

Byzantine Agreement/Protocol

A fundamental problem of distributed computing is that of simulating a (secure) broadcast channel (see also KommunikationsInfrastruktur), within the setting of a P2P network.

A byzantine aggrement can always be reached, if more than 2/3 of the parties are honest, i.e., cast vote for the correct result according to their actual input (which in turn might be falsified).

If this sound too "technical", here is a real world application/implementation scrutinised.

Byzantine Agreement in Askemos

Askemos deploys atomic broadcast protocol (see Sintra section 2.2) to synchronize ProcessStep?'s with slight variations:

- A ProcessStep? is defined in such a way, that the binary value to agree in the voting (the checksum of state changes during the step) is often deterministic. Therefore the agreement protocol does not need to proceed in (possibly infinit many) rounds.

"Often deterministic" means here, is deterministic, if it depends exclusive on the input and internal state of the process. But if, for instance, the process reads additional input with "fetch" while processing the message, or depends on local values like the current time, it can become non-deterministic. See the accuracy test for actual measurements of such a case. The BALL implementation could easily proceed in additional rounds as standard byzantine algorithm do. I don't think additional rounds should be made standard behaviour instead we should put that into applications control.

- If the network is fragmented messages can be lost. If nodes miss - one by one - the final ready message in the agreement, the network can get out of sync in such a way, that resynchronisation becomes impossible. The BALL implementation extents the echo/ready messages in such a way, that the last phase can be recovered.

Further Work

Utilizing a setup or preprocessing phase it is possible to lower that requirement to some extend, Y. Lindell, A. Lysyanskaya and T. Rabin show

http://www.wisdom.weizmann.ac.il/home/lindell/public_html/composeBA_abs.html upper bounds of utility of that approach.

TODO The 0.7.x version of BALL deploys HTTPS as node-node protocol. Availability of a second message bus (from the references) is desired feature. The current implementation will be kept readily available and brought forward to protect against anticipated, future security bug in the alternate message bus, to be be deployed at the (accepted) cost of degrading performance until the bug is fixed.

References

Byzantine Generals Problem

Leslie Lamport, Robert Shostak, and Marshall Pease, ACM Transactions on Programming Languages and Systems, Vol. 4(3), July 1982, Pages 382-401

L. Kesteloot: "Fault-Tolerant Distributed Consensus" (1995).

Sintra (16. 4. 2002)

ByzantineAgreement

M. Naor and U. Wieder. A simple fault tolerant distributed hash table, 2003

<http://citeseer.ist.psu.edu/560557.html>

spread

a unified messaging bus (candidate for use in ball implementation).

The recovery algorithm of spread is quite similar to our implementation. The main difference is that when spread delivers a message to the application layer the corresponding Askemos event is the permanent commitment of a transaction (see ProcessStep₂).

Ensemble

Ensemble another unified messaging bus (candidate for use in ball implementation)

Sintra (<http://www.zurich.ibm.com/~cca/papers/sintra.ps>)

A fault tolereant replication architecture based on ByzantineAgreement.

Sitar: <http://sitar.anr.mcnc.org/>

building intrusion tollerant systems from off the shelf components.

on (broadcast) group membership protocols

<http://www.cs.colorado.edu/~mishras/research/papers/pdcs03.pdf>

Bft <http://pmg.lcs.mit.edu/bft/>

A byzantine file system. (No byzantine processes.)

symetric cluster management

<http://sources.redhat.com/cluster/faq.html>

ibm in Zürich: <http://www.zurich.ibm.com/csc/infosec/dti.html>

--

<http://epubs.siam.org/sam-bin/dbq/article/18708>

--

<http://www.cs.bham.ac.uk/~dxp/prism/byzantine/>

--

A CORBA₂ based implementation:

<http://beta.ece.ucsb.edu/immune/Immune.html>

A small essay <http://szabo.best.vwh.net/coalition.html>

see also <http://www.google.com/search?q=Byzantine+Generals+Problem>

related notes

On hardening the underlying host system: <http://immunix.org>

Last modifikation: Thu, 03 Aug 2006 13:17:16 +0200

Author(s): jfw,

Document number A849640f672ed0df0958abc0712110f3c page ByzantineAgreement delivered to public at
Wed, 08 Sep 2010 15:45:37 +0200