Bimodal Multicast - Probabilistic Broadcast

Antonio Bucchiarone

Fondazione Bruno Kessler, Trento – Italy
bucchiarone@fbk.eu

11 November 2020

Bimodal Multicast

- The Probabilistic Broadcast, that is referred to *Bimodal Multicast*, is inspired to *Gossip Protocol* and it satisfies several properties:
  - **Atomicity**, almost all or almost none, hence the protocol guarantees with high probability that each multicast reaches almost all processes and with low probability that every multicast reaches a small set of processes.
  - **Throughput stabi**lity, the variation of throughput is lower than the normal multicast.
  - **Ordering**, the messages are delivered in FIFO order.
  - **Multicast stability**, the protocol guarantees the stability of messages. Detection of lost messages, the processes recover the loss message through a particular procedure.
  - **Scalability**, the costs are constant or grow slowly which are based on the network size.

Bimodal Multicast

- The purpose of probabilistic broadcast is
  - To exchange messages between nodes using the Gossip protocol behaviour and re-transmitting the lost messages when a transmission error occurs.

  - we can analyze:
    - the *throughput* over time with some particular settings (e.g. Round Retransmission Limit),
    - the *performance* of the protocol,
    - the *probability of success* of pbcast with different network sizes,
    - the *delivery distribution* of the protocol.

Implementation - I

- Agent-based model involving processes which exchange message through them.
- The Probabilistic Broadcast is composed by several agents
  - MessageHandler
  - Message
  - PBCastBuilder
  - Process
  - Tree
  - Node
- The main agent that is involved in our analysis is ***Process***
  - its main goal is sending and delivering messages that are sent from the root of the spanning tree to its children and then the messages are forwarded through the *spanning tree*.
- *PBCastBuilder* agent computes the spanning trees of all processes

Implementation - II

- The ***number of children*** is chosen before starting the simulation using two parameters Children for node (min) and Children for node (max).
- For each iteration, the *process* can perform two different actions: *send* message or r*eceive* message.
- The number of actions is limited for each iteration by a *workload* which is chosen *randomly* between minimum and maximum values set using Workload for node (min) and Workload for node (max) parameters.

- The processes are located within three *ContinuousSpace.*

- The *ContinuosSpace* gives us the opportunity to represent:
  - the spanning tree of a single process
  - the messages that are exchanging in all spanning tree in the current step
  - the messages that are exchanging in a single spanning tree which is related to the first *continuousSpace.*

Execution

- For *each iteration* is possible to see the *sent messages* which are represented by *edges.*
- The edges can assume different colour to indicate several situations, such as message received (green), message lost (red) and message retransmitted (yellow).
- The processes have a **status** which can acquire four different values: QUIETE (blue), SENDER (red), RECEIVER (green) and CHILD (yellow).
- The architecture **is not composed by a coordinator** agent which triggers the event. Therefore, the **Process agent triggers the pbcast through the step method**.

Implemented Protocols

- The implemented Probabilistic Broadcast Protocol is composed by two sub-protocols:
  - **Optimistic Dissemination Protocol** is the first stage of the Probabilistic Broadcast and implements IP multicast or randomized dissemination protocol, in case the connection is not present, where the processes use a spanning tree in order to exchange the messages.
  - **Two-Phase Anti-Entropy Protocol** is the second phase of our protocol and its purpose is recovering the loss messages since the multicast is unreliable.

Optimistic Dissemination Protocol

- It used by Probabilistic Broadcast in order to implement unreliable multicast.
- All messages use an unreliable multicast primitive using IP multicast or randomized dissemination protocol.
- Every process broadcasts messages using a generated spanning tree.
- The messages is sent to the children of the sender that deliver the message and then forward it including the tree identifier.
- There are several spanning trees, which are many as the number of the processes involved in the experiment.
- The spanning trees are realized at the beginning of the experiment and the code is implemented in the PBCastBuilder agent.
- The root of each spanning trees is different. The number of the children is chosen using the parameters described above.
- The children are selected inspecting the grid in the continuousSpace, the process examines the space in order to add N children.
- The spanning tree is built starting from a root process which uses a distance variable in order to add the near processes as children of the tree. The processes are spread in the continuousSpace in order to avoid overlapping processes.

Two-Phase Anti-Entropy Protocol

FONDAZIONE BRUNO KESSLER

- It is executed randomly by nodes using the value of gossip probability parameter.
- The node, that starts the protocol, sends a gossip message which contains the round number and the digest.
- The receiver computes the digest of its history and compare it with the received one in order to verify whether it has received all messages.
- If the receiver does not have all messages, then it sends a solicitation message to the sender in order to retrieve the lost messages.
- This protocol is able to retrieve the lost messages, but also to anticipate the delivery.
- The receiver can send a solicitation message in order to retrieve the messages that do not yet reach the process since they have to pass through the spanning tree.
- The protocol has several optimizations in order to limit its cost

Simulation - Parameters

- Children per node (Max)
  - number of maximum children for every process.
- Children per node (Min)
  - number of minimum children for every process.
- Gossip message limit
  - it indicates the limit which is reached in order to remove the message from the history of each process.
- Message loss probability
  - the probability to lose a message during packet exchange.
- Number of processes
  - number of processes during the experiment.
- Number of processes in a round
  - number of processes that are select as receivers of the gossip messages.
- Probability to send a gossip message
  - the probability to send a random gossip message to round members.

Simulation - Parameters

- Probability to send a pbcast
  - the probability to send a pbcast through the spanning tree.
- Process ID to view
  - shows the spanning tree of particular process where the messages are exchanged.
- Retransmission limit per round
  - the maximum number of message that a process can retransmit
- Single pbcast: the pbcast can sent from a single process (0) or from all processes (-1 is the standard mode).
- Workload per node (Max)
  - the maximum number of operations that a process can perform in a single step.
- Workload per node (Min)
  - the minimum number of operations that a process can perform in a single step. The workload decreases the efficiency of the protocol with low values, since the broadcast is spread slowly.

Analysis

- **PBCast Bimodal Delivery Distribution**
  - the evolution of the delivery distribution of the protocol with different network sizes and with 5% of lost messages.
  - The likelihood which almost all receive the multicast is very high.
  - The likelihood that a small number of processes will receive a multicast is low.
  - The intermediate values has a low probability as we can see for 20, 25 and 30 as number of processes