THE BCS PROFESSIONAL EXAMINATIONS
BCS Level 5 Professional Graduate Diploma in IT

April 2007

EXAMINERS' REPORT

DISTRIBUTED & PARALLEL SYSTEMS

General comments

Despite the increasingly massive world-wide dependence on computing systems, fewer candidates seem prepared to tackle this paper. However it is a really demanding exercise to get oneself prepared to range confidently across the range and depth of material covered.

Question 1

a) Explain how may data be ***marshalled*** prior to transmission in a message

(5 marks)

b) What are the requirements for communication and synchronisation between cooperating processes in a distributed system?

(6 marks)

c) Discuss the protocols required to support
(i) client-server and
(ii) group communication.

(14 marks)

Answer Pointers

a) Marshalling is to do with the assembling of data items and consists of their translation into an agreed standard external representation by a middleware layer of software. CORBA is one common data representation, whilst Java uses reflection to serialise objects. [Coulouris, Dollimore & Kindberg (2), p103-104; CDK(3), p138-145]

b) Interprocess communication requirements are to map data structures and data items to messages, adopting appropriate standards, and marshalling them prior to send/receive operations. Synchronous and asynchronous communication requires to block or non-block the 'send' process, which specifies one of a range of potential location independent destinations, in order to to handle the possibility of queues forming. The degree of reliability and style of communication is expected to vary according to the application required but is built upon acknowledgment protocols at the cost of added overhead. [CDK(2), p101-107; CDK(3), p127-129]

c) Client-server communication is normally via a synchronous request-reply protocol so that two send and two receive operations are involved, each requiring a system call. Examples that reduce overheads using only three system calls exist, whilst delivery failures often use timeouts dependent upon the type of RPC exchange protocol in use. Note that the protocol may require to recognise a re-transmitted message following a timeout unless operations are idempotent. The use of a history reduces retransmission requirement but comes with a storage penalty. [CDK(2), p108-112; CDK(3), p145-153]
Group communication using an atomic multicast message is received by all members of a process group such that in multicast protocols the use of a hardware multicast reduces the number of messages and the time taken by the originator to transmit the message. [CDK(2), p112-119; CDK(3), p153-158].

Examiner's gGuidance Notes

Question 1 was a popular question and reasonably well done with roughly half those answering this largely bookwork material showing good preparation.

Question 2

a) Names are used in a distributed system to refer to a variety of resources. Identify at least three different types of name encountered within distributed systems and give examples.

(6 marks)

b) Outline the main requirements for a name service.

(6 marks)

c) Compare those issues appropriate to a typical name service that might be designed for a large organisation, (such as a university or technical college) and those issues appropriate to a name service designed for a worldwide distributed system (such as DNS or GNS).

(13 marks)

Answer Pointers

a) Names are needed to request a computer system to act upon a given resource chosen from many; to enable the sharing of processes and to enable users identify themselves. These are for instance names supplied in the form of an address (physical network addresses and logical internetwork addresses), of an identifier (port, process or group identifiers), of text (human-readable names) or of files (usually using human readable text).
For example a textual file name might look like: /users/smith/prog.c
a port or service-specific id might look like: 164997-9
a network address might look like: 2:60:8c:2:b0:5a
[CDK(2), p254-256; CDK(3), p355 for an example accessing a resource from a URL].

b) The main requirements for a name service are: an ability to handle an arbitrary number of names, for the service to have a loing lifetime, high availability, the isolation of faults and the tolerance of mistrust. [CDK(2), p283, CDK(3), p382]

c) The basic design issues are the structure of the name space – the syntatctic rules governing names. Most divide name space into domains – discrete sections each with a single authority controlling the binding of names therein. The set of bound names must be managed such that for a typically large organisation this would mean naming users, services, computers and groups of these. The service name space typically would be split into directories nested arbitrarily deeply and organisationally partitioned to correspond to the departments within the organisation. For larger implementations the naming database may be stored at multiple name servers, each of which stores at least part of the set of names within a naming domain. Types of navigation that are supported are: iterative, multicast, and (non-)recursive server-controlled. Replication and caching assist in delivering high availability. Further issues and illustrations may be drawn from DNS, (for naming computers and addressing across the Internet) or the Global Name Service (which tackles the issue of reconfiguring the name space as organisational changes occur). [CDK(2) p257-278, CDK(3) p357-371,374-377]

Examiner's Guidance Notes

Question 2 was a question that cut across the syllabus rather than down into the syllabus and in consequence put off those unwilling to 'think on their feet'. Only two of the few candidates who attempted this question were well rewarded.

Question 3

a) What do you understand by the term *scalability* in respect to the performance of a parallel computer system.

(4 marks)

b) Parallel computing can be achieved in a number of ways, either by using specially designed parallel computers or by using a specially configured arrangement of general purpose computers.

Describe ***one*** example of a specially designed parallel computer and ***one*** example of a specially configured arrangement of general purpose computers.

(14 marks)

c) Indicate the environment and problem types best suited for the examples you have chosen in (b) above and comment on their scalability.

(7 marks)

Answer Pointers

a) The term scalability is rather imprecise but is used to indicate how system design allows performance to increase in proportion to increase in size (of the hardware system as 'architectural' scalability or of data handling as 'algorithmic' scalablility). [Wilkinson, p11]

b) There are several choices in each category that can be described. For specially designed machines, either a shared memory multiprocessor system or a specifically designed network (mesh, hypercube, etc.) of multiprocessors (using message-passing multicomputers or distributed shared memory) may be described. For specially configured arrangements of gp computers, interconnected computers as a cluster or grid of computers may be described. There are valid variants in interpretation that are possible and, if properly argued, will be accepted. [Wilkinson, p13-38]

c) Answers will vary, dependent upon the examples chosen in (b), but in general specially designed systems gear toward specific problem types. It is also worth noting that multicomputer configurations (that is message-passing multicomputers arranged in a special network or as a cluster) will physically scale more easily than a shared memory multiprocessor systems. [Wilkinson, p13-38]

Examiner's Guidance Notes

Question 3 was a good discriminator between those who were prepared to think as opposed to those who prefer to recall.

Question 4

a) Discuss how different parallel programming strategies may be evaluated.

(4 marks)

b) Assume that, as part of a complex calculation, a sequence of numbers is to be added. Without providing code level detail, briefly describe each of the following three strategies:

data-partitioning
recursive divide-and-conquer
pipelining.

(15 marks)

c) Compare these three strategies, using an appropriate evaluation.

(6 marks)

Answer Pointers

a) Time complexity analysis (yielding computation, communication and execution times) may give an insight, but only when an algorithm is coded and executed can the elapsed time be known. It is through elapsed time that comparison between strategies be made. Often message-passing software includes facilities for timing. Note that 'profiling' a program allows for the identification of those source statement repetitions that may become a consequent target for optimisation. [Wilkinson, p72-74]

Data partitioning involves dividing the sequence into p parts of n/p numbers each at which p processors (or processes) can each add one sequence independently to create partial sums. The p partial sums need to be added together to form the final sum.
[Wilkinson, p106-111]

Divide and conquer involves dividing the sequence into two parts such that each part recursively calls to add sublists in such a manner until each sublist comprises only one number, (thus constructing a binary tree). An efficient parallel implementation reuses processors at each level of the binary tree such that there are log p processors (processes) required. [Wilkinson, p111-116]

Pipelining involves a series of separate processor (process) stages in which each stage adds an element of the sequence to an accumulating sum. If the pipeline is connected as in a ring structure, then the resultant sum emerging from the final stage can be returned. Note however that the sequence would be data-partioned initially in order to reduce the stages required. [Wilkinson, p140-148]

Bearing in mind that elapsed time involves both computation and communication time, the computation time speed up (of parallel implementation over sequential implementation) for data partitioning tends toward p for large n. Likewise, for divide-and-conquer the very best computation speed up is p (when all p processors are computing their partial sums). Pipelining a single instance of the addition is unrewarding and requires n pipeline cycles. However for m instances of the addition (where m is large) the trend is toward execution in a single cycle. [Wilkinson, p72, p107-116 and p141-148]

Examiner's Guidance Notes

Question 4 was poorly attempted with the notable exception of the one candidate who gave a perfect answer.

Question 5

You have agreed to talk for 30 minutes at the next meeting of your local BCS branch. The title of your talk is

Comparing Distributed and Parallel Systems: Is there a future for parallel systems?

Sketch out approximately eight presentation slides, with associated notes, that you would use for your talk.

(Your answer will be assessed for its quality of approach, accuracy of content, clarity of expression, range of discussion, and depth of argument (5 marks each)).

Answer Pointers

This type of question makes a regular appearance at the end of the question paper each year. This year, the topic requires candidates to argue a case depending upon their experience and knowledge. Reference to Global Grid computing is expected.

The number of slides indicate to the candidate that they should spend approximately 5 minutes on each slide.

Examiner's Guidance Notes

In ***question 5*** candidates generally gathered reasonably marks. Those who took an integrated approach (rather than discuss the distributed and parallel systems separately) and drew conclusions were welcome. Diagrams were often lacking.