IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

ADVANCEMENTS IN HETEROGENOUS COMPUTING

Rishi Thakur
Student, Amity University Chhattisgarh, Raipur

Mr. Advin Manhar
Professor, Amity University Chattisgarh, Raipur

ABSTRACT

Several advancements in computer hardware and system fields have made advances in technologies that provide us high efficiency rates and faster processing with some induced challenges like heating and overloading of systems. In this paper, we discuss about some of these advancements.

Keywords - Provenance

INTRODUCTION

The aim of this research is to make seamless, efficient and a fundamental structure of future digital systems. A decade's worth of research has worked towards this vision of building these efficient systems through a software centric approach in order to capture and process provenance data.

One drawback of these system sis that they capture domain-specific coarse data and sometimes the data is incomplete.. These limitations are due to the fact that software based approach captures run-time performance data.

We assume future systems will overcome these limitations by capturing more accurate, precise and complete data.

Precision –

Precise data assures better confidence in the lineage of computation. The provenance capture of data naturally over laps with certain areas of computer science that generally perform similar functions like debugging, logging. Traditionally these functions have been software eccentric, but recent trends in technology support for these functions in hardware because hardware implementations impose least runtime overheads on user applications.

Completeness –

Completeness of data requires systems that captures lineage information across two dimensions i.e. within hosts and across hosts.

Within hosts, major changes in hardware architectures and application performances have become critical aspects of computers. Traditional computing on a host has been CPU centric wherein data is extracted from storage or memory, moving it closer to the CPU where computation is to be done. However, with the increased flexibility provided by the programmable SSDs and memory interfaces, we are achieving a change away from the traditional forms of computing.

Across hosts, systems have generally lacked observability due to specialized hardware in devices like routers, firewalls etc. However, as network service management is becoming critical, more emphasis is laid on community hardware that can run software implementing various network functions.

In the rest of the paper, we will discuss about specific technological advancements we can leverage and the various challenges that these technologies impose.
To this end, I focus on the area of Heterogenous Computing and Network Provenance

Heterogenous Computing –

Modern architecture of hardware has become increasingly heterogenous, comprising of different processing elements which are specialized to perform specific functions. The use of heterogenous architectures mainly began with the introduction of user programmable GPU's about a decade ago. Developers saw significant improvements in performance of their applications.
Heterogenous system architecture is a vast topic that goes beyond than just being confined to co-processors. Currently, there is a focus in Near Data Processing (NDP) research. NDP changes the computing paradigm by moving the processing power near to where the data is stored. This provides benefits in terms of power efficiency and reduced time complexity due to increase in performance speed as data does not have to move back and forth between the storage location and the CPU. NDP has been achievable due to the emergence of programmable storage or programable memory which allows us to directly execute the custom code on the device, closer to data.
In this section, we will discuss about three specific technologies in heterogenous computing –

(i) Programmable SDDS's
(ii) Co-processors

along with their implications for completeness of data.

(i) Programmable SSD –

Modern SSD's have inbuilt low-frequency CPU firmware that perform minimal functions such as error-correction, read/write caching etc. However, most SSD firmware is proprietary and does not allow the developer to add new functions.

Recent research shows a new interface in SSD's which allows the developers to directly offload certain application specific functions to directly run on the SSD. This gives us a huge improvement in performance, bandwidth, access latencies and reduced power consumption as data does not have to move back and forth between the SSD and the host.

Application for provenance –

Provenance-aware systems generally run on the host CPU either as apart of the OS or the application. When applications are directly performed on the SSD, the transformation will not be collected by the provenance collection system of the host CPU and hence the provenance capture will be incomplete. Hence, collecting and integrating data in these heterogenous environments makes provenance data capture more accurate.

Challenges –

The processing units inside SSD's have limited processing power due to low power consumption or power-efficient requirements. Addition of provenance capture functions to run on SSD's can overload the I/O performance, reducing the performance benefits provided by NDP.

(ii) Co-Processors –

There are many types of co-processor technologies in current day computing. The two most used common Co-Processors are GPGPU (General-Purpose Graphic Processing Units) and MIC (Many Integrated Cores). Co-processors enable higher degree of parallel processing using several hundreds of simple architecture based low frequency cores. Co-Processors are directly connected to the CPU using PCle. For using co-processors, all the data is copied from the CPU to the Co-processor and processed. After computation, the results are copied back to the CPU.

This computing paradigm is commonly used in high-performance applications as there are very heavy loads which require huge amounts of processing power.

Application for provenance –

Co-processors have their own memory. Hence, the computations and data transformations done by the co-processors are asynchronous and is not reflected in the operating system of the host CPU. Any updates made by the co-processor directly on the host system is not visible to the data collecting system on the host. Hence, co-processors should be mounted on the services which collect provenance data.

Challenges –

Co-processors are basically used to increase performance. Capturing data at critical paths can directly impact / hinder the performance advantages these technologies provide.

Network Provenance –

In recent years, developments in networking systems has lead to reduction in complexity of networking systems by dividing single complex entities into several simple smaller entities within the network. This advancement has lead to two results –

(i) Software Defined Networking
(ii) Network Functions Virtualisation

(i) Software Defined Networking –

SDN is an approach in which the data and control of the network are kept separate and the control activities are dedicated towards fixed hosts. This separation of controls and programmable systems has proved fruitful and SDN is leveraged to collect provenance data of individual data packets.

(ii) Network Functions Visualisation –

NFV applies the same approach as SDN of reducing complexity to special purpose devices such as firewalls, intrusion detection devices etc. NFV aims to implement these functions on virtual servers which are running on enterprise hardware, hence eliminating tight coupling these functions had with proprietary software.

Application for provenance –

Shifting towards commodity hardware which implements network functions and enables to run provenance systems as a service on these devices. This shift also helps in reducing design and development inputs as distinct devices can now run as virtual devices running on the same host server. Hence, we can collect more complete provenance data with lower input costs and also gain visibility across hosts.

Challenges –

NFV and SDN are still emerging technologies. Ensuring completeness of provenance data across multiple hosts requires synchronization and co-ordination across all hosts in the existing network system. Collecting precise data is limited due to the observable information disclosed at the end-host.

Conclusion

In This Paper, we discussed about several technological advances that provide opportunities for computer systems. We discussed about various technological advancements that have occurred in recent times.

REFERENCES

1. Nikhil Handigol, Brandon Heller, Vimalkumar Jeyakumar, David Mazires, Nick McKeown, ”I know what your packet did last hop: using packet histories to troubleshoot networks”, Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation (NSDI’14). USENIX Association, Berkeley, CA, USA, 71-85.
2. The European Telecommunications Standards Institute. Network Functions Virtualisation (NFV); Use Cases. GS NFV 001 (V1.1.1), Oct. 2013.
3. The European Telecommunications Standards Institute. Network Functions Virtualisation (NFV); Architectural Framework. GS NFV 002 (V1.1.1), Oct. 2013.
4. Big Data Analytics, Data Science & Machine Learning - Nvidia, http://www.nvidia.com/object/data-science-analytics-database.html

5. Sudharsan Seshadri, Mark Gahagan, Sundaram Bhaskaran, Trevor Bunker, Arup De, Yanqin Jin, Yang Liu, Steven Swanson, ”Willow: A User-Programmable SSD”, Proceedings of OSDI. 2014, 67-80.

6. Arup De, Maya Gokhale, Rajesh Gupta, Steven Swanson, Minerva: Accelerating Data Analysis in Next-Generation SSDs, Proceedings of the 21st IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM), April 2013

7. Jaeyoung Do, Yang-Suk Kee, Jignesh M. Patel, Chanik Park, Kwanghyun Park, and David J. DeWitt, ”Query processing on smart SSDs: opportunities and challenges”, In Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data (SIGMOD ’13). ACM, New York, NY, USA, 1221-1230. DOI=10.1145/2463676.2465295 http://doi.acm.org/10.1145/2463676.2465295

8. M. Chiosi et al. Network Functions Virtualisation: An Introduction, Benets, Enablers, Challenges & Call for Action. ETSI White Paper, Oct. 2012

9. Intel 64 and IA-32 Architectures Software Developers Manual, Chapter 36, vol.3C, pages 36.1-36.37, http://www.intel.com/content/dam/www/public/us/en/documents/manuals/64- ia-32-architectures-software-developer-manual-325462.pdf

10. Han, Bo, Vijay Gopalakrishnan, Lusheng Ji, and Seungjoon Lee. ”Network function virtualization: Challenges and opportunities for innovations.” Communications Magazine, IEEE 53, no. 2 (2015): 90-97.