Project Summary

Overview, Key Words, and Subtopic Name

Rack Scale Architecture is a model where one operating system shares the hardware resources of other machines in the same server rack to increase performance and the computing abilities of an operating system. Intel has created hardware solutions to achieve this, but they are very costly and inflexible. This product, Software-Defined Rack Scale Architecture (RSA) is a new feature for the Xen Hypervisor that emulates Intel's RSA through a software fabric instead of a hardware one. In this project, a virtualized operating system has the ability to leverage hardware from other machines. Individuals who manage data centers and clusters would benefit from this product since it enables them to design flexible computing architectures that make supporting this model trivial.

Keywords: rack scale architecture, rack scale vms, virtualization, memory management, internet of things, cloud computing

Subtopic: Rack Scale VMs through a Software Fabric

Intellectual Merit

The Small Business Innovation Research Phase I project will provide facilities for data center administrators to create flexible server architectures to account for the rapid growth of the IoT paradigm. This growth coupled with the advent of cloud computing pushes data center administrators to constantly rethink their server architecture to account for dynamic workloads.

Currently, resources in a data center are locked at the individual server level. Computing workloads will change with the aforementioned trends, thus requiring resources to adapt for them. Some data centers are unable to upgrade hardware as frequently as their workloads change. This increases strain hindering the data center incapable of efficiently adapting to the new workloads. Furthermore, RSA is typically achieved by hardware solutions. Products such as remote direct memory access (RDMA) enabled network cards and switches are required to enable sharing of computing resources across hosts.

This software solution to rack space architecture through the Xen hypervisor enables data centers to bring flexibility to existing hardware. Administrators can specify one master virtual operating system to share the resources within a specified computing pool. The master operating system will use intelligent algorithms to share and balance memory throughout the pool.

Broader/Commercial Impact

This innovation will provide facilities for commodity data centers or data centers who are unable to account for modern workloads to reliably scale. IoT and cloud computing are paradigms that are taking over the industry. A multitude of businesses are choosing the two as their main service providing backends. With this shift, there needs to be a way for data centers to reliably adapt to new workloads. RSA is the solution for this, but it is difficult to attain solely through hardware due to its high cost and inflexibility. This innovation provides the solution through a software fabric which is easy to adapt in data centers that are unable to reorganize resources or upgrade hardware.

Project Description

Elevator Pitch

Recent trends in the computing industry emphasize cloud computing and the "Internet of Things" (IoT). The former is a computing model where developers rely solely on "the cloud", or servers in a data centers, as their service provider. IoT refers to the ever-growing network of internet connected devices. The type of device connected to the internet includes more than just a computer or mobile phone, now we have light bulbs, refrigerators, laundry facilities, automobiles, and much more. These devices have very static requests that individually, are very simple to process. But the problem arises because of the volume of IoT devices surpasses the capabilities of a standard server. Intel estimates that the number of IoT devices will reach upwards of 30 billion by the year 2020. This means that servers processing requests for IoT devices will need to scale upwards, or process requests from an ever growing amount of devices. Currently, there is no easy way for a single server to adapt to handle such capacity without a hardware level upgrade.

To adapt to this growth and trend in computing, data centers need a means to dynamically scale their hardware. One solution is already present to cope with the trends--virtualization. With virtualization, one server can run thousands of isolated virtual machines. This technique alleviates data center stress in the cloud computing boom, but still does not provide an efficient means to handle the trends that IoT brings. IoT servers need a scalable set of physical computing resources. As the number of IoT devices increase, a server needs more random access memory, or RAM, to process requests. Traditionally, RAM is upgraded by physically swapping chips in the computer. This is a tedious job, and can prove to be very challenging when upgrading RAM in hundreds of servers.

Rack scale architecture (RSA) intends on providing a solution to this by giving one operating system access to other servers' hardware. Intel is a leader in developing this architecture, but their solution requires administrators to change server organization and install specialized hardware. This hardware becomes really expensive, thus making the optimized data center layout unreachable for most consumers.

This project, Software-Defined Rack Scale Architecture brings the novelty of Intel's RSA solution, but through a solely software fabric. Leveraging the open source Xen hypervisor, this product provides a virtual machine the means to share memory with other physical servers. Using Software-Defined RSA, a data center administrator is able to add physical servers to a resource pool with which the target operating system can attain additional resources.

This innovation provides mechanisms to handle new workloads that IoT and cloud computing bring. Specifically, this product focuses on providing intelligent memory management algorithms across physical servers. Traditionally, memory is shared across servers using remote direct memory access (RDMA) enabled network interfaces. These can be expensive. Using LibVMI, a virtual machine introspection library, the Xen hypervisor will be able to monitor when the target virtual machine tries to access data from a "remote" page of memory. Then, the hypervisor will be able to copy memory from the remote physical machine into the virtual machine. Furthermore, Xen will be responsible for intelligently mapping memory least used into remote physical machines while keeping the most used memory local to the virtual machines physical host. With this, the virtual machine will not suffer from the costs of retrieving and copying memory from a remote host as much as it could.

Commercial Opportunity

This innovation, Software Defined Rack Scale Architecture (RSA), will cater to the data center market. The data center market is comprised of systems administrators, systems developers, and Internet of Things (IoT) enthusiasts who will have a need to provide and use a platform that transparently scales to serve the growing number of IoT devices being used. Intel estimates that the number of IoT devices in use will be upwards of 30 billion by year 2020. Since data center and systems administrators will have to create a platform that scales well to account for trends in technology, they will find this innovation ideal to provide the highest quality of service.

Systems developers and IoT enthusiasts will always be following the trends in technology to create new products. Since the trends in technology show an increasing demand for IoT devices, developers and enthusiasts will provide. Vendors and retailers in technology are hiring these two types of developers to create more IoT devices. As these products are being released, their developers and their vendors/retailers will expect the end user to enjoy a high quality of service while using the IoT products. If the data centers hosting the backend for IoT devices are not adapting to the new trends in technology, then IoT developers will encourage the data centers to upgrade their platforms.

Data centers must adapt to new trends in technology to provide the highest quality of service to developers and their customers. If data centers fail to do so, the end users of an IoT device will not be happy and will cease to use the device. When end users stop using products, their creators and vendors will be negatively be affected since their profits will decrease. This decrease of profit is caused by the data center that they use to host their IoT device backend. The developer or vendor of the IoT device will pursue new data centers to host their backend. Ultimately, if a data center does not adapt to trends in technology, then they will lose contracts and not make a profit gain.

This innovation provides a flexible, software defined solution to address the IoT trends in technology. With this, data centers do not need to purchase new hardware or remodel existing platforms. Rather, they can provision existing hardware and server configurations into a RSA based setup. This flexible model is beneficial to data centers--now an administrator does not need to spend time and money reconfiguring and rebuilding existing server configurations, rather they simply install and configure software. Installing and configuring software is exponentially trivial compared to installing and configuring hardware. This triviality increases the profit gains that a data center will experience since they are minimizing the work to adapt to the IoT trends in technology.

Currently, there are no other competitors with this innovation. All advances in RSA are hardware defined. This innovation is one of the first software defined approaches to RSA which makes it very attractive. A hardware approach requires a massive redesign of data centers. New hardware needs to be ordered, installed, and configured to enable RSA within a data center. This software defined RSA solution does not need any of that. Instead, data center administrators are able to dismiss the overheads faced with new hardware and add support for RSA with ease. The competitive landscape might change as this innovation nears completion. It could change such that other companies such as Intel or VMware release software solutions to RSA. Intel is the creator of one of the first hardware solutions to RSA. VMware is a virtualization and cloud giant who is constantly innovating the industry. These two companies have the potential to release a software solution, but as of now, no evidence of this exists on their webpages.

Since there are not any competitors in the market as of now, a release of Software Defined RSA would be widely consumed by data center administrators. This innovation is planned to be an open source product. An open source product enables the release of this toolset to be made prematurely. Open sourcing this product enables the systems community to contribute to its end goal. The goals of this innovation are to provide the foundations to emulate RSA via a software fabric and to provide algorithms to manage memory across distributed physical servers. This is not a full solution to software defined RSA, but it is enough to encourage involvement by the community. Data center administrators and systems developers would readily consume the premature product working together to develop it further. Releasing this innovation prematurely also enables a quicker adoption of it. Since this product has the ability to adapt data centers to account for new trends in technology, administrators are able to justify investing time into it. If many data center administrators and systems administrators need a flexible and cheap approach to RSA, then the first product that they see will be the foundation to their solution. This means that this innovation will be adopted to be the standard software defined RSA solution.

The open source nature of this innovation means that the revenue gained directly from the use of this product is $0. However, revenue will be gained from a more support based model. Data centers who adopt this innovation will need support to install, configure, and maintain their RSA based servers. Data centers will be charged a service fee that scales according to the number of servers and the number of racks they are using in their RSA. Furthermore, data centers will be charged on the availability of service. For example, a data center who needs 24/7 support of their RSA will be charged more compared to the data center who wishes to have incidental support. This service oriented revenue model will be similar to RedHat's revenue model. Our revenue model will be comprised of various service tiers: tier 0 users will pay $0.00/year for no support; tier 1 users will pay $3,000.00/year for incidental support on each rack or a 10 server cluster; tier 2 users will pay $7,000.00/year for 24/7 remote support on each rack or a 10 server cluster; tier 3 users will pay $15,000.00/year for 24/7 onsite support on each rack or a 10 server cluster.

Providing these service tiers, tier 0 and 1 provide the greatest opportunity for security risks. Tier 0 or tier 1 users may configure their racks in insecure fashions. RSA introduced an assumed trust factor. Since an operating system shares the hardware resources of different physical servers, an assumption regarding trust has to be made. Users of RSA must assume that all servers within a rack are trusted. This means that the contents of memory are not malicious or corrupt. If the RSA based cluster is misconfigured and security holes are introduced, then this trust factor is violated and there is a potential data leak. This risk can affect all users of this innovation since its relative to the data that they are serving on top of this platform.

Societal Impact

Software defined Rack Scale Architecture (RSA) addresses scalability problems that data centers and service providers face when adapting to the Internet of Things (IoT) trends in technology. The groups of people that would be affected by this innovation are data center administrators or systems administrators, systems developers, IoT developers, and IoT users. These groups of people are a smaller niche within the information technology and the computer science industries. The affected groups of people compose the full stack of an IoT device or application. From IoT users, to IoT developers, to IoT hosts/supporters, and to systems developers, these groups of people will face change due to a software solution to RSA.

A software solution to RSA means enables data center administrator to provide a more scalable and reliable platform to host IoT backends that IoT developers create. Having a more reliable backend implies that IoT users will experience a greater quality of service with their devices. This impact creates a higher demand for IoT devices by end users since they are happier with what developers create. Developers trust their data centers moreso and can create more complex IoT devices and backends. Data centers will be able to scale more efficiently to handle the complex and new IoT devices. Using the methods that this innovation brings and the foundation that it creates for software based RSA, systems developers will be able to provide more features for it and drive the technology to grow even more.

Hardware is ever changing as well, and is being more optimized to handle more data and process it faster. This means that RSA needs to be updated to handle new hardware in technology. Having a software defined RSA implies that updating it is flexible and trivial compared to hardware defined RSA. This means that systems developers will be able to grow this innovation after its open source release to constantly keep it up-to-date with standards in technology.

There are not any environmental issues, health issues, or regulatory issues behind this software. However, there are data centers and cases where this provides some security holes. For example, if the RSA rack is misconfigured and the assumed trust is violated, then there is a critical security breach within the cluster. Data centers that are used by the federal government, health care providers, and other sensitive industries would need to take special care in assuring that trust is not violated. These users of the innovation need to either assure proper configurations or purchase the higher two service tiers to guarantee a secure configuration that does not leak data. In those industries, software defined RSA would need regulatory policies in place. There will be an online wiki outlining proper configuration to assure no security breaches exist in the rack.

The assumed trust constant introduces unethical use cases. For example, if a company purchases the greatest support tier, tier 3, then they receive onsite support. If the support technician has malicious intent, they are able to violate the trust constant and create a security breach. This breach in security can then be used by the technician to steal data or compromise sensitive operations. To account for this, our service technicians will be thoroughly vetted and be subjected to detailed background investigations to assure their compliance to our security policies. Furthermore, RSA service teams will provide periodic system integrity evaluations to tier 2 and above users to assure that their RSA cluster is free of security violations.

Technical Discussion and R&D Plan

Challenges and Risk

Software Defined Rack Scale Architecture (RSA) faces many challenges. A significant challenge will be creating a paging algorithm that intelligently determines when to swap memory pages of a process between the local host of the operating system to the remote memory server. One of the main properties of this application is to trick the operating system to believe that it has more physical memory than available to the local host by leveraging other physical hosts on the same server rack. This paging algorithm will need to be intelligent in the sense that it can determine which pages are "hot", accessed frequently, and "cold", accessed infrequently. Hot pages should remain locally, on the physical host of the operating system, and cold pages should be swapped to the remote memory server. Pages' status as either hot or cold dynamically changes as the operating system continues to execute. The paging algorithm needs to be aware of this, and intelligently determine when a page's status changes to remain performant. This algorithm is responsible for the performance of RSA. This project introduced a known bottleneck to system performance: the network. Swapping memory page contents already creates a performance hit on an operating system. Swapping a page over a network to a remote machine adds another bottleneck to the memory performance hit. This bottleneck is the network data transfer. In this project, this bottleneck is a known constant since it cannot be avoided. In lieu of this constant bottleneck, the intelligent memory paging algorithm needs to be as performant as possible to avoid further performance degradation.

There is also some risk introduced to a system using RSA. Before, all the system's hardware is in the same physical machine as the operating system. This means that the hardware is "trusted" and the system has full confidence that the hardware's integrity is maintained. This trust is an assumed constant. A system using RSA no longer uses only the hardware local to the operating system. It also uses the hardware on remote machines that are running the Memory Server component. To maintain system integrity, a user of RSA must make the assumption that the hardware of the operating system machine and the memory server machines are trusted. If a user does not trust other physical machines, then they should not be used as a memory server to the RSA application. This introduces more risk since a systems administrator must protect multiple machines compared to one to uphold system-wide integrity. Furthermore, if one memory server is compromised, then the user must assume that the entire system is compromised. The RSA application must be used within a single server rack that is air-gapped--the server rack is physically isolated from any unsecure or unknown networks. This is not something provided by RSA, nor something that will be focused on during the Phase 1 Project. However, this is a risk factor that systems administrators and users must take into consideration.

Technical Innovations

RSA's main innovations lie within the memory paging algorithm and the way it leverages existing technologies to handle memory swapping. Traditionally, RSA is a hardware-defined solution which provides hardware-level memory management algorithms. At this lower level, managing the memory is much easier than at the software level because it has direct access to memory. Managing memory at the software-level is more difficult since there are more layers of abstraction that an application has to go through. This implies that memory management will not be as fast. Software-Defined RSA provides an intelligent memory paging algorithm which brings memory

management to the software-level while being as performant as possible. The algorithm is intelligent, meaning it profiles running operating systems to determine which pages are more or less frequently accessed. A system profile will mark memory pages as either "hot" or "cold": hot meaning frequently accessed and cold meaning less frequently accessed. This profile is then used to assist swapping data to and from the remote memory server. Hot pages will be kept local to the operating system's hardware while cold data will be swapped with data stored in the remote memory server. The remote memory server can store memory data from the operating system. The algorithm also needs to account for restoring swapped data when it needs to be used again. Although the algorithm will not be as performant as raw hardware-level swapping, the intelligent nature of it ensures that it will not degrade performance to an unusable state.

RSA leverages the Xen Hypervisor and LibVMI to build its functionality. The Xen Hypervisor is a software that multiplexes a physical computer's hardware across many virtual machines. Xen also provides a management operating system, Dom0, which manages and shares resources that other virtual machines have. LibVMI is a virtual machine introspection library. This library enables monitoring and managing components of a virtual machine as it executes. LibVMI provides a way to monitor memory related events and manage memory contents of a virtual machine. Software-Defined RSA is built using LibVMI to interface with an operating system's memory and to determine when data should be swapped between the remote memory server. The RSA application runs inside Dom0, the management virtual machine running on Xen, and it manages the memory of another operating system running in a virtual machine. This is a novel use of Xen and LibVMI since RSA is not the intended application of LibVMI.

Phase 1 Objectives and Milestones

First, Software-Defined RSA must be able to detect memory events in an operating system, and copy data to and from that memory location. This is critical since it is the main functionality of the project. The RSA application must be capable of detecting memory events on marked pages, and then copy data from the remote memory server to and from those pages. Without the ability to detect and handle memory accesses, this project cannot facilitate its most basic task. The first objective is to ensure this functionality. Using LibVMI, a memory accesses on a marked page within the target operating system must trigger an event to the RSA application. Upon receipt of the memory event, the LibVMI based RSA application must copy data out of the VM, request data from the memory server, send the VM's data to the memory server, and then copy the newly acquired data into the the VM.

Once this functionality is established, the next milestone is to implement an algorithm which marks memory pages as "hot" or "cold" based on their access rate. This milestone brings the intelligence to the memory paging algorithm. Without this intelligent nature, the memory paging algorithm becomes a huge bottleneck if it tries to swap all memory pages. Furthermore, without an intelligent memory paging algorithm, increasing performance means marking target pages by hand. These target pages are the ones that will be monitored and swapped with the remote memory server. This responsibility would fall into the hands of the end-user. Commercially, this application would fail if an end-user has to provide that much detail and be that involved in the configuration of it. To improve Software-Defined RSA's commercial performance, usage needs to be as simple as possible. The most simplified way of using this product would be by specifying a target application and then depend on this product to manage memory swapping.

Another critical milestone for Software-Defined RSA is the ability to bring data that was swapped to the memory server back. If an application is in need of some data that was swapped to the remote server, the memory paging algorithm is responsible for requesting that data back. The RSA application would make a call to the memory server to retrieve the previously swapped data and it then replaces that data with what the operating system currently has in memory. Without this capability, swapping data renders that data lost forever. This means that the application that's being extended by RSA has to recreate data after swapping it. Not providing this feature would degrade performance and commercial ability.

Finally, the last milestone is combining all of the aforementioned milestones together. Software-Defined RSA must work as one seamless system. When the application is executed, it will profile the target application inside the target operating system to determine which pages are "hot" or "cold" With this profile, the application will then begin to swap cold pages of data to the memory server so the application can behave such that it has a greater amount of memory. Once the application needs a previously swapped memory page, the RSA application will request that data back from the memory server. All of this is possible due to the integration of LibVMI functionality, an intelligent memory paging algorithm, and leveraging the architecture of the Xen Hypervisor.

R&D Plan and Timeline

Initial goals involve investigating the use of LibVMI and Xen to provide memory event handling and copy memory content to and from a host. This needs to be completed and built first since it is a dependency for the rest of the features. Also, initial goals comprise of creating a memory server that can allocate memory, store swapped data, and serve requests. Lastly, initial goals comprise of creating a network library that allows the LibVMI based RSA application to communicate with the Memory Server.

Next, research will be focused on profiling process memory usage. Once a profiling method is secured, focus will be on researching memory paging algorithms. After a memory paging algorithm is determined, focus will shift to combining the two. The RSA Application needs an intelligent memory paging algorithm that profiles a process' memory footprint and then swaps hot and cold pages to and from the memory server.

Once the memory algorithm is intact, efforts will be focused on combining the algorithm with the platform created in the first set of goals. The LibVMI based RSA application needs to intelligently and automatically swap memory pages that are hot and cold. This step will require the most testing and experimentation to ensure its integrity. To test this component, RSA will provide example applications to be run within the virtual machine, or target operating system. One such example application will be a single memory access application. This application will be used to determine if one page access is handled--handling a page access means an event will be fired, data will be requested from the remote memory server, and the received data will be copied into the machine. A second example application will be created to be run inside the target operating system. This application will trigger the RSA application to detect many memory events, and it will test the RSA application's intelligent memory paging algorithm. A set of unit tests will complement this example application so the memory server can be tested to see how many pages are being used by the target operating system.

Revisions Made

In prior sections, Alan has made comments regarding voice, technical jargon, and paragraph level organization. I have made efforts to make my voice more active, remove clusters of technical jargon and replace it with more refined and explained thoughts, and reorganize paragraphs with defined topic sentences and sentence organization. I understand how the paragraphs seem unorganized, there are not well defined topic sentences to guide a reader through my thoughts. Adding these in will make sections clearer and easier to follow. Removing dense clusters of technical jargon will make the topic, and content more digestible by a greater variety of readers. Adapting my passive voice to an active voice will make the style seem more professional and more suited for a non-technical audience.