

Review Questions

Instructions: You are not required to turn in answers to these review questions, but you should do them on your own after you read this section's papers to test your knowledge. Use Piazza or the TAs if you have questions or get stuck on any of these.

LRVM paper

- 1) What are the disadvantages in using Camelot for Coda?
- 2) Justify inclusion and exclusion of the mechanisms in LRVM as compared to a full blown transactional system such as Camelot.
- 3) LRVM is implemented as a library to be linked with an application desiring its services. What are the implications of this design approach?
- 4) LRVM lives above the operating system. The VM subsystem and the LRVM subsystem are thus independent of each other. What implications does this have on functionality and performance? In particular, how does this decoupling show up for applications running on top of LRVM?
- 5) LRVM provides "bounded persistence". Explain.
- 6) The authors claim that structuring the LRVM as a user-level library as opposed to a collection of tasks communicating via IPC led to better scalability as evidenced by lower CPU load for LRVM compared to Camelot. While this may be true since Camelot is built on top of Mach, would you agree with this reasoning as ground truth?
[Hint: Recall some of the papers we read on OS structures.]
- 7) How would you justify that LRVM is lightweight?
- 8) Develop the data structures and pseudo code for performing incremental truncation.

Rio Vista paper

- 1) What are the sources of problems in computer systems that leads to failure? How are these relaxed in Rio Vista?
- 2) The Vista RVM library is 700 lines of code as opposed to 10K lines for a comparable functionality implemented in LRVM. Explain why.
- 3) Develop the data structures and pseudo code for the Vista library (begin transaction, set-range, end transaction).

Quicksilver

- 1) What is the origin of the Quicksilver name for the OS presented in this paper? :-)

- 2) Give examples with justifications of servers that fall into the "stateless", "volatile", and "recoverable" classes, respectively.
- 3) Distinguish between "owner" and "participant" of a transaction.
- 4) Identify points of similarity between LRVM and Quicksilver.
- 5) Quicksilver does not abort a transaction immediately upon detecting failure. Why?
- 6) Discuss with examples of situations that are appropriate for each type the four different commit protocols (one-phase immediate, one-phase standard, one-phase delayed, and two-phase).
- 7) Discuss how Quicksilver deals with participants that may join a transaction that is in the process of committing.
- 8) How does Quicksilver manage its log? What are the optimizations used to reduce latency for log writing and dealing with memory pressure of excessive log generation?

Transactions in TxOS operating system

- 1) Give user level and system level examples of where transactional semantics will help in increasing concurrency while ensuring atomicity and isolation for the actions.
- 2) Discuss the semantics of transactional support in TxOS relative to Quicksilver.
- 3) Explain the system level race condition (with an example) caused by "time of check to time of use" phenomenon in system code.