NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications

Prof. Sandip Chakraborty

Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Lecture 26: Consensus for Permissioned Models

CONCEPTS COVERED

- Permissioned Model
- State Machine Replication

KEYWORDS

- **Consensus on State Machines**

Permissioned Model

- A blockchain architecture where users are authenticated a priori
  - A Membership Service Provider (MSP) helps to obtain the chain membership
- Users know each other
  - However, users may not trust each other
  - Security and consensus are still required.
- Run blockchain among known and identified participants

Permissioned Model

- Particularly interesting for business applications – execute contracts among a closed set of participants

- Example: Provenance tracking of assets in a supply chain

A Use Case

Smart Contracts over Closed Networks

- **Smart Contracts:** “A self-executing contract in which the terms of the agreement between the buyer and the seller is directly written into the lines of code” - http://www.scalablockchain.com/

Smart Contracts over Closed Networks

- **Agreement on a Smart Contract Execution:**
  - Store the contract on a blockchain
  - Once an event is triggered, execute the codes locally on each peer
  - Generate transactions as the output of the contract execution
  - The peers of the blockchain network validates the transaction, and the output is committed in the blockchain – may trigger the next event to execute the code further

Smart Contracts over Closed Networks

- **Agreement on a Smart Contract Execution:**
  - Store the contract on a blockchain
  - Once an event is triggered, execute the codes locally on each peer
  - Generate transactions as the output of the contract exe
  - The tra blo e coc

Do we really need to execute the code on each peer?

When does each peer execute the code?

Smart Contracts over Closed Networks

- Execute contract at a subset of nodes, and ensure that the same state is propagated to all the nodes
  - Majority of the peers should agree on the state
  - **Validation:** Generate a "***proof***" that a peer has agreed on the "*state of execution*"

Smart Contracts over Closed Networks

- Execute contract at a subset of nodes, and ensure that the same state is propagated to all the nodes
  - Majority of the peers should agree on the state
  - **Validation:** Generate a "***proof***" that a peer has agreed on the "*state of execution*"

How will we generate the proof?

Smart Contracts over Closed Networks

- Execute contract at a subset of nodes, and ensure that the same state is propagated to all the nodes
  - Majority of the peers should agree on the state
  - **Validation:** Generate a "***proof***" that a peer has agreed on the "*state of execution*"

Smart Contract as State Machines

- **State Machine Replication:**
  - Represent the smart contract as a state machine – Remember, any deterministically executable code can be represented as a state machine

Smart Contract as State Machines

```
S1:
while (moreGoods == 1)
   DeliverGoods();
S2:
if (allOrderComplete == 0) goto
S1;
else {
   S3:
   printf("Goods transfer
complete");
}
```

State Machine Replication

Replicate the state machine on multiple independent servers

State Machine Replication

State Machine Replication

State Machine Replication

State Machine Replication

State Machine Replication

State Machine Replication

State Machine Replication

State Machine Replication

State Machine Replication

Inform

Importance of Consensus

Importance of Consensus

Majority Voting – Agree on S2

Conclusion

- Consensus is still required within a closed environment
- How can we extend state machine replication for permissioned models?

Thank you