Systems and Networking Ph.D. Qualifier Exam

Research Questions

Spring 2006

Students are expected to provide written answers to questions by 11:59pm EST on Sunday, January 22, 2006. Please e-mail your answers both in a PDF format and in a "source" format (e.g., LaTeX or Microsoft Word) to gback@cs.vt.edu.

There are six questions on this examination. Each of the six questions will be given approximately equal weight in determining the overall grade. However, as indicated below, the parts of each of the six questions are not necessarily given equal weight in determining the score for that question.

Read the questions carefully and address them fully. You are expected to deal with the questions as posed. Nevertheless, if you have substantial doubt about a question, or believe a question is flawed, please contact Dr. Godmar Back (gback@vt.edu) as soon as possible.

The parts of a question on this examination are varied in the level of knowledge and creativity required to provide a full and complete answer. Some parts of a question require only comprehension or application of the concepts presented in the readings on which the questions are based. Other parts of a question require analysis, synthesis, evaluation, or extension of those concepts. Although it is necessary to answer the former type of questions correctly, the grading of the examination will place greater emphasis on the answers to the latter form of questions. Each question provides in an "instruction/hints" section additional guidance on the form of the desired answer and the relative emphasis to be given to each part in the grading.

This examination is conducted under the University's Graduate Honor System Code (http://www.gradhonor.grads.vt.edu/). Students are encouraged to draw from other papers than those listed below, to the extent that this strengthens their arguments. However, the answers submitted must represent the sole and complete work of the student submitting the answers. Material substantially derived from other works, whether published or on the web, must be explicitly and fully cited. For more information on proper citation of sources, we strongly recommend that you study the guidelines and examples given in Appendix IIIa of the Constitution of the Graduate Honor System. Note that your grade will be more strongly influenced by arguments you make rather than arguments you cite.

1. Quantum Architectures

In "Building Quantum Wires: The Long and the Short of it," Oskin et al. [5] discuss possible implementations of quantum scale architectures.

a) Explain how teleportation channels are able to communicate data. Be specific.

b) For swapping channels and teleportation channels answer the following:

   i. How fast is the resulting communication?

   ii. What are the inherent advantages and disadvantages of this approach versus conventional bus-driven approaches?

   iii. What role does error correction play in this type of communication?

c) In your opinion:

   i. What is the likelihood systems using these technologies will actually come to the market?

   ii. What benefits (if any) will applications see if the technologies emerge on real systems?

   For both part i) and ii), independent of which position you take, explain your position and support your opinions with facts!

Instructions/Hints*: Limit your answer to no more than 3 pages total. Parts a) and b) try to gauge your understanding of how quantum communication may work in practice. Focus on the ways in which the transmission occurs rather than details such as choosing phosphorus versus another element. Ideally, you should discuss the basics of EPR pairs and the part they play in communication. Overall, describe the technology from a computer science perspective, not a physics perspective. Part c) asks you to critically evaluate the paper and the likelihood it will lead to new technologies in the marketplace. It is important that you substantiate your position with facts whenever possible.*

2. Memory Design

In "LIRS: An Efficient Low Interreference Recency Set Replacement Policy to Improve Buffer Cache Performance," Jiang and Zhang [2] present a variation on the LRU algorithm.

a) Describe the limitations of the LRU page replacement policy. Describe a synthetic application (not from the paper) you could implement on most systems that would perform poorly under LRU. Be specific, you may provide source code or pseudo code as necessary.

b) Describe the advantages of LIRS page replacement policy. Apply the LIRS policy to your example from part a). Does LIRS perform better, worse, or the same? Why? Be sure to describe step-by-step how the algorithm selects pages for replacement.

Instructions/Hints: *Limit your answer to no more than 4 pages total including source code if any. This example tests your underlying understanding of memory systems. Your example should work on most memory systems. It might be helpful to actually implement the code on a real system for proof of concept, but this is not required.*

3. Event Filtering and Notification for Distributed Systems

The extensible event-based infrastructure for networked virtual worlds described by Pubrick et al. in [10] and the wide-area event notification service described by Carzaniga et al. in [12] both provide infrastructures that support the construction of dynamic, extensible event-based systems.

a) What is, in your opinion, the key to scalability in both approaches? Be specific!

b) Statement: *The wide-area event notification service [12] provides a more generic, application-independent infrastructure, when compared to the extensible event-based infrastructure [10], hence the event-based infrastructure in [10] may be expressed as a special case of [12].*

   By comparing and contrasting those two approaches provide arguments and justifications for or against that statement.

c) The wide-area event notification service [12] is application-independent so it could be used for networked virtual worlds. Propose a scenario where a networked virtual world is built on top of the wide-area notification service. You can use as an example the application described in [10] and replace the distributed event filter. You can also use some other networked virtual world application.

***Instruction/Hints**: Limit your answer to no more than 4 pages total. Parts a) and b) require a basic understanding of the concepts presented in the papers. Part c) asks for your ability to make a judgment. There is no right or wrong answer; rather we are looking for your ability to take a position and substantiate it with arguments.*

4. Distributed File Systems

Distributed file systems form the basis for many distributed systems and applications. There are many distributed file systems, often with only subtle architectural differences.

a) What are, in your opinion, distinguishing characteristics for the distributed file system described by Grönwall et al. in [11], as compared to other distributed file systems?

b) Compare a distributed file system of your choice (e.g. NFS, Coda) with the distributed file system described in [11]. Clearly specify comparison criteria and principles used.

c) The wide-area event notification service [12] is application-independent so it could be used for distribute file systems. Propose a scenario where a distributed file system is built on top of the wide-area notification service. How feasible is this approach? Be specific!

Instruction/Hints*: Limit your answer to no more than 4 pages total. Part a) requires a basic understanding of the concepts presented in the papers. Parts b) and c) ask for your ability to make a judgment. There is no right or wrong answer; rather we are looking for your ability to take a position and substantiate it with arguments.*

5. Failure-Oblivious Computing

Rinard et al. propose failure-oblivious computing [8] as a technique for surviving software failures.

a) Construct an example of a software error from which failure-oblivious computing could recover without introducing adverse side effects. Be specific! For your example, discuss the behavior of the alternatives discussed in the paper, i.e., standard and safe compilation.

b) Construct a specific scenario in which Rinard's approach would fail catastrophically. Be specific! For your scenario, discuss the behavior of the alternatives standard and safe compilation.

c) Would Rinard's approach be applicable to kernel code, such as code that is part of the Linux kernel? If so, how? If not, why not? Discuss specifically the case of guarding against failures in extensions (kernel modules.) State your assumptions if necessary!

d) Rinard claims that type-safe languages such as Java or ML can benefit from failure-oblivious computing as well. Examine under which conditions programs written in type-safe languages might benefit from failure-oblivious computing. How big would the benefit be, in your opinion?

e) Failure-oblivious computing has received a fair amount of skepticism; among other things, for its assertion that robust programs can be built by ignoring error conditions. Take a critical position either for or against failure-oblivious computing and argue your position. Make sure you include in your discussion other approaches to providing robustness with which you are familiar.

Instructions/Hints*: For both parts a) and b), it is not acceptable to use examples from the paper (or subsequent papers that refer to that paper). Ideally, you would use examples you have encountered in your own coding/projects. Be sure to simplify the examples if necessary. You may use actual source code to illustrate your example, but don't provide only code.*

Part e) asks for your judgment and your ability to formulate and argue a position; this subpart has the highest weight. If you reuse arguments from papers you read, including Rinard's paper, make sure that you don't merely repeat them, but that you critically evaluate them.

6. Proportional Share Resource Management

This question asks you to explore the topic of proportional share scheduling of resources, and specifically the algorithms and techniques used in the ESX server and the GR^3 CPU scheduling algorithm.

a) Waldspurger [6] describes a proportional-share algorithm for memory, a space-shared resource. Caprita et al. [7] describes a proportional-share CPU scheduling algorithm. Discuss, in detail, how space-shared resources differ from time-shared resources when it comes to proportional-share scheduling. Outline what aspects are shared for between the two groups of problems, what the key differences are, and why solutions from one area may not carry over into the other one.

b) Waldspurger developed his techniques in the context of the ESX server, a virtual machine monitor.

   i. He states in Section 5.3 that it is not necessary to identify specific active and idle pages individually when sampling idle memory. Considering that ESX server uses demand paging on VM pages, explain why that is so!

   ii. In order to translate "physical" addresses to hardware addresses, would a software TLB require the use of shadow page tables?

c) Discuss which of Waldspurger's techniques would be applicable to the scheduling of physical memory among processes in a conventional OS, and which would need to be modified. Be sure to cover all of his contributions.

d) Conversely, Caprita developed their algorithm in the context of a regular OS. Could GR^3 be used in a virtual machine monitor such as the ESX Server? What would the benefit be and what potential problems do you see?

Instructions/Hints*: Although part a) is an open-ended question, limit your answer to no more than 3 pages. Be as concise in your answer as you can. Part b) asks for your understanding of the paper. Parts c) and d) test your ability to apply the techniques in the paper in a new context.*

7. References

Note: This list of references matches the reading list. Not all papers in this list are covered in the written questions. However, questions can be drawn from all papers during the oral examination.

[1] K. Sankaralingam, R. Nagarajan, H. Liu, J. Huh, C.K. Kim D. Burger, S.W. Keckler, and C.R. Moore. Exploiting ILP, TLP, and DLP Using Polymorphism in the TRIPS Architecture, *30th Annual International Symposium on Computer Architecture (ISCA)*, pp. 422-433, June 2003.

[2] Song Jiang and Xiaodong Zhang, LIRS: an efficient low inter-reference recency set replacement to improve buffer cache performance, *Proceedings of the 2002 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, (SIGMETRICS'02)* , Marina Del Rey, California, June 15-19, 2002.

[3] Xiaodong Li, Zhenmin Li, Pin Zhou, Yuanyuan Zhou, Sarita V. Adve, Sanjeev Kumar. Performance-Directed Energy Management for Storage Systems, *IEEE Micro*, vol. 24, no. 6, pp. 38-49, November/December, 2004.

[4] Rajagopalan Desikan, Charles R. Lefurgy, Stephen W. Keckler, and Doug Burger, n-chip MRAM as a High-Bandwidth, Low-Latency Replacement for DRAM Physical Memories *Technical Report TR-02-47*, Department of Computer Sciences, The University of Texas at Austin, September 27, 2002.

[5] Mark Oskin, Frederic T. Chong, Isaac L. Chuang, John Kubiatowicz, Building Quantum Wires: The Long and the Short of It. *30th Annual International Symposium on Computer Architecture (ISCA)*, pp. 374-385, June 2003.

[6] Waldspurger, C. A. 2002. Memory resource management in VMware ESX server. First published in OSDI '02: Proceedings of the 5th symposium on Operating systems design and implementation, *SIGOPS Oper. Syst. Rev. 36*, SI (Dec. 2002), pp. 181-194. DOI= http://doi.acm.org/10.1145/844128.844146.

[7] Bogdan Caprita, Wong Chun Chan, Jason Nieh, Clifford Stein, and Haoqiang Zheng. Group Ratio Round-Robin: O(1) Proportional Share Scheduling for Uniprocessor and Multiprocessor Systems. *Usenix '05: Proceedings of the Usenix 2005 Annual Technical Conference, General Track* (Apr. 2005), pp. 337-352, Anaheim, CA.

[8] Martin Rinard, Cristian Cadar, Daniel Dumitran, Daniel M. Roy, Tudor Leu, and William S. Beebee, Jr. Enhancing Server Availability and Security Through Failure-Oblivious Computing. *OSDI '04: Proceedings of the 6th symposium on Operating systems design and implementation*, (Dec 2004), San Francisco, pp. 303-316.

[9] Galen C. Hunt, James R. Larus et al. An Overview of the Singularity Project. *Microsoft Research MSR-TR-2005-135*.

[10] Pubrick et al. An Extensible Event-Based Infrastructure for Networked Virtual Worlds, *Presence*, 2003.

[11] Björn Grönwall, Assar Westerlund, Stephen. The Design of a Multicast-based Distributed File System. *Proceedings of the 3rd Symposium on Operating Systems Design and Implementation*, New Orleans, Louisiana, February, 1999.

[12] Antonio Carzaniga, David S. Rosenblum, Alexander L. Wolf. Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems*, vol 19, no. 3, pp. 332-383.