Advanced Computer Systems

Theory Assignment 3

University of Copenhagen

Julian Sonne Westh Wulff <bkx655@alumni.ku.dk>
Toke Emil Heldbo Reines <bkx591@alumni.ku.dk>

January 6, 2022

1 Reliability

1.1 What is the probability that the daisy chain network is connecting all the buildings?

The daisy chain is fragile, in that any disconnected link will result in an incomplete network, as any node only has 1 connection - it might be fair to assume some tree structure in which any node/building is only connected to one other node/building (parent), giving us $n-1$ total number of links. This means the the probability of all buildings being connected is $(1-p)^{n-1}$, and the probability of the opposite is $1-(1-p^{n-1})$.

1.2 What is the probability that the fully connected network is connecting all the buildings?

The fully connected network has a link between all buildings, it is a complete graph. A complete graph has $E = (n*(n-1))/2$ number of edges, links in this case. For any building to be disconnected, we need all links to a given building to fail. Any given building has $n-1$ links, and all links to a building needs to malfunction/disconnect for the network to be incomplete. For any node, all edges must fail, so the probability of a `Fully connected` network has all buildings connected is: $(1-p)^{(E)^{n-1}}$.

In the given example of 3 buildings and 3 links, the probability of 1 link disconnecting in `Fully connected` is $1-(1-p)^3$ so the probability of 2 links failing, which is what will result in a disconnected network is $1-(1-p)^{3^2}$.

In reality, our assumption above is incorrect, as the problem is more complex. The network is disconnected, if there exists a pair of buildings between which no path connects them. The network could be split into 2 sub-networks, unable to communicate with each other, but yet there would be no buildings completely isolated from the network. So the actual solution would be to calculate the probability of the existence of any cut in the network (graph) such that the network would in fact be 2 sub-networks/graphs. Also, no link can fail more than once.

1.3 The town council has a limited budget, with which it can buy either a daisy chain network with two high reliability links (p = .000001), or a fully connected network with three low-reliability links (p = .0001). Which should they purchase?

To save money, they should accept the cheapest offer, just like KU did with the Niels Bohr building. If the cost is equal between a `Daisy chain` and a `Fully connected` network, they should choose the most reliable network. By comparison `Daisy chain` the daisy chain has a probability of success of $1-10^{-12}$ and the `Fully connected` $1-10^{-24}$. Thus the `Fully connected` network is the most reliable and should be chosen. Though other factors such as scalability and maintainability might also be of concern and might therefore also be considered in this regard.

Event	Vector Clock At Process Before The Event	Vector Clock in the Message (either received or sent)	Action Taken by Process	Vector Clock at Process After the Event
A	$\{0,0,0\}$	$\{1,0,0\}$	A2	$\{1,0,0\}$
B	$\{0,0,0\}$	$\{1,0,0\}$	A2	$\{1,0,0\}$
C	$\{0,0,0\}$	$\{1,0,0\}$	A2	$\{1,0,0\}$
D	$\{1,0,0\}$	$\{1,1,0\}$	A2	$\{1,1,0\}$
E	$\{1,0,0\}$	$\{1,1,0\}$	A2	$\{1,1,0\}$
F	$\{1,1,0\}$	$\{1,1,1\}$	A2	$\{1,1,1\}$
G	$\{1,0,0\}$	$\{1,1,1\}$	A2	$\{1,1,1\}$
H	$\{1,1,1\}$	$\{1,1,0\}$	A1	$\{1,1,1\}$
I	$\{1,1,0\}$	$\{1,2,0\}$	A2	$\{1,2,0\}$
J	$\{1,1,1\}$	$\{1,2,0\}$	A3	$\{1,2,1\}$
K	$\{1,1,1\}$	$\{1,2,0\}$	A3	$\{1,2,1\}$
L	$\{1,2,0\}$	$\{1,1,1\}$	A3	$\{1,2,1\}$

Table 1: Table of message exchanges among processes

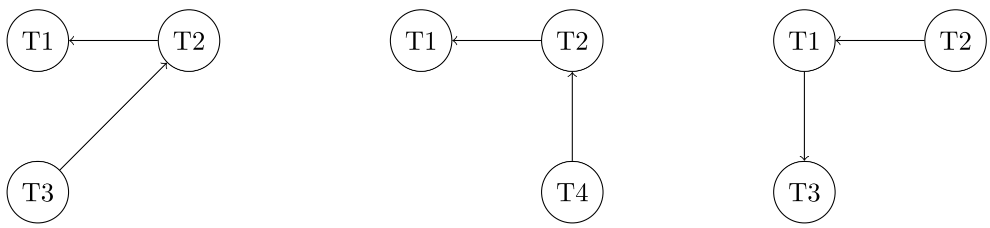
2 Vector Clock

All `A2` actions are trivial as all entries in the incoming vector contains a value equal to or larger than the process' current entries. In event `H` the first action `A1` is used, because the incoming vector clock was outdated, and therefore the incoming object was outdated. In the last three events `J, K, L`, action `A3` was chosen, as the incoming vector clock contained both outdated entries and updated entries (lower and larger values).

3 Distributed Transactions

3.1 Construct the local waits-for graphs on each node. Is there a local deadlock on any of the nodes?

By observing the the waits-for graphs in Figure 1 we see that all of them is acyclic and thus all of them has no local deadlocks.



(a) Node 1 (b) Node 2 (c) Node 3

Figure 1: Local waits-for graphs for Node 1 – 3

3.2 Construct a global waits-for graph. Is there a global deadlock?

By observing the the waits-for graphs in Figure 2 we see that graph is cyclic (from $T1$ to $T3$ to $T2$ to $T1$) and thus has a global deadlock.

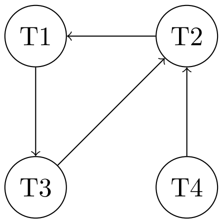


Figure 2: Global waits-for graphs for Node 1 – 3

3.3 Explain a scenario starting from the situation above in which T3 is allowed to commit.

In order to resolve the deadlock and let $T3$ commit, $T2$ would first have to be aborted. Once the deadlock has been resolved, $T3$ would be committed, using the two-phase commit protocol, first by having the transaction coordinator issuing the voting/prepare phase by sending a `prepare` message to node 1 and 3 - we can disregard Node 2, as it does not contain any of the objects used by $T3$. Before sending the prepare message to Node 1 and Node 3 the coordinator makes sure to record this in its log. Once this message is received by the nodes, their voting decision is recorded in each of their local logs followed by a reply back to the coordinator. Here all of the nodes have to reply with a yes/ok to allow the second phase (termination phase) of the successful commit to continue. Here each of their individual decisions are recorded in their local logs, prior to sending the answer. Once the coordinator receives yes/ok from both node 1 and 3, it then issues a `commit` message to the nodes which would start the local commit. Before issuing this outcome (commit/abort) message to the sites/nodes the outcome is logged in the coordinator. When the transaction has been committed locally in nodes, this is recorded in the local log, followed by an acknowledgment message to the coordinator. Lastly, when the coordinator has received an acknowledgment message from both nodes, the commit has been successful and the client receives the result.