TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

EXAMINATIONS — 2006

MID-YEAR

COMP 413

DISTRIBUTED SYSTEMS

Time Allowed: 3 Hours

Instructions: You must answer six questions out of seven possible questions.

Everyone must answer **QUESTION ONE**.

PLUS FIVE additional questions chosen from questions two to seven.

Each question is worth 25 marks.

Within each question, the marks for subparts are shown.

There are a maximum of 150 marks in total.

Non-programmable calculators without full alphabetic keys are permitted.

Paper foreign language to English dictionaries are permitted.

Question	Topic	Marks
1	Grid Computing	25 marks
2	Communication	25 marks
3	Threads and Mobile Code	25 marks
4	Synchronisation	25 marks
5	Fault Tolerance	25 marks
6	Replication Consistency	25 marks
7	Security	25 marks

Question 1. Grid Computing [25 marks]

Demonstrate the application of *one or two* distributed systems concepts to grid computing. Your discussion should include examples from *one or two* of the research articles read for the Friday seminar series.

We can't give an answer as it will depend on the concepts and articles chosen. This question will be given to the class before the exam. It will be marked by both John and Ian. The markers will be looking for a clear demonstration of understanding of the concept and its application to grid computing.

Question 2. Communication [25 marks]

(a) [10 marks] Consider remote procedure calls (RPC). Describe the major components and steps involved in a server registering itself with a name service and subsequently a client binding a local interface to the remote server.

Main components: server program, server stub, IPC runtime, name service, client and client stub.

Server program exports its interface by invoking server stub function. Server stub calls the IPC runtime. The IPC runtime associates the interface with the server program and contacts the name server to associate the interface with the host where the server resides and the port on which the IPC runtime is listening for connections. (4 marks)

Client program invokes the client stub to import or bind to the server interface. The client stub passes the interface name or id to the IPC runtime. The IPC runtime interrogates the name service for the ¡host, port¿ of the runtime system supporting the interface (there may be more than one). It then queries that runtime to obtain a final binding to the interface (maybe allocate a port just for that service) which can be used in remote invocations. The IPC runtime returns the result to the client stub. The client stub records the result and returns a local identifier, for use in future invocations, to the client program. (6 marks)

(b) [10 marks] Consider remote method invocations (RMI) versus local method invocations. Identify and briefly explain the difficulties that a developer of a RMI mechanism might face in providing transparency with respect to synchronization, exceptions and garbage collection.

Central to RMI is use of an interface that is present at both the client and server. Interfaces do not have constructors.

Synchronization controlling access to object state using monitors etc. is easier with local objects because shared memory but more complicated because communication via

messages, clients on different machines and fail independently of the server hosting the object.

Exceptions: programs can encounter errors and unexpected conditions, exceptions provide a clean way to structure error handling code. In each method heading, the method declares the exceptions that it might encounter. THROWS/CATCHES. Remote exceptions are added.

Garbage collection. In C++ you must explicitly allocate and deallocate memory, in Java and other languages this is automatically managed making programming much easier. In a distributed system you might know that a client is still referencing a remote object. It may have failed without informing you or you are not sure that it is slow or dead.

(c) [5 marks] Compare and contrast the approaches used by Java RMI and Sun RPC to hiding the differences between data representations used by different hardware architectures.

Java allows pass-by-value and pass-by-reference whereas Sun RPC only allows pass-by-value. Sun RPC's IDL is language-neutral whereas Java's RMI is language specific and includes a means for the developer to specify whether parameters are passed by value or by reference.

Sun RPC uses a standard representation for sending values across the network. Each architecture requires a mapping from the network format to its own local format. Whereas Java assumes that everyone is running a Java VM and this means that the VMs use a standard representation. Rather like the Sun RPC there must be a Java VM developed for each architecture. Which is more scalable? Could argue that both require special work done for each architecture. Could argue that its easier to write a routine to convert the network format to host format than re-engineering an entire VM for each architecture.

Question 3. Threads and Mobile Code [25 marks]

(a) [5 marks] Explain how multithreading can be used to conceal the latency of WANs.

Can contact multiple servers. Can update display immediately and wait for response

(b) [10 marks] Compare the design and performance of a single-threaded webserver with a multithreaded implementation.

Multithreaded webserver would typically have a single thread handling connection and multiple worker threads. Alternative organisation is a finite-state machine that has parallelism and nonblocking system calls (2 marks). Record state of current request in table, issue non-blocking IO, get next message (3 marks). Message might be satisfaction of IO request or message over the network (2 mark). No context switch cost. Might be better performance if IO bound rather than CPU bound. Removes the need for locking but probably not a problem for the webserver. (3 marks)

(c) [10 marks] What challenges must be overcome when implementing code mobility in heterogeneous systems? Briefly explain some possible solutions for implementing weak and strong mobility.

Heterogeneous systems may be composed of hosts with different operating systems and architectures. The conventions for the layout of object code, data and activation records may vary. In addition libraries used by the code may not be present or may be different from host to host (3 marks). Implementing weak mobility is relatively straightforward because only the code and not the state of the program is required to be moved. Therefore, a different version of the code for each operating system could be written and libraries could be migrated with the code (3 marks). Implementing strong mobility is more difficult because of the requirement to halt execution and transfer the state of the program with the code. The usual approaches to this is to limit to points at which transfer may take place. Common approaches are to insert checkpoints in the code or only migrate when procedure calls take place and the stack is available (4 marks).

Question 4. Synchronisation [25 marks]

This question considers two approaches to synchronisation: the use of distributed algorithms and transactions.

(a) [6 marks] There are three main algorithms for distributed mutual exclusion: centralised; distributed and token ring. Briefly compare their vulnerability to failure.

> *Centralised: single point of failure, coordinator crash. Distributed: crash of any process can halt the algorithm. Ring: loss of a token or process crash can halt the algorithm. Centralised actually more fault-tolerant than either distributed or ring as long as the coordinator stays available.*

(b) [6 marks] Consider a ring-based election algorithm. Calculate how many messages are circulated in the worst case for a single election. Make sure that you explain the reasoning behind your calculation.

> *Main points: identification of worst case (2 marks), calculation and explanation of the case (4 marks).*
>
> *In the worst-performing case, anti-clockwise neighbor has the highest identifier. n-1 messages required to reach the neightbour, neighbour announces its election making n more messages. The elected message is sent N times. This makes 3N-1.*

(c) Consider the following interleaving of two transactions T and U.

Transaction T	*Transaction U*
`w = read(i)`	
`write(i,w+10)`	
`x = read(j)`	
	`y = read(j)`
	`write(j,y+30)`
`write(j,x+20)`	
	`z = read(i)`

(i) [4 marks] Explain whether the interleaving shown above is *serially equivalent* or not.

> *Ordering is not serially equivalent. The pairs of conflicting operations (writes and reads to i and j) are not done in the same order at both objects. OR some argument showing that result is not the same as U following T or T following U.*

(ii) [6 marks] Explain how a *one writer-many readers* locking scheme could be applied to enforce serial equivalence and prevent the interleaving described above.

> *Pessimistic scheme is the use of locks. Could use per-object locks. Only one writer and multiple readers. T gets write lock on i. U tries also but read conflicts with write so delayed. T runs to completion. T releases all locks and U now runs to completion.*

(iii) [3 marks] Briefly discuss the principal disadvantages of using locking to enforce serial equivalence.

Two main disadvantages. Possiblity of deadlock and lost efficiency – might be overally pessimistic.

Question 5. Fault Tolerance [25 marks]

(a) [3 marks] What is a fault tolerant system?

A fault tolerant system continues to operate in the presence of faults. Either faults do not result in errors or errors are recognised and managed.

(b) [10 marks] Define the five characteristics of a *dependable system*..

define availability, reliability, safety, maintainability, security – maybe this should be worth more?

(c) Reliable multicast.

(i) [5 marks] Discuss the role of reliable multicast in achieving fault tolerant systems. Your answer should explain its major advantages; justify different orderings; and identify major implementation problems.

When we use groups of replicated processes they have a requirement to communicate with each other. Reliable multicast is a common, application independent component. It simplifies the implementation of cooperating groups.

The different orderings recognise different requirements. In some applications FIFO ordering may be sufficient while others may require causal or total. Answer should indicate difference and examples of applications would be excellent.

Problems include scaling (acks are not multicast) which can be addressed using negative acknowledgements, and group membership.

(ii) [3 marks] Differentiate *FIFO* from *causal* ordering in group multicast.

In FIFO ordering all messages from source A are delivered in the order sent at all other members of the group. Messages from different sources may be interleaved in different ways at different members.

Causal ordering places greater constraints on message delivery. Where the arrival of one message may have influenced the contents of a subsequent message that is sent those messages should always be delivered in the same order at all members of the group.

(d) [4 marks] Explain the role of *virtual synchrony* in group communication.

Virtual synchrony is a way of handling changes to group membership, either intentional or failures. Whenever a change occurs or is noticed a member of the group sends out a "view change" message which acts as a synchronisation barrier. All existing messages are delivered (or discarded) and the view (membership) is changed. Message delivery resumes.

Question 6. Replication Consistency [25 marks]

(a) [5 marks] What are the principal advantages and disadvantages of a replicated data store (RDS)?

Advantages: availability, improved performance through scalability and location. Disadvantages: must manage concurrent invocations, maintain consistency.

(b) [5 marks] Define *sequential consistency* in the context of a RDS.

If a RDS provides sequential consistency then all processes/clients using the RDS see the same sequence of operations. The sequence may or may not be what actually happened. The sequence must be feasible under strict consistency.

(c) [10 marks] Most consistency models other than sequential have a goal of providing semantics that are, for some context, equivalent to sequential and are less costly to implement.

(i) For a RDS providing FIFO consistency explain how it *is and is not* equivalent to sequential consistency.

(ii) Explain why FIFO consistency is less costly to implement.

(iii) Give an example of an application that would work correctly with FIFO consistency.

FIFO consistency maintains the order of the writes done by individual procesess, but may interleave them differently when they are observed by different processes. In particular FIFO does not attempt to maintain possible causal relationships amongst writes. When compared with sequential it only maintains the write operations of individual processes. Other operations may appear in different orders at different processes, but if these operations are not on shared data the correct execution of the process is not affected.

FIFO is less costly to implement as it only requires that writes from each process are tagged with that processes pid and a sequence number. Sequential consistency requires some form of global ordering.

FIFO consistency works in situations with largely independent processes that seldom interact. A distributed file system would meet this criteria and the few processes that interact can use locks external to the RDS for synchronisation.

(d) [5 marks] Explain the operation and the benefits of quorum based protocols in a RDS.

Quorum based protocols are designed to allow operations on a RDS to proceed when all of the replicas have not been updated. Each variable is assigned a version id. The idea is that to do a write operation you must be able to assemble a write quorum of more than

half of the replicas and each replica must be up to date. The write is then performed on this set and later propagated to others in the background.

To perform a read you must assemble a read quorum which is a set of replicas such that RQ+WQ¿n. This ensures that at least one member of RQ is up to date. Querying each for the most recent value produces the desired result.

Question 7. Security [25 marks]

(a) [8 marks] Explain how home forwarding is used to establish connections to mobile entities.

Each mobile entity has a home and a home agent that represents it in its absence. The router usually performs this for mobileIP.

At other locations it has a foreign agent that allocates a "care of" address for use during its stay. When the mobile entity changes location (address) it most notify its home agent of its new address and the foreign agent's address.

When a packet arrives for an absent mobile entity the home agent (1) responds to the request giving the new address and (2) tunnels the request to the foreign agent which in turn forwards it to the mobile entity. The mobile entity can then respond directly to the request. The connection is established without forwarding through the home agent.

(b) [8 marks] Identify *two new security risks* introduced by the use of home forwarding and give a brief explanation of how you would reduce the risk.

There are several new activities that are possible sources of risk.

1. *The mobile entity gives its location to its home agent. We must guard against an intruder providing the address of an interloper. A standard authentication protocol using either a shared key or pki can be used to address this.*
2. *The home agent gives the care of address to the source of a packet/connection request. We must know that the home agent is working on behalf of the mobile entity. For new connections this could be handled by authentication at the application level after the connection is established. For existing connections authentication of the home agent could be used if the mobile entity shared a secret with both the sender and the home agent.*
3. *Sender addresses are forwarded by the foreign agent. These could be protected if the home agent signed the message.*

(c) [9 marks] Several of the security paradigms that we have looked at are capability based in the sense that the client holds credentials that are used to authorise operations. Give three security issues that arise from the fact that the credential resides with the holder and in one sentence for each state how the issue is addressed.

Expect three of the following.

- *The credential may be modified. Addressed by digital signature.*
- *The credential may be held indefinitely. Addressed by revocation mechanism.*
- *The credential may be captured by a third party. Addressed by imbedding the id of the original older in the signed credential and requiring authentication. item The credential may be fabricated. Addressed by including a secret held by the issuer in the signature.*

****