SciCloud: Cloud for high-performance scientific computing

Joseph Anthony C. Hermocilla

1 Thesis

High-performance scientific computing can be done over cloud infrastructure and can be guaranteed a performance that is comparable to bare metal machines currently used in supercomputers or dedicated commodity clusters.

2 Rationale

Since the digital computer was invented, it has been an important tool in doing science. In fact, without it, progress in science will be quite slow. Several scientific breakthroughs have been made possible because of digital computers. From the sequencing of the human genome to the discovery of the Higg's Boson particle, the digital computer has its very important role. From simple arithmetic to modeling biological and physical systems, the digital computer is able to deliver the result of experiments in an accurate and timely manner.

Scientific applications, such as equation solvers and simulations, are very CPU-intensive and Memory-intensive. These applications, because of their heavy computational and storage requirements, are usually run in high-performance computing (HPC) infrastructures using supercomputers or commodity clusters for distributed computations. Setting up such infrastructures requires a lot of initial capital and technical investments. Once the infrastructure has been set up, maintenance costs will also be required to update the hardware and software components. Although these infrastructures are indeed useful, studies have shown that they are underutilized. Their usage depends on the computing demand pattern of scientists and researchers which often make these infrastructures idle. Most HPC clusters in research centers cater to a mix of researchers from different study areas. For example, physicists and chemists may share the same HPC cluster running on Linux operating system. Differences in usage requirements of these researchers affect the overall operation of the cluster. Provisioning the hardware and software requirements needed for scientific computing is thus very challenging because of these factors.

Despite the recent advances in hardware (increased processing speed and memory) and their decreasing costs, typical users do not harness the full capacity of their machines, either because the software they are using does not support them or their computing needs do not require access to the full capacity of their machines. Web browsing and word processing, for example, usually do not require high-performance machines whereas gaming needs them for an enhanced user experience. Another example is the case of laboratory computers used for instruction in academic institutions. Despite having the latest processors and a good amount of memory, they are typically used just for simple programming exercises which do not require the full access to hardware performance. These underutilized machines can be used for scientific computing needs.

Machines used in clusters are usually dedicated, which means that they cannot be used for other general-purpose computing needs. When running tasks, the number of machines used from the clusters used are usually statically set often with the assistance of system administrators managing the cluster. A scientist who needs to perform a simulation need to contact the cluster administrator to schedule the job that must be run.

2.1 What is Scientific Computing?

According to Wikipedia, Scientific Computing, also known as Computational Science, involves the use of computers for solving problems in science. It is concerned with model construction, which is mostly mathematical, as well as quantitative analysis. It is also involved in computer simulation and other forms of computation that are used to solve problems in various scientific disciplines. Scientists develop software that models the system being studied and run them with various parameters. The nature of the models requires a lot computations(floating point operations) and data.

2.2 Desktop Computers, Supercomputers, Clusters, and Grid Computing

The nature of software created for science, involving complex calculations, will require a different approach to their implementation and execution. Desktop computers are useful for small scale simulations but are slow for complex experiments. Supercomputers are computers, way beyond the capabilities of desktop computers, that have a specialized architecture optimized for floating point operations. However, these types of computers are usually very costly. Clusters are a collection of typical desktop computers connected through a local area network and is less costly. They execute scientific applications in a distributed manner usually using distributed shared memory or message passing. Grid computing extends the capabilities of clusters by going beyond the local area network extending through a wide area network, in a totally distributed manner. Grid computing is usually federated.

2.3 What is Cloud Computing?

Cloud computing is a recent buzzword in computing. The diversity of applications running on top of the Internet, from e-commerce and banking to social networks, forces the vendors to address the issue of scale. In the case of social networking sites, for example, the continuously increasing number of users will demand additional physical computing resources to be provisioned. Application developers who develop software can focus on implementing functionality instead on optimizing code to address increasing demand.

The widely accepted definition of cloud computing is from the National Institute of Standards and Technology[54]. It defines cloud computing as

> "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction . This cloud model promotes availability and is composed of five essential characteristics , three service models , and four deployment models ."

There are five main characteristics of cloud computing[54]. First, on-demand access allows users to provision computing resources without requiring human interaction from the service provider. Second, the capabilities are available over the network accessible through standard mechanisms. Third, the resources are pooled and serve multiple users in a multi-tenant model. The physical and virtual resources are dynamically assigned and reassigned depending on the demand. Fourth, provisioning of capabilities is elastic to quickly scale out and rapidly released to quickly scale in. Lastly, cloud systems provide a metering capability at some level of abstraction appropriate to the type of service.

Service models for cloud computing include Software-as-a-Service(SaaS), Platform-as-a-Service(PaaS), and Infrastructure-as-a-Service(IaaS)[54]. SaaS allows users to access and use software running on a cloud infrastructure, for example Google Docs. PaaS allows users to deploy their own applications in a cloud infrastructure using programming languages and frameworks supported by the provider. IaaS provides the user with the most basic computing resources such as processor, storage, and network.

There are four ways to deploy a cloud[54]. A private cloud is operated for a single organization which can be managed by the organization itself or a third party and can be housed within or outside the organization's premises. A community cloud is shared by several organizations to support a specific community. Public clouds are available for the general public. Lastly, hybrid clouds is a composition of two or more clouds.

Another notable characterization of cloud computing was developed at the University of California Berkeley[8].

3 Related Work

A few studies have been made regarding the use of cloud for scientific computing. In 2010, Ostermann et. al. conducted a performance analysis of EC2 cloud computing services for scientific computing using microbenchmarks and kernels. In 2011, Iosup et. al. conducted a performance analysis of cloud computing services for Many-Tasks Scientific Computing[40]. In 2012, Zhao et. al. presented an early effort in designing and building CloudDragon, a scientific cloud platform which is based on OpenNebula. Most recently, Ludescher et. al. developed a cloud-based execution frameworks for scientific problem solving environments which combines a private and public cloud setup[48].In 2010, Schad et. al. performed runtime measurements in the cloud and noted the variability due to the multitenant nature of the cloud[66].

4 Objectives

The main objective of this research is to develop cloud computing solutions to improve high-performance scientific computing. Specifically, this work aims to

1. evaluate existing public cloud offerings (Amazon EC2, Microsoft Azure) for applicability in scientific computing tasks;
2. develop and deploy a private cloud over commodity desktop computers which satisfy scientific applications requirements;
3. develop a novel framework and platform to make it easy and efficient for scientists to develop scientific applications; and
4. deploy a private cloud with software for scientific applications.

5 Methodology

To achieve Objective 1

1. Obtain accounts from Amazon EC2 and Microsoft Azure.
2. Review existing benchmarks and execute them on the platforms mentioned above.

To achieve Objective 2

1. Identify key characteristics of scientific applications in terms of computation, storage, and network.
2. Deploy a 5-node private cloud setup for testing.
3. Study internal architecture of existing cloud computing frameworks (Eucalyptus[56] and OpenStack[69]) to identify modules that can be modified to support the requirements of scientific applications.

4. Design and implement a cloud system with an architecture and implementation specific to support scientific applications.

To achieve Objective 3

1. Identify key characteristics of scientific applications in terms of architectural and data representation.
2. Identify patterns in creating scientific applications and develop software abstractions.
3. Create an application programming interface (API) for scientific applications.
4. Create sample applications using the newly created API.

To achieve Objective 4

1. Survey a group of scientists to identify their most commonly used scientific applications.
2. Partially implement the features of the identified applications using the API developed in Objective 3
3. Deploy the application as a Software-as-a-Service.

References

[1] Home Â» OpenStack open source cloud computing software.

[2] Performance analysis of high performance computing applications on the amazon web services cloud.

[3] vonLaszewski-08-cloud.pdf.

[4] Orna Agmon Ben-Yehuda, Muli Ben-Yehuda, Assaf Schuster, and Dan Tsafrir. Deconstructing amazon EC2 spot instance pricing. *ACM Transactions on Economics and Computation*, 1(3):1–20, September 2013.

[5] Jeongseob Ahn, Changdae Kim, Jaeung Han, Young-ri Choi, and Jaehyuk Huh. Dynamic virtual machine scheduling in clouds for architectural shared resources. *Proceedings of HotCloud*, 12, 2012.

[6] Glenn Ammons, Vasanth Bala, Todd Mummert, Darrell Reimer, and Xiaolan Zhang. Virtual machine images as structured data: the mirage image library. *Usenix HotCloud*, 2011.

[7] Renuka Apte, Liting Hu, Karsten Schwan, and Arpan Ghosh. Look who's talking: discovering dependencies between virtual machines using CPU utilization. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, pages 17–17, 2010.

[8] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, and Ion Stoica. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.

[9] Jason Baker, Chris Bond, James Corbett, J. J. Furman, Andrey Khorlin, James Larson, Jean-Michel LÃ©on, Yawei Li, Alexander Lloyd, and Vadim Yushprakh. Megastore: Providing scalable, highly available storage for interactive services. In *CIDR*, volume 11, pages 223–234, 2011.

[10] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the art of virtualization. *ACM SIGOPS Operating Systems Review*, 37(5):164–177, 2003.

[11] Doug Beaver, Sanjeev Kumar, Harry C. Li, Jason Sobel, and Peter Vajgel. Finding a needle in haystack: Facebook's photo storage. In *OSDI*, volume 10, pages 1–8, 2010.

[12] Anton Beloglazov, Sareh Fotuhi Piraghaj, Mohammed Alrokayan, and Rajkumar Buyya. Deploying OpenStack on CentOS using the KVM hypervisor and GlusterFS distributed file system. Technical report, Technical Report CLOUDS-TR-2012-3, Cloud Computing and Distributed Systems Laboratory, The University of Melbourne, 2012.

[13] David Bermbach, Markus Klems, Stefan Tai, and Michael Menzel. Metastorage: A federated cloud storage system to manage consistency-latency tradeoffs. In *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, pages 452–459, 2011.

[14] Filip Blagojevic, Cyril Guyot, Qingbo Wang, Timothy Tsai, Robert Mateescu, and Zvonimir Bandic. Priority IO scheduling in the cloud.

[15] Tom Bostoen, Sape Mullender, and Yolande Berbers. Power-reduction techniques for data-center storage systems. *ACM Computing Surveys*, 45(3):1–38, June 2013.

[16] Kevin D. Bowers, Ari Juels, and Alina Oprea. HAIL: a high-availability and integrity layer for cloud storage. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 187–198, 2009.

[17] Chris Bunch, Navraj Chohan, and Chandra Krintz. Appscale: open-source platform-as-a-service. Technical report, UCSB Technical Report 2011-01, 2011.

[18] Mike Burrows. The chubby lock service for loosely-coupled distributed systems. In *Proceedings of the 7th symposium on Operating systems design and implementation*, pages 335–350, 2006.

[19] Brad Calder, Ju Wang, Aaron Ogus, Niranjan Nilakantan, Arild Skjolsvold, Sam McKelvie, Yikang Xu, Shashwat Srivastav, Jiesheng Wu, and Huseyin Simitci. Windows azure storage: a highly available cloud storage service with strong consistency. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, pages 143–157, 2011.

[20] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. Bigtable. *ACM Transactions on Computer Systems*, 26(2):1–26, June 2008.

[21] Susanta Nanda Tzi-cker Chiueh and Stony Brook. A survey on virtualization technologies. *RPE Report*, pages 1–42, 2005.

[22] N.M. Mosharaf Kabir Chowdhury and Raouf Boutaba. A survey of network virtualization. *Computer Networks*, 54(5):862–876, April 2010.

[23] Brian F. Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan, and Russell Sears. Benchmarking cloud serving systems with YCSB. In *Proceedings of the 1st ACM symposium on Cloud computing*, pages 143–154, 2010.

[24] Antonio Corradi, Mario Fanelli, and Luca Foschini. VM consolidation: A real case based on OpenStack cloud. *Future Generation Computer Systems*, (0).

[25] Jeffrey Dean and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[26] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels. Dynamo: amazon's highly available key-value store. In *SOSP*, volume 7, pages 205–220, 2007.

[27] Mr Bhupesh Kumar Dewangan and Mr Sanjay Kumar Baghel. Issues, research and implementation of cloud computing. *International Journal of Science, Engineering and Technology Research*, 1(4):pp–20, 2012.

[28] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. Technical report, DTIC Document, 2004.

[29] Niroshinie Fernando, Seng W. Loke, and Wenny Rahayu. Mobile cloud computing: A survey. *Including Special section: AIRCC-NetCoM 2009 and Special section: Clouds and Service-Oriented Architectures*, 29(1):84–106, January 2013.

[30] Sahan Gamage, Ramana Kompella, and Dongyan Xu. vPipe: one pipe to connect them all!

[31] Saurabh Kumar Garg, Steve Versteeg, and Rajkumar Buyya. A framework for ranking of cloud computing services. *Special Section: Utility and Cloud Computing*, 29(4):1012–1023, June 2013.

[32] Gary Garrison, Sanghyun Kim, and Robin L. Wakefield. Success factors for deploying cloud computing. *Communications of the ACM*, 55(9):62–68, 2012.

[33] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The google file system. In *ACM SIGOPS Operating Systems Review*, volume 37, pages 29–43, 2003.

[34] JoÃ£o V. Gomes, Pedro R. M. InÃ¡cio, Manuela Pereira, MÃ¡rio M. Freire, and Paulo P. Monteiro. Detection and classification of peer-to-peer traffic. *ACM Computing Surveys*, 45(3):1–40, June 2013.

[35] Nelson Gonzalez, Charles Miers, Fernando Redigolo, Marcos Simplicio, Tereza Carvalho, Mats Naslund, and Makan Pourzandi. A quantitative analysis of current security concerns and solutions for cloud computing. *Journal of Cloud Computing: Advances, Systems and Applications*, 1(1):11, 2012.

[36] Albert Greenberg, James Hamilton, David A. Maltz, and Parveen Patel. The cost of a cloud: research problems in data center networks. *ACM SIGCOMM Computer Communication Review*, 39(1):68–73, 2008.

[37] Ajay Gulati, Ganesha Shanmuganathan, Anne Holler, and Irfan Ahmad. Cloud-scale resource management: challenges and techniques. In *Proceedings of the 3rd USENIX conference on Hot topics in cloud computing*, pages 3–3, 2011.

[38] M. Hedges, T. Blanke, and A. Hasan. Rule-based curation and preservation of data. *Future Generation Computer Systems*, 25:446 – 452, 2009.

[39] James Horey, Edmon Begoli, Raghul Gunasekaran, Seung-Hwan Lim, and James Nutaro. Big data platforms as a service: challenges and approach. In *Proceedings of the 4th USENIX conference on Hot Topics in Cloud Ccomputing*, pages 16–16, 2012.

[40] A Iosup, S Ostermann, M N Yigitbasi, R Prodan, T Fahringer, and D H J Epema. Performance analysis of cloud computing services for many-tasks scientific computing. *IEEE Transactions on Parallel and Distributed Systems*, 22(6):931–945, June 2011.

[41] Seny Kamara, Charalampos Papamanthou, and Tom Roeder. Cs2: A searchable cryptographic cloud storage system. *Microsoft Research, TechReport MSR-TR-2011-58*, 2011.

[42] Karthik Kumar and Yung-Hsiang Lu. Cloud computing for mobile users: Can offloading computation save energy? *Computer*, 43(4):51–56, 2010.

[43] Gunho Lee, Byung-Gon Chun, and Randy H. Katz. Heterogeneity-aware resource allocation and scheduling in the cloud. In *Proceedings of the 3rd USENIX Workshop on Hot Topics in Cloud Computing, HotCloud*, volume 11, 2011.

[44] Seung-Hwan Lim, Jae-Seok Huh, Youngjae Kim, and Chita R. Das. Migration, assignment, and scheduling of jobs in virtualized environment. *Migration*, 40:45, 2010.

[45] Xing Lin, Yun Mao, Feifei Li, and Robert Ricci. Towards fair sharing of block storage in a multi-tenant cloud. In *Proceedings of the 4th USENIX conference on Hot Topics in Cloud Ccomputing*, pages 15–15, 2012.

[46] Dionysios Logothetis, Chris Trezzo, Kevin C. Webb, and Kenneth Yocum. In-situ mapreduce for log processing. In *2011 USENIX Annual Technical Conference (USENIX ATC'11)*, page 115, 2011.

[47] Flavio Lombardi and Roberto Di Pietro. Secure virtualization for cloud computing. *Advanced Topics in Cloud Computing*, 34(4):1113–1122, July 2011.

[48] Thomas Ludescher, Thomas Feilhauer, and Peter Brezany. Cloud-based code execution framework for scientific problem solving environments. *Journal of Cloud Computing: Advances, Systems and Applications*, 2(1):11, 2013.

[49] Prince Mahajan, Srinath Setty, Sangmin Lee, Allen Clement, Lorenzo Alvisi, Mike Dahlin, and Michael Walfish. Depot: Cloud storage with minimal trust (extended version). Technical report, Technical Report TR-10-33, Department of Computer Science, The University of Texas at Austin, 2010.

[50] S. M. Mahbub Habib, S. Ries, and M. Muhlhauser. Cloud computing landscape and research challenges regarding trust and reputation. proceedings IEEE 2010 symposia and workshops on ubiquitous. *Autonomic and Trusted Computing*, pages 410 – 415, 2010.

[51] Sean Marston, Zhi Li, Subhajyoti Bandyopadhyay, Juheng Zhang, and Anand Ghalsasi. Cloud computing - the business perspective. *Decision Support Systems*, 51(1):176–189, April 2011.

[52] Viktor Mauch, Marcel Kunze, and Marius Hillenbrand. High performance cloud computing. *Including Special sections: High Performance Computing in the Cloud & Resource Discovery Mechanisms for P2P Systems*, 29(6):1408–1416, August 2013.

[53] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. OpenFlow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, 2008.

[54] Peter Mell and Timothy Grance. The NIST definition of cloud computing (draft). *NIST special publication*, 800(145):7, 2011.

[55] Antti P. Miettinen and Jukka K. Nurminen. Energy efficiency of mobile clients in cloud computing. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, pages 4–4, 2010.

[56] Daniel Nurmi, Rich Wolski, Chris Grzegorczyk, Graziano Obertelli, Sunil Soman, Lamia Youseff, and Dmitrii Zagorodnov. The eucalyptus open-source cloud-computing system. pages 124–131. IEEE, 2009.

[57] Simon Ostermann, Alexandria Iosup, Nezih Yigitbasi, Radu Prodan, Thomas Fahringer, and Dick Epema. A performance analysis of EC2 cloud computing services for scientific computing. In *Cloud Computing*, pages 115–131. Springer, 2010.

[58] John Ousterhout, Parag Agrawal, David Erickson, Christos Kozyrakis, Jacob Leverich, David MaziÃšres, Subhasish Mitra, Aravind Narayanan, Guru Parulkar, and Mendel Rosenblum. The case for RAMClouds: scalable high-performance storage entirely in DRAM. *ACM SIGOPS Operating Systems Review*, 43(4):92–105, 2010.

[59] Michael Pearce, Sherali Zeadally, and Ray Hunt. Virtualization. *ACM Computing Surveys*, 45(2):1–39, February 2013.

[60] GÃ¡bor PÃ©k, Levente ButtyÂŽan, and BoldizsÃ¡r BencsÃ¡th. A survey of security issues in hardware virtualization. *ACM Computing Surveys*, 45(3):1–34, June 2013.

[61] Ling Qian, Zhiguo Luo, Yujian Du, and Leitao Guo. Cloud computing: an overview. In *Cloud Computing*, pages 626–631. Springer, 2009.

[62] Asfandyar Qureshi, Rick Weber, Hari Balakrishnan, John Guttag, and Bruce Maggs. Cutting the electric bill for internet-scale systems. *ACM SIGCOMM Computer Communication Review*, 39(4):123–134, 2009.

[63] Micheal G. Reed, Paul F. Syverson, and David M. Goldschlag. Anonymous connections and onion routing. *Selected Areas in Communications, IEEE Journal on*, 16(4):482–494, 1998.

[64] Bhaskar Prasad Rimal, Eunmi Choi, and Ian Lumb. A taxonomy and survey of cloud computing systems. pages 44–51. IEEE, 2009.

[65] Chunming Rong, Son T. Nguyen, and Martin Gilje Jaatun. Beyond lightning: A survey on security challenges in cloud computing. *Special issue on Recent Advanced Technologies and Theories for Grid and Cloud Computing and Bio-engineering*, 39(1):47–54, January 2013.

[66] JÃ¶rg Schad, Jens Dittrich, and Jorge-Arnulfo QuianÃ©-Ruiz. Runtime measurements in the cloud: observing, analyzing, and reducing variance. *Proceedings of the VLDB Endowment*, 3(1-2):460–471, 2010.

[67] Malte Schwarzkopf, Derek G. Murray, and Steven Hand. The seven deadly sins of cloud computing research. *HotCloud, June*, 2012.

[68] Roland Schwarzkopf, Matthias Schmidt, Christian Strack, Simon Martin, and Bernd Freisleben. Increasing virtual machine security in cloud environments. *Journal of Cloud Computing: Advances, Systems and Applications*, 1(1):12, 2012.

[69] Omar Sefraoui. OpenStack: toward an open-source solution for cloud computing. *International Journal of Computer Applications*, 55(3):38–42, October 2012.

[70] Alan Shieh, Srikanth Kandula, Albert Greenberg, and Changhoon Kim. Seawall: performance isolation for cloud datacenter networks. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, pages 1–1, 2010.

[71] Josef Spillner, Johannes MÃŒller, and Alexander Schill. Creating optimal cloud storage systems. *Special Section: Utility and Cloud Computing*, 29(4):1062–1072, June 2013.

[72] S. Subashini and V. Kavitha. A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications*, 34(1):1–11, January 2011.

[73] Sahil Suneja, Elliott Baron, Eyal De Lara, and Ryan Johnson. Accelerating the cloud with heterogeneous computing. In *Proceedings of the 3rd USENIX conference on Hot topics in cloud computing*, pages 23–23, 2011.

[74] Aryan TaheriMonfared and Martin Gilje Jaatun. As strong as the weakest link: Handling compromised components in OpenStack. pages 189–196. IEEE, November 2011.

[75] Byung Chul Tak, Bhuvan Urgaonkar, and Anand Sivasubramaniam. To move or not to move: The economics of cloud computing. In *Proceedings of the 3rd USENIX conference on Hot topics in cloud computing*, pages 5–5, 2011.

[76] Vijay Vasudevan, Amar Phanishayee, Hiral Shah, Elie Krevat, David G. Andersen, Gregory R. Ganger, Garth A. Gibson, and Brian Mueller. Safe and effective fine-grained TCP retransmissions for datacenter communication. In *ACM SIGCOMM Computer Communication Review*, volume 39, pages 303–314, 2009.

[77] Hoang Tam Vo, Chun Chen, and Beng Chin Ooi. Towards elastic transactional cloud storage with range query support. *Proceedings of the VLDB Endowment*, 3(1-2):506–514, 2010.

[78] Simon Waddington, Jun Zhang, Gareth Knight, Jens Jensen, Roger Downing, and Cheney Ketley. Cloud repositories for research data - addressing the needs of researchers. *Journal of Cloud Computing: Advances, Systems and Applications*, 2(1):13, 2013.

[79] Cong Wang, Qian Wang, Kui Ren, and Wenjing Lou. Privacy-preserving public auditing for data storage security in cloud computing. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9, 2010.

[80] Hongyi Wang, Qingfeng Jing, Rishan Chen, Bingsheng He, Zhengping Qian, and Lidong Zhou. Distributed systems meet economics: pricing in the cloud. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, pages 6–6, 2010.

[81] Xiaolong Wen, Genqiang Gu, Qingchun Li, Yun Gao, and Xuejie Zhang. Comparison of open-source cloud management platforms: OpenStack and OpenNebula. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on*, pages 2457–2461, 2012.

[82] David Wentzlaff, Charles Gruenwald III, Nathan Beckmann, Kevin Modzelewski, Adam Belay, Lamia Youseff, Jason Miller, and Anant Agarwal. An operating system for multicore and clouds: mechanisms and implementation. In *Proceedings of the 1st ACM symposium on Cloud computing*, pages 3–14, 2010.

[83] Yong Zhao, Yanzhe Zhang, Wenhong Tian, Ruini Xue, and Cui Lin. Designing and deploying a scientific computing cloud platform. *Grid Computing (GRID), 2012 ACM/IEEE 13th International Conference on*, pages 104–113, September 2012.

[84] Andrew J. Younge, Robert Henschel, James T. Brown, Gregor von Laszewski, Judy Qiu, and Geoffrey C. Fox. Analysis of virtualization technologies for high performance computing environments. pages 9–16. IEEE, July 2011.

[85] Saman Zarandioon, Danfeng Daphne Yao, and Vinod Ganapathy. K2C: cryptographic cloud storage with lazy revocation and anonymous access. In *Security and Privacy in Communication Networks*, pages 59–76. Springer, 2012.

[86] Wenying Zeng, Yuelong Zhao, Kairi Ou, and Wei Song. Research on cloud storage architecture and key technologies. In *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, pages 1044–1048, 2009.

[87] Qi Zhang, Lu Cheng, and Raouf Boutaba. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1):7–18, April 2010.

[88] M. Zhou, R. Zhang, W. Xie, W. Qian, and A. Zhou. Security and privacy in cloud computing: A survey. *Proceedings IEEE Sixth International Conference on Semantics, Knowledge and Grids*, 2010:106 – 112, 2010.