Distributed Systems: Concepts and Design

Chapter 19 Solutions to Exercises

19.1 What is a volatile system? List the main types of changes that occur in a ubiquitous system

19.1 Ans.

A volatile system is one in which certain types of change are the rule rather than the exception. In a ubiquitous system, the set of devices and software components can change regularly, as well as the set of users. Devices may fail, e.g. due to battery failure. Users may bring devices in and take them out again. Changes in the set of associations may result. The state of connectivity between devices may also change, due to the way ubiquitous systems are integrated with physical objects that may affect wireless transmission.

19.2 Discuss whether it is possible to improve upon the 'pull' model of service discovery by multicasting (or broadcasting) and caching replies to queries.

19.2 Ans.

We start with the 'pull' model described in the chapter but turn it into a hybrid with the 'push' model by having services multicast/broadcast their existence when a query occurs. All potential client devices can then hear and cache the service advertisements. When a component on the device requires a service, it first consults the local cache. If no matching services are found or if a more up-to-date list of matching services is required, then the device still multicasts its query. However, now it includes information about any matching services in its cache. Other potential clients may now learn about those services' existence. Moreover, only services not in the query need reply.

While it is intuitively reasonable, this hybrid scheme needs validation through simultaion or experimentation against the required implementation parameters. The extra multicasts incur a cost.

19.3 Explain the use of leases in a discovery service to cope with the problem of service volatility.

19.3 Ans.

The problem is that services may fail or become disconnected at any time, leaving stale entries in the service directory.

Assume a lease expiry time of t seconds. The list of services in a directory is fresh to within t seconds because any server that is still alive will renew its lease before its expiry time. If a server leaves, crashes, or becomes disconnected from a lookup service, the latter will delete its entry as soon as the lease expires. Therefore an unavailable server is registered only for a maximum of t seconds after it becomes unavailable. A client that, nonetheless, obtains a stale server entry, will discover this when it tries to connect.

19.4 The Jini lookup service matches service offers to client requests based on attributes or on Java typing. Explain with examples the difference between these two methods of matching. What is the advantage of allowing both sorts of matching?

19.4 Ans.

Attributes - attributes describe properties of a service, for example a printer might specify its speed, resolution, whether it prints on one of two sides of the paper, its location in the building.

Types - specify the Java data types implemented by the service. e.g. Printer, ColourPrinter.

Both sorts of matching. e.g. if match only by type (e.g. for ColourPrinter), there may be several such services and the client can specify its selection by means of attributes e.g. to get the nearest one. If match only by attributes, the type may not be exactly correct.

19.5 Describe the use of IP multicast and group names in the Jini 'discovery' service which allows clients and servers to locate lookup servers.

19.5 Ans.

Each lookup service has a set of groups associated with it. A client or server needing to locate a lookup server specifies the names of a set of groups (they are interested in) in a multicast request.

Lookup servers belonging to those groups reply by sending a proxy. Lookup servers also advertise the groups they are associated with when they send their "I am here" messages.

19.6 What is data-oriented programming and how does it differ from object-oriented programming?

19.6 Ans.

In a data-oriented programming system such as the Web, the methods in the interface between components are always the same. What is allowed to vary is the data that is passed through those methods and the semantics of the processing that is applied to that data. In object-oriented programming, by contrast, each object can have an interface with arbitrary methods. Client are usually (although not necessarily) programmed on the assumption of specific processing semantics associated with those methods, complicating interoperation with independently written objects.

19.7 Discuss the issue of how the scope of an event system can and should be related to the physical extent of a smart space in which it is used.

19.7 Ans.

This is a 'boundary principle' question. Components in a hotel guest room probably don't need to know (and shouldn't know) about events occurring in the guest room next door. However, certain hotel management components hosted elsewhere in the building may have an interest.

One approach would be for events to be propagated by default only to components within the smart space. If necessary, those events could also be relayed to components outside the smart space by a proxy component in the smart space.

But how to ensure that propagation is physically limited in scope? The techniques introduced for physically scoping asociation (Section 16.2.2) can all be applied to this case, too.

19.8 Compare and contrast the persistence requirements associated with event systems and tuple spaces in the infrastructure of smart spaces.

19.8 Ans.

Events are intended to be delivered immediately to consumers and, by default, they do not persist. A smart space implementation could, however, choose to record all the events that occur for debugging or replaying applications -- in which case a separate event store would be required, which subscribes to and consumes all events.

Tuples in tuple spaces have lifetimes that are independent of the lifetimes of the components that place them them there. That is a benefite in itself, given changes in the set of participating components. However, it could turn out to be a problem, since some tuples might never be taken from the tuple space even though they have no further value. For that reason, a tuple space implementation might delete tuples after a suitable period of time.

19.9 Describe three ways of sensing the presence of a user beside a display and thus motivate some features required in an architecture for context-aware systems.

19.9 Ans.

E.g. active badge sensor, face recognition by camera, voice detection.

An architecture for context-aware computing should (a) be able to accommodate such heterogeneous sensors; (b) abstract values such as "user present" or "user X is present" from the data they provide; (c) deal with the uncertainties and errors in the technologies, e.g. by fusing the results from several types of sensor; (d) detect changes e.g. from absence to presence.

19.10 Explain and motivate in-network processing for wireless sensor networks

19.10 Ans.

In-network processing refers to processing performed at the nodes of the network, as opposed to processing performed outside the network on data gathered from the network. The nodes of a sensor network combine sensing, processing and communication. By processing data sensed from the node or routed through it, we can prevent unnecessary communication – which is a significant drain on power. For example, duplicated data can be surpressed; sensed values of no interest do not need to be forwarded.

19.11 In the BAT location system, only three ultrasound receivers are used by default to obtain a 3D position whereas four satellites are required to obtain a 3D position in satellite navigation. Why is there a difference?

19.11 Ans.

As a matter of geometry, only three non-colinear reference points are required (given that we can eliminate solutions on the 'wrong' side of the plane formed by the reference points, e.g. eliminating solutions above a ceiling). But that presupposes that all distances are known.

In the BAT system, all times are computed by the difference between electronic and ultrasound signals and distances are measured using an estimate of the speed of sound. If we used a fourth reference point, then in principle we could solve for the actual speed of sound and thus determine position more accurately.

In the case of a satellite navigation system such as GPS, however, the receiving unit does not have an accurate way of measuring the signal propagation times from satellite to receiver directly. In this case, a fourth satellite's data is used to solve for the unit's clock offset from the satellites; it then becomes possible to deduce the propagation times from timestamps inserted by the satellites. The unit then estimates distances using an estimated speed of electromagnetic propagation (taking some account of atmospheric conditions)

If an object is at a known altitude (e.g. sea level) then only three satellites are needed.

19.12 In some location systems, tracked objects gives up their identifiers to the infrastructure. Explain how this may give rise to concerns about privacy, even if the identifiers are anonymous.

19.12 Ans.

Consider that a user carries a tracked object (for example, a wearable or handheld device) everywhere. The places that track the user could collude. If the identifier is directly correlated with the identity of the individual, then the user's movements become known, which they might not want. Even if the identifier is anonymous, the more places that identifier is observed, the greater is the chance that it can be correlated with identifying information. For example, if the user pays or signs for something and they are the only visitor in that place at that time, then their identity becomes known and thus their movements. Moreover, in principle, movements such as regular journeys (as between home and work) might also be used to make inferences. How many employees of Acme Widgets Ltd live near enough to the Zenith Coffee bar to visit it every working morning?

19.13 Each of many sensor nodes are to be scattered throughout a region. The nodes are to communicate securely. Explain the problem of key distribution and outline a probabilistic strategy for distributing keys.

19.13 Ans.

The nodes may have sufficient computational power to perform only secret-key encryption. If the nodes all share the same key then an attacker could compromise the whole collection by tampering with one node. If the number of nodes is large, there may be insufficient memory to store a pairwise key for each other node. A

compromise position is for nodes to share keys only with their nearest neighbours, and to rely on chains of mutually trusting nodes that encrypt messages hop-by-hop, rather than using end-to-end encryption.

But which keys should each node possess, if it is not known in advance which will be the node's neighbours? One approach is to distribute keys probabilistically. Each node gets a random set of keys drawn from a distribution which makes the probability of it sharing a key withsufficiently many neighbours high enough so that a fully connected, secure netwrok can probably be constructed. Nodes can test for keys in common by exchanging messages with encrypted intelligible text as a challenge.

19.14 We described several technologies that provide receive-constrained channels for use in secure spontaneous device association. Which of those technologies also provide send-constrained channels?

19.14 Ans.

Physical contact. Clearly this also provides send-constrained channels.

Infrared. By default, this provides us with a channel that is send-constrained only up to the boundary of a room, say, since an infrared receiver on, say, a PDA or mobile phone could receive data from many possible sources. However, it is possible to mount a receiver behind a lens to prevent signals from sources outside a required direction.

Audio. This provides us with a channel that is send-constrained only up to the boundary of an insulated room, say, since received audio could emanate from many possible sources.

Laser. By default, this provides us with a channel that is send-constrained only up to the nearest optical barrier. However, by mounting a receiver at the end of a narrow, opaque, non-reflective tube one could limit sources to lie in a given direction.

Barcode and camera. This combination can provide useful send-constrained channels. A camera can be focused and positioned so as to read data only from symbols presented from a given orientation and distance. For example, a user can place her camera phone so as to be sure of reading just the symbol presented by a given device.

19.15 Show how to construct a send-constrained channel from a receive-constrained channel, and vice versa. Hint: use a trusted node connected to the given channel.

19.15 Ans.

Let rc be a receive-constrained channel. We shall show how to construct a channel s(rc), which is send-constrained.

We use a trusted node N. When it receives a message m, it uses the receive-constrained channel to return to the sender a signed hash sig{h(m, t)}, where t is the time by N's clock.

Let c be any (possibly unconstrained) channel connecting the parties that we wish to be able to communicate. We construct s(rc) from c and N. The rules for sending and receiving on s(rc) are as follows:

To send m on s(rc):

send m to N

receive < t, sig{h(m, t)}> from N over rc

send <m, t, sig{h(m, t)}> on c.

To receive m on s(rc):

receive <m, t, h> on c

verify h = sig{h(m, t)}

verify currency of t and freshness of h

Discard m if verification fails, else receive m.

We assume that the receivers' clocks are synchronised to N's clock. The timestamp (and hence state of the sender) is deemed current if it is within a given bound of the time on the receiver's clock. To prevent replay attacks, the receiver need remember the hashes for only a limited time: older messages will have a non-current timestamp.

In a similar fashion, we can implement a receive-constrained channel from a send-constrained channel. All messages are sent (over any channel) to a trusted node, which stores them. Receivers must use a particular send-constrained channel to reach that node, which responds with the next message for them.

19.16 A group of smart spaces are connected only by a space between them such as a hallway or square. Discuss the factors that determine whether that intervening space can act as a mix zone

19.16 Ans.

A mix zone is an area in which users do not access location-aware services (i.e., do not give up an identifier correlated to a location). The idea is that users change their (pseudonymous) identifiers in the mix zone, and the mix zone acts to prevent correlations between the identifiers used by a particular individual.

To serve as a mix zone, there needs to be sufficiently many travelling through it at any given time. Moreover, it should be impossible to make an inference based on distances and likely speeds of travel. If, for example, two spaces are known to be close to one another, then an absence of an identifier in one correlated with a new presence in another shortly after may provide evidence that those two identifiers belong to the same person.

19.17 Explain the contextual factors to be taken into account when adapting multimedia content.

19.17 Ans.

The bandwidth to the device (the content may need to be compressed); the preferences of the user (e.g. do they prefer audio to visual data?); the capabilities of the user's device (e.g. do they have audio ourput? What is the screen size?); the needs implied by the user's current task (e.g. what type of map do they need?). Etc.

19.18 Assume that a device can execute 3 million instructions for the same amount of energy (3J) used to transmit or receive 1 Kbit of data 100m by radio. The device has the option of sending a 100KByte binary program to a compute server 100m away, which when run will execute 60 billion instructions and exchange 10000 1Kbit messages with the device. If energy is the only consideration, should the device offload the computation or execute it itself? Assume computation on the device is negligible in the offloaded case.

19.18 Ans.

Cost of executing the program on the device = 3J * (60 billion/3 million) = 60,000J

Cost of executing the program remotely = 3J * (800 /* program */ + 10000 /* messages */) = 32,400J

Device should offload the computation.

19.19 A Cooltown user squirts the URL of a sound file or a streaming radio station to an Internet radio. Suggest a modification to the eSquirt protocol that would enable the user to control the volume from their portable device. Hint: consider what extra data the squirting device should provide.

19.19 Ans.

One approach would be to provide a transport address along with the URL of the content to be rendered. The recipient device could then make a callback to the sending device, supplying a URL for obtaining the recipient device's control web page.

19.20 Discuss the applicability to mobile and ubiquitous systems of techniques drawn from the areas of (a) peer-to-peer systems (Chapter 10); (b) coordination and agreement protocols (Chapter 15) and (c) replication (Chapter 18).

19.20 Ans.

This is mainly a question for class discussion but some key observations are:

(a) peer-to-peer systems are volatile but they are not necessarily physically integrated, e.g. devices involved might be conventional PCs and typically there is no need to apply the boundary principle. Some of the p2p algorithms assume there is a connected infrastructure.

(b) the coordination and agreement protocols are in general designed on the assumption that failure is the exception (e.g. election algorithms) whereas in volatile systems it is the rule. Many of these techniques are too

expensive in communication or too slow to converge to be run frequently. Notions such as consensus are too strong to implement and have to be replaced by more pragmatic, weaker abstractions such as soft state. However, multicast provides a good level of indirection as long as relatively cheap-to-implement semantics suffice.

(c) Chapter 15 deals with disconnected operation. However, in general the techniques considered in this chapter tend to assume a redundancy of resources whereas in ubiquitous and mobile systems the opposite is generally true.
