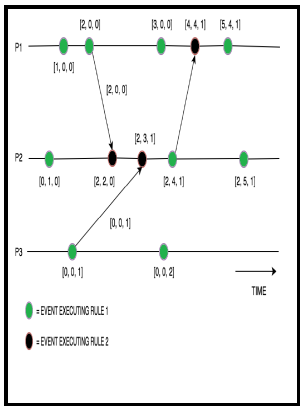
Consensus power of shared-memory distributed systems.

- - On the Power of Breakable Objects



Description: -
-consensus power of shared-memory distributed systems.
-consensus power of shared-memory distributed systems.
Notes: BMED copy has 111 leaves.
This edition was published in 2000

Tags: #Relationships #between #broadcast #and #shared #memory #in #reliable #anonymous #distributed #systems

Distributed Consensus in Block Chain

We use our model to derive two kinds of results: 1 lower bounds on contention for well known basic problems such as agreement and mutual exclusion, and 2 trade-offs betwe. That is, the output value of a consensus protocol must be the input value of some process. Unfortunmely, the only known implementation of a concurrent time stamp sys tem has been theoretically unsatisfying since it requires unbounded size time-stamps, in other words, unbounded memory.

On the Power of Breakable Objects

A process can write a value to such an object, and gets back a snapshot of its contents. Protocols that solve consensus problems are designed to deal with limited numbers of faulty.

CiteSeerX — Citation Query On Achieving Consensus Using a Shared Memory

Some models may deal with fully connected graphs, while others may deal with rings and trees. A distributed system is uniform if all processors with the same number of neighbors are identical.

CiteSeerX — Citation Query On Achieving Consensus Using a Shared Memory

The protocols are consistent, nontrivial, and they guarantee that each process decides after a finite expected number of steps. The purpose of this paper is to give an introduction to this research area, using an iterated model based on the safe-consensus task Afek, Gafni and Lieber, DISC'09.
.

An Introduction to the Topological Theory of Distributed Computing with Safe

In one round, a process may send all the messages it requires, while receiving all messages from other processes. The two message-passing models considered are a com.

CiteSeerX — Citation Query On Achieving Consensus Using a Shared Memory

The overhead introduced by these emulations is polynomial in the number of processors in the systems.

Related Books