software systems and make exploiting heterogeneous compute resources easier for application programmers. Heterogeneity is also increasingly impacting IoT and mobile platforms, requiring operating systems and virtualization researchers to identify better methods for abstracting and managing a diverse array of sensing and actuating devices under stringent performance and energy requirements. Finally, increasing hardware support for virtualization in mobile platforms offers the opportunity to leverage this hardware to provide better security, isolation, and functionality for the mobile application ecosystem.

Several emerging classes of applications are also likely to drive systems research in the future. Operating systems and virtual machines can provide dedicated support and abstractions for machine learning and emerging AI applications. We expect the need to support such applications to affect platforms ranging from cloud servers to mobile devices to embedded platforms in autonomous vehicles and other domains. Increasingly, computation is being performed by applications with extremely large in-core memory footprints; these applications require efficient support for dynamic management of large amounts of local and remote memory. As massively distributed computation becomes more common, reducing tail latency becomes a priority. It is therefore vital that operating system and virtual machine researchers strive to reduce performance variability at all layers of the system, so that applications can achieve more predictable performance across similar executions.

Although operating systems and virtual machines have traditionally treated applications as black-boxes, white-box approaches that reason about application behavior promise to improve both reliability and performance in complex software systems. One promising approach is verification of software throughout the entire software stack to enforce important security and reliability properties statically. Another emerging approach is dynamic or static analysis of causality across OS and application boundaries to troubleshoot bugs, improve performance, and increase software reliability. Both of these approaches may lead to cross-layer optimization across the application/OS/VM boundary, which can both reduce software complexity and improve performance.

Opportunities for clean-slate OS design exist in several areas, including security and moving from human-to-service communication models to service-to-service communication. Greater compartmentalization of the OS and runtime components can make developing complex software systems easier and aid in maintenance by making it easier to isolate, identify, and deal with failures. The cost of compartmentalization can be potentially mitigated by automated white-box techniques described previously.

Finally, we agreed that the OS and VM communities should strive to develop better benchmarks, similar to the Top500 in the HPC community and SPEC/PARSEC/etc. in the hardware and programming languages communities. This will help provide greater emphasis on repeatable evaluation of computing systems.

NSF CSR PI Meeting, June 2, 2017, Orlando, Florida
Report on Breakout Session: Integrated Networked Systems and Internet of Things

Saurabh Bagchi
Purdue University

Last update: June 19, 2017

A group of about 25 NSF CSR PIs got together for the breakout session on "Integrated Networked Systems and Internet of Things". There was constructive discussion with most of the attendees contributing new ideas or expanding on ideas being discussed. We focused mostly on the question of what are the important research challenges that the CSR community needs to focus on, within the ambit of the definition of our breakout session. Toward the end of our session, we did an informal vote on the most pressing ideas that the community needs to do further work on.

I categorize the topics under 3 threads:

1. Non-functional attributes
2. Functional attributes
3. Social and economic factors

1. Non-functional attributes

I. ***Security***: It is important to secure these networked devices, especially as some of them will be embedded in our physical spaces, and many of them will carry sensitive personal or business information. The **security has to be achieved in an energy efficient manner**, considering the limited energy resources for many of the devices. This implies that the key generation and key verification have to be done in a manner that does not drain too much energy since the latter in particular may be a frequent operation. The **key management also has to be done in a scalable manner** and such that it respects the device constraints (such as, lack of floating point hardware). The **devices and the network must guarantee user privacy** because, as mentioned above, they may sense and carry sensitive private or business-critical information. This aspect takes particular meaning when multi-party automated negotiations will happen with the private data, among devices belonging to different administrative domains or owners. The driver for such negotiations is to provide more tailored services, at the right time, but the challenge will be to preserve the appropriate, user-controlled level of privacy in all such negotiations[1].

II. ***Resilience***: This encompasses making the overall system resilient to failures of individual devices or parts of the network. We expect that such failures will be more frequent than in traditional server-class systems. Such resilience demands **clean fault isolation boundaries** (elements within the boundary are affected by the fault, but those outside are not), which in turn relies on a clear understanding of the **fault propagation characteristics**. We should develop techniques and tools which let us come up with the above two primitives in a generic way that is applicable to a variety of target systems. The general principle to strive for is how to **design a resilient system, out of individual components that are not inherently resilient**, to failures in the cyber or the physical domain.

III. ***Safety***: Some of the target systems will be used for safety-critical applications. It is important to design techniques that can **validate the safety properties of such systems**. Safety implies that if the system fails, it transitions to a safe state, *i.e.*, one where human life or well-being is not threatened. In such systems, the function will have to be completed within reasonably short time bounds. While safety in real-time systems is a well-understood topic, additional challenges arise here because **the devices are more constrained in resources** and are at a lower price point, and **the systems are comprised of a large number of such devices**.

2. Functional attributes

I. ***Heterogeneity and interoperability of devices and networks***: There is likely to be a profusion of devices and networks from multiple vendors. For this field to flourish, it is important that such **heterogeneous components can interface with one another**. This involves designing APIs for integrated development of such systems. This also involves putting in place mechanisms for **discovering resources and services at runtime**. Due to the large scale of the anticipated systems, there is a requirement for **large-scale systems management solutions**. Again, we can take a leaf out of the book of managing traditional digital devices today, such as in enterprise settings, but adapt them to the challenges of this environment, as enumerated in point 1 above.

II. ***Systems modeling***: Much of the system development in this field is done in an ad hoc manner, and there is the need to have **rigorous systems modeling and evaluation of the**

---

[1] The participants were cognizant of the fact that some of these concerns overlap with the scope of the SaTC program. But there are system design challenges to enable the security and the privacy, which we felt falls squarely within the scope of the CSR program.

model. We can reuse existing systems modeling tools, but they will have to be **adapted to work at multiple scales**, both spatially (from small dust-like embedded devices to handhelds) and temporally. In addition to the reuse, the field may also require **the development of new modeling formalisms**, *e.g.*, to handle the large scale, the uncertain physical environments, or the low duty cycle of some of the constituent devices.

III. ***Context awareness***: It is important to imbue the software that will run on our target systems with **awareness of the context** in which the device is placed and in which the software is executing. The context is multi-dimensional. First, there is the **physical context** of the environment in which the device is embedded. Second, there is the **computational context** denoting the context in which the software is executing (*e.g.*, this computation needs to be completed within 10 seconds, this computation can cede priority to this other kind of computation as long as they are from the same system owner). Third, there is the **security context** denoting the security sensitivity of the computation or the data that the computation will generate.

3. Social and Economic Factors

I. ***Usability of systems***: **How humans will interact with the system** should be a prime factor driving the design and the development of such systems. Usability is important for most systems, but particularly for this domain, since many of the devices will be embedded in the physical spaces that we inhabit at home or at work. This will lead to interactions that are frequent, multi-modal, and perhaps in many cases, short lasting. A complementary question to study is **how is human behavior moderated and modified** due to the presence of such systems in our midst.

II. ***Smart contracts***: There is the need to develop **smart contracts for interactions among multiple system owners**, say for use of the data or the processing power being made available through the target systems. Contracts may also come into play among multiple OEMs that provide the devices that constitute the system. These contracts may be done at a fine time and space granularity and therefore **need to be more fine-grained and lightweight** than current solutions such as Ethereum.

III. ***Sustainability and fair use***: We have to consider end-of-life issues with these devices. We are used to changing our digital devices a lot more frequently than aspects of our physical spaces. When the two come together, how do we handle the issue of change. If our change cycles for the physical spaces are going to be accelerated due to the target systems, **how do we handle**

the maintenance issues after the end-of-life of a digital asset. Frequent obsolescence of our digital assets has already led to a **problem with mounting e-waste**. We need to carefully consider how this problem will be exacerbated and then handled due to the explosion of personal IoT devices that we are presaging. When a multitude of devices will co-exist in the same wireless spectrum, we have to set in play policy, and attendant technology, to ensure **fair usage of this shared resource**. This goes beyond wireless spectrum even, to include fair usage of computation made available, say through edge devices.

Summary statement

The domain of “Integrated Networked Systems and Internet of Things” is heralding a new age where the traditional divide between the cyber and the physical is starting to fall away. If properly harnessed, this can enable greater efficiency of our work lives, greater enjoyment of our personal lives, and greater safety, security, and resilience for our digital and physical assets. To achieve this trifecta requires solving several challenging research, deployment, and policy problems, as laid out above, in the space of functional attributes, non-functional attributes, and social and economic factors. The CSR community, with its prior work, is uniquely qualified to draw upon this experience to address the current and forthcoming challenges.

Summary of 2017 NSF PI Meeting – Hardware and Architecture Session

Several wide-ranging issues regarding new research directions and required community support in hardware technologies and computer architecture research were discussed during the discussion session.

As the end of Moore's law and Dennard scaling will happen within the next decade or so, the hardware and computer architecture community needs to explore innovative architectures to exploit new hardware technologies available on the horizon, for example, 3D integration, high-bandwidth memory (HBM), quantum computing, carbon nanotubes, etc. These new hardware technologies could allow system performance to be further improved together with improved power efficiency.

Another important recent trend is the emergence of many important new applications such as deep learning, virtual reality, autonomous systems, and internet-of-things. Many of such applications demands hardware specialization to achieve higher performance and power efficiency. FPGA-based systems and application-specific (ASIC) design are some of the approaches gaining importance.

With such diverse new hardware technologies and new applications, selecting appropriate hardware and software boundaries, i.e. the core of computer architecture research, will become crucial. The programmability and the system software support will also be critically important to build power efficient and high-performance computer systems. For recent biologically-inspired systems, e.g. DNA-based storage devices, we need to look for new computation models to encompass such drastically new hardware technologies.

To develop innovative computer architectures, new system abstractions and implementations are needed, in particular, for memory systems. As storing and moving data consume more power than computation itself, many new and existing memory technologies require new memory abstractions. Those memory technologies include high-bandwidth memory (HBM) that integrates both logic and memory devices on the same substrates, new persistent memory devices, and intelligent network-based storage systems.

Areas that require large team efforts within the community to make progress will include system design infrastructures that can support ASIC/FPGA and hardware/software co-design frameworks, as well as design space exploration, evaluation, and testing. With system specialization gaining more significance, such design and testing infrastructures will become indispensable.

NSF CSR PI Meeting
May 2017
Break out session on High Performance Computing
Summary
Viktor Prasanna

The breakout session on HPC started with a discussion of the evolving nature of HPC landscape including the infrastructure and applications. Discussions also focused on tools, access to resources, and some directions for NSF to consider. Following is a summary of the discussions:

1. **HPC Applications**: Traditional HPC applications have focused on large scale scientific and engineering simulations. It was felt new applications are emerging, for example, those focused on data science which lead to new workloads. The traditional architectures and the infrastructure is not optimized for these. There is also new directions being explored in us Machine Learning for complex applications.

2. **Programming, Performance Optimization and Code Maintenance:** New architectures are becoming increasingly complex. Providing environments for domain experts to effectively use these machines is an on-going challenge. While portability and productivity are some of the key requirements, performance optimization is another area worthy of further exploration. Maintaining and updating highly optimized code for complex architectures is a challenge faced by the larger community. Simplified programming models, for example, resource oblivious programming environments may offer abstractions for domain experts and reduce the barrier for adaptation of emerging platforms. Code adaptation (statically at compile time as well as at run time), performance portability across architectures and debugging tools were identified as key areas to further invest in. Programming models for HPC in Data Science is another emerging area that was discussed.

3. **Novel HPC platforms and Paradigms:** As the workload changes, new platforms and computing paradigms have evolved over the past few years. HPC on the cloud was briefly discussed. The key challenge is in reducing the cloud resource management and communication overheads to make clouds attractive for HPC workloads. HPC at the Edge was another direction that was discussed. It was noted that next generation domain specific platforms are being explored by communities including distributed systems and embedded systems which will enable very low latency computations at the Edge.

4. **Fundamental Understanding of HPC Systems**: With the complexity of evolving platforms and increase in the number of software layers it is increasingly difficult to model such systems for architecture design and application optimization. Simple high level models, tools to improve the accuracy of such models and possible use of machine learning in adapting such models for both system design as well as application acceleration are some key directions relevant to the community.

5. **Access to HPC Resources:** While the DoE community has access to (very) large scale facilities, it was felt the NSF community is somewhat constrained in access to such large machines. It was noted access to real machines is needed to conduct meaningful systems research by conducting at-scale experiments. While some large scale facilities are available for the community, the

overheads in obtaining access to these may deter research in using such platforms. It was also noted that with the current NSF investment in research in this area, the prior researchers may have migrated to seeking funding from DoE and other agencies focused on Exascale and other large systems efforts.

Summary of Breakout Session: Embedded and Real-Time Systems

The challenge of embedded and real-time systems is that they have to operate in a well-defined envelope of software and hardware characteristics, including execution time, resource usage (in particular energy), quality of computed answers, reliability/robustness, and security/privacy. In *hard* real-time systems, such characteristics have to be guaranteed, i.e., have to be valid for any application of the real-time system in its target operational environments. In *soft* real-time systems, the upper/lower bounds (worst case) of these characteristics are not guaranteed, but described by a probability distribution. For soft real-time systems, “violating” desired system characteristics may be tolerated, i.e., are not mission critical. The frequency and severity of violations that can be tolerated will be application dependent.

To address this challenge, embedded and real-time systems have to be designed “from the ground up” to be predictable/verifiable with respect to the characteristics mentioned above, both for hard and soft real-time systems, and in the presence of uncertainty. This has profound implications on the hardware and software design.

- Hardware and software implementations of system components have to come with guaranteed models of their behaviors with respect to the different system characteristics. Depending on the application, these models may target the worst case or probability distributions, where probability distributions reflect uncertainties.
- These models have to be compositional, allowing larger, complex systems to be defined and verified by the models of their component systems.

As a result, the panel has identified several specific research challenges:

1. Hardware designs with predictable and verified characteristics;
2. Energy efficient systems with predictable energy behaviors;
3. Approximations that enable tradeoffs between produced application outcomes and different system characteristics;
4. Security and privacy models and implementations;
5. Reliability and robustness models and implementations;
6. Programming abstractions and their implementations (e.g., APIs) that enable compositional design of real-time software.

In order to effectively address the listed research challenges, an open-source research and evaluation infrastructure is desirable, which includes hardware and software components, as well as evaluation benchmarks and application scenarios. The lack of a commonly accessible research infrastructure was highlighted by the panel for the areas of autonomous driving and aviation. Here, the industry has developed closed software and hardware systems that are hard or impossible to verify. Developing models and predictable systems on top of these closed components is therefore not possible. The panel therefore recommends that NSF reaches out to industry partners to encourage them to make their systems more open to the research community, or to fund efforts to develop a commonly accessible infrastructure for autonomous systems, including autonomous driving.

Edge and Cloud Computing

Breakout Session Report

CSR PI Meeting, June 2, 2017

Krishna Kant, Temple University

The breakout session asked us to focus on the following 4 questions:

1. What are the emerging research problems?
2. What are the maturing research problems?
3. What are the research problems that need larger collaborative teams to make progress
4. What will future computing system look like and how will they be used

The breakout session attracted a good number of researchers, which indicates interest in the area. The participants were asked to consider a wide variety of factors including: HW & SW architecture, distributed computing issues, applications, data management, business model & economics issues, service management, and all relevant cross-cutting issues such as security, energy, reliability, etc.

Emerging Research Problems:

The group mentioned many common issues among the emerging research areas such as energy usage, security, privacy & trust, fault tolerance & reliability, and cloud/edge management. One area of specific interest was in how to accelerate the build-out of the edge. In particular, what incentives could be put in place, who would own and deploy the edge infrastructure, and how various providers and users would collaborate in order to create the virtuous cycle of innovation and deployment. A related issue concerns heterogeneity and interoperability. A successful building of a comprehensive edge-computing infrastructure would involve many different players and thus result in substantial heterogeneity. Providing seamless services to the customer in spite of this heterogeneity poses many research challenges.

The edge infrastructure is envisioned to provide not only cloud computing services closer to the demand points, but also support a wide variety of other services including real-time data streaming servicers and even more critical cyberphysical system services. The latter could include support for intelligent transportation systems, physical security/surveillance, support for microgrids, health monitoring, etc. These applications are bring in many constraints including dynamism (shifting workload patterns), strict timing constraints, high robustness/reliability requirements, attack resistance, etc. This would require edge infrastructure to satisfy a wide range of requirements for various services supported and handle the changing workload intensity and nature (e.g., substantial capacity needed for intelligent transportation applications during rush hours).

Scalability of edge computing was another important issue. With edge devices deployed all around, it is extremely important to devise scalable mechanisms to manage both the edge infrastructure and the services provided by them – including diagnosis of problems, handling of misconfigurations, dealing with attacks and disruptions.

The question of maturing research areas was not well-articulated in the breakout session, perhaps because many traditional areas need revisiting in view of new usage models, services, and infrastructures.

Collaborative Research Needs:

With edge computing envisioned to support cyberphysical systems and smart city needs, the applications to be supported go much beyond computer science and require collaboration with researchers in many different domains. Similarly, building end-to-end solutions spanning the cloud, edge, fog, and device will require large-scale collaborative projects. Usability will be a substantial concern for these solutions, as they will be used by ordinary people, rather than by computer professionals. Usability encompasses the issues of intuitive interfaces, largely automated configuration, adaptability, and self-diagnosis of failures and misconfigurations, and self-healing properties. An associated challenge in large-scale projects is the establishment of test beds at scale.

What would Future Systems Look Like:

The discussion on this subject identified four key characteristics:

- Future systems will be application centric & will inherently consider human-in-the-loop issues. In particular, the questions of autonomy, robustness, exploiting and coping with human behavior will be an integral part of future systems.
- Future systems must pay close attention to policies and mechanisms in order to satisfy competing objectives for owners/providers for cloud, edge, & device level services.
- Future systems will be inherently data driven, will automatically analyze data in real time and make use of the insights to improve and adapt their functionality.
- Future systems will consider security & privacy issues from the beginning and build suitable mechanisms and tradeoffs from ground up.