The University of Melbourne
COMP90015 Distributed Systems: Principles and Paradigms
Semester 2, 2016 Final Examination

Department of Computing and Information Systems
COMP90015 Distributed Systems: Principles and Paradigms
Reading Time: 15 minutes
Writing Time: 3 hours

Open Book Status: Closed Book

This paper has 3 pages including this page

Identical Examination Papers: none
Common Content: none

Authorized Materials:
No materials are authorized.

Instructions to invigilators:
No papers may be taken from the exam room.

Instructions to students:
All answers are to be written in the script book(s) provided.
Attempt all questions - partial credit is available.
The examination is worth 60% of the subject assessment.

Paper to be held by Baillieu Library: yes

continued on next page

Q.1. (a) [**5 marks**] Consider a large computer lab, e.g. like those found on a university campus. The computer lab allows users to login to any computer using their username and password and makes various software and other resources available that are specific to that user. List and briefly explain, with respect to this example, five challenges of distributed systems.

(b) [**5 marks**] Consider the provisioning of general purpose applications in the cloud, where a web browser is used to connect to and operate the application. For example, an instance of a spreadsheet application can run on the cloud and a user can connect to it via their web browser to work on spreadsheet data. What type of distributed system *architectural model* best describes this? Give a reason for your answer. Give three reasons that support the use of this architectural model. Give an example of an application for which the architectural model is not suitable.

Q.2. [**5 marks**] In your first project you implemented a multi-server system that broadcast various kinds of messages between the servers to provide the service. Answer the following questions related to your first project:

(a) In the project each subsequent server could connect to only one existing server at startup. Now consider allowing each subsequent server to connect to multiple existing servers at startup. What major problem would this cause, assuming the protocol does not change in any other way? What changes to the protocol/implementation can you suggest that could fix this major problem? Briefly explain your suggestion.

(b) Explain the failure model that was assumed in the first project with respect to all components and interactions.

Q.3. (a) [**3 marks**] Briefly explain three fundamental aspects of a communication chanel that are important from the point of view of a distributed system.

(b) [**2 marks**] What is meant by a *timing failure*? Briefly explain an example of a distributed system where a timing failure would have significant impact.

Q.4. (a) [**3 marks**] State three differences between UDP and TCP communication.

(b) [**2 marks**] What is a benefit of XML over JSON format? What is a benefit of JSON over XML format?

(c) [**5 marks**] Explain *maybe* semantics, *at-most-once* semantics and *at-least-once* semantics, in terms of a client sending a request to a server. What level of semantics does the RR protocol achieve? What level of semantics does Java RMI enforce?

Q.5. (a) [**5 marks**] Consider a "music popularity" application that runs on mobile devices. Each user's mobile device reports the name of the song that is currently playing on the device, to a central server. The central server ranks the songs in terms of popularity across all devices and makes the results known back to

continued on next page

all users, so that the most popular songs can be seen by everyone. Choose either *tuple space*, *message queue* or *publish/subscribe* paradigm and explain how the application could be implemented using your chosen paradigm. Draw an architectural diagram to aid your explanation.

(b) [**5 marks**] Consider implementing a *distributed shared memory* middleware. Describe what kind of protocol and messages that you would use, and why. Discuss the operating system support that your implementation would require.

Q.6. (a) [**4 marks**] Explain the difference between a worker pool multi-thread model and a thread-per-connection model, including details concerning what concurrency control and queueing is required in each case.

(b) [**1 marks**] Explain the difference between *virtualization* and *emulation.*

Q.7. (a) [**2 marks**] Some distributed systems, such as secure shell, require the user to explictly trust a public key provided by the server, when connecting for the first time. Name the security risk that this presents and explain why this is a security risk.

(b) [**3 marks**] Explain what is meant by *challenge-response* and give an example of its use.

Q.8. (a) [**5 marks**] Describe the caching policy used by NFS at the client. What parameters are used? What checks are done, in what order and why?

(b) [**2 marks**] Explain how DNS can be used to distribute the load of incoming requests over a set of servers.

(c) [**3 marks**] Consider an application where resource objects are stored across a multi-server system. As a requirement of the system, given a name of an object, it is critical that the object needs to be accessible as fast as possible. Describe a name space and name format that would be best to address this requirement and explain why. Is your name format an example of a pure name or not?

END OF EXAMINATION