

CS138 Programming Assignment 4: Fault

Assignment Out: April 4, 2006
Assignment Due: April 17, 2006 (11:59 pm)

1 Introduction

You have a distributed database which ensures consistency across simultaneous updates at different nodes. Now what happens when nodes go down and can't pass messages?

You're going to modify your Distributed Database (we'll stop calling it "simple" at this point) to deal with fault tolerance issues. Fault tolerance is a very general concept, so before you get down to coding, you will take part in specifying what exactly it means in the context of your DDB.

2 The Assignment

There are two parts to this assignment: specification and coding. As part of your handin, you must hand in a document which is a list of guarantees that your system makes against failure. The coding piece is to implement these guarantees. You get to design your own guarantees. If you implement reasonable behavior for the following set of cases, you will get full credit, but significant extra credit will be offered for implementing stronger or more general guarantees.

You must design guarantees for and handle the following cases:

- The root goes down – the machine crashes, or the process crashes.
- Any other node goes down.
- Temporary network failure – a node is disconnected from the network, but doesn't crash. It then returns to the network some time later (\leq some amount of time T).

For single-node failure as in the cases above, your system must at least:

- Maintain database consistency
- Continue normal operation of the database without the failed node

3 Guarantee Design

As part of your electronic handin, you should hand in a file called `GUARANTEES` (or something to that effect) which contains the list of guarantees that your system provides. This is an itemized list of “trouble case: system response” pairs. This should be written for a hypothetical maintainer of the database, who understands and wants details about how the tree structure will change.

Example:

- The root node’s system crashes: the system will discover the crash within ten seconds. The old root’s first child will become the new root. All other nodes will join themselves into the structure in an unspecified order.

Note that you don’t have to handle this case in exactly this way.

4 Implementation

Once you’ve designed your guarantees, you must implement them in your DDB. There’s no support code for this part of the task. Ensure that your old functionality (`sddb` and `quorum`) all still work, and make sure to test your new fault tolerance guarantees.

5 Code Exchange

As usual, you should write a `README` to document notable bugs and features, as well as other information that you think the TAs should know.

To hand in Fault, run the following script from the directory containing your source code:

```
rm -f *.class; /course/cs138/bin/cs138_handin fault
```

6 Graduate and Extra Credit

- Network Partition - Create and implement a guarantee for when the network is partitioned – i.e., some subset of nodes can no longer communicate with any nodes outside the subset. This guarantee must at least allow the nodes in the majority partition (if there is one) to continue database operation.
- Notification - implement user notification when there is a failure. This will probably require interface changes to the constructor and/or `DDBQuorum` interface to provide some sort of notification interface (simply printing errors to the screen isn’t useful). More credit will be

given for notifications of many different types of failure events. Examples: notification when another node goes down, recognition of network failure at this node, recognition of network partition.

- Multiple Simultaneous Failure - Handle the case where you have some small number $k > 1$ of nodes failing simultaneously.
- Restart and Recovery - When nodes come back up or become reconnected to the network, ensure that the database stays consistent in the case of multiple failure, network partition, or long-term disconnection.
- Other Guarantees - Implement other stronger or more general guarantees. Contact the TAs to ensure you can receive credit for what you do. Feel free to post ideas on the newsgroup and we can have an open discussion about guarantees.

For graduate credit: you can do either Multiple Simultaneous Failure or Network Partition, or you come up with something else that is about the same difficulty (run it by the TAs first if you do).