Impossibility Results for Distributed Transactional Memory

Paper Reading Group

Costas Bunsch
Maurice Herlihy
Miroslav Popovic
Gokarna Sharma
Presents: Maksym Planeta

12.11.2015

Table of Contents

Table of Contents

Distributed transctional memory

- Distributed system
- Set of objects
- Gather all objects

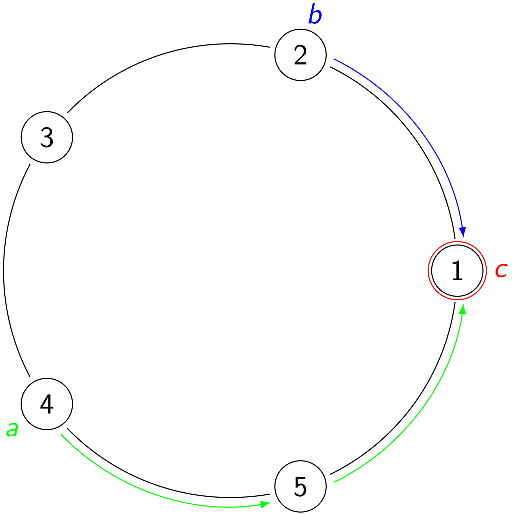
Types of transactional memory

- Data-flow
- Control-flow
- Hybrid-flow

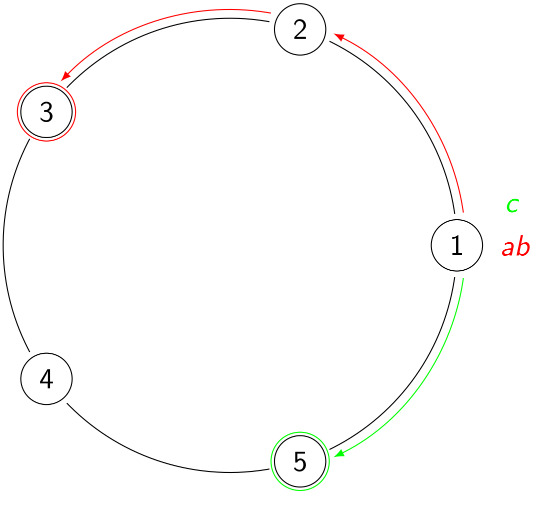
System model

- Data-flow based implementation
- Network graph
- Instantaneous transaction
- Atomic receive-compute-send
- All delays are equal
- Single copies
- One transaction per node

Transactions and objects



Several transactions on the same objects



Evaluated network

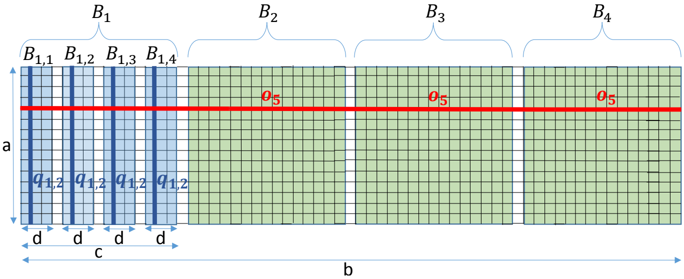


Figure 1: The graph G for the time-communication impossibility result, with $a = c = 16$, $b = 64$, $d = 4$, and $\gamma = \delta = 4$.

Transaction scheduling

In this schedule the transactions in G execute sequentially, one after another, in a column by column way starting from the leftmost column of G up to the rightmost column. All transactions in column j finish execution before the transactions in column $j+1$ start execution.

Communication costs

For a set of transactions $\mathcal{T}$ in graph G, the communication cost of an execution $\mathcal{E}$ is the sum of the traversed path lengths of all messages sent during $\mathcal{E}$. The communication cost of scheduling algorithm A is the maximum communication cost over all possible executions for $\mathcal{T}$

Execution time

For a set of transactions $\mathcal{T}$ in graph G, the time of an execution $\mathcal{E}$ is the time elapsed until the last transaction finishes its execution in $\mathcal{E}$. The execution time of scheduling algorithm A is the maximum time over all possible executions for $\mathcal{T}$.

Impossibility Result

> *in this instance it is impossible to simultaneously optimize execution time and communication cost*

If one optimizes for lower communication costs, one pays with higher execution time. And vice versa.

Positive result

We give algorithms which minimize independently the communication cost or execution time in an arbitrary graph G.

Small communication costs

The problem of minimizing the communication cost is NP-hard, by a reduction from graph TSP.

Given a graph G, we can approximate the optimal communication cost using a universal TSP tour.

Small execution time

We will reduce the vertex coloring problem to this problem. The coloring problem aims at finding the chromatic number $\chi(H)$ of a graph H, and it is an NP-hard problem.

Small execution time algorithm

We construct G to be isomorphic with H such that each edge in G has weight 1.

Small execution time algorithm

We construct G to be isomorphic with H such that each edge in G has weight 1. Each node in G holds a transaction.

Small execution time algorithm

We construct G to be isomorphic with H such that each edge in G has weight 1. Each node in G holds a transaction. For each edge $\epsilon = (u, v)$ in H we create a new object in G to be used only by the respective transactions in the adjacent nodes in G.

Small execution time algorithm

We construct G to be isomorphic with H such that each edge in G has weight 1. Each node in G holds a transaction. For each edge $\epsilon = (u, v)$ in H we create a new object in G to be used only by the respective transactions in the adjacent nodes in G. Objects can be initially placed in any of the nodes with transactions that request them.

Small execution time algorithm

We construct G to be isomorphic with H such that each edge in G has weight 1. Each node in G holds a transaction. For each edge $\epsilon = (u, v)$ in H we create a new object in G to be used only by the respective transactions in the adjacent nodes in G. Objects can be initially placed in any of the nodes with transactions that request them. Given a transaction execution schedule in G, we can find a valid vertex coloring in H by simply converting the time step that each transaction executes to a color.

Small execution time algorithm

We construct G to be isomorphic with H such that each edge in G has weight 1. Each node in G holds a transaction. For each edge $\epsilon = (u, v)$ in H we create a new object in G to be used only by the respective transactions in the adjacent nodes in G. Objects can be initially placed in any of the nodes with transactions that request them. Given a transaction execution schedule in G, we can find a valid vertex coloring in H by simply converting the time step that each transaction executes to a color. Therefore, an execution schedule in G has duration χ time steps if and only if H has a valid coloring with χ colors.