Divergence visualization in Lasp program execution

Creupelandt Grégory 59191300

Advisor: Peter Van Roy

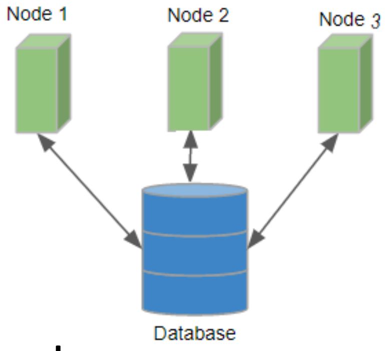
1. Context:

Usual approach:

Distributed variables are handled by a database:

- Causality is important
- Message order is important (scheduler)

➡ Consensus **is** required !

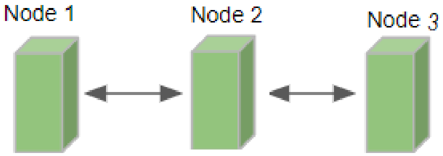


New approach:

Distributed variables are handled by CRDT (peer-to-peer):

- Causality is automatically handled by metadatas
- Message order has no impact (no scheduler required)

➡ Consensus **is not** required !



No need for database server !
Much less cumbersome algorithms !
Already Adopted by some big companies:

2. How does it work ?

- CRDT is for Conflictfree Replicated Data Type.
- It's a datastructure with values and metadatas.
- From time to time, nodes send it to their peers.

Important notion: it is **CONVERGENT:**

Information is diffused everywhere
Messages are comutative

➡ Every node eventually end up with the same state !

3. Lasp: an Erlang solution for distributed application using CRDT

Fully automated, developer has very little control.

➡ My job: Develop an API to allow more control:

Tools to **visualize** the convergence time.
Tools to **modify** the convergence time.
Use these tools to **test** some hard cases.

4. Update speed VS Convergence speed

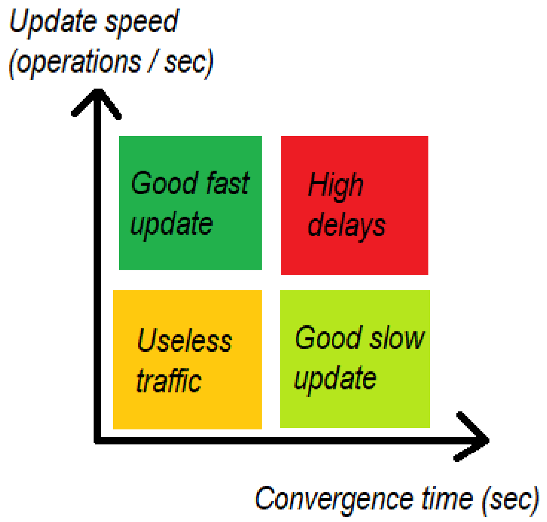
Is it useful to have a system that converges in 5ms when we update the value only once per minute? Probably not.

➡ Useless high network traffic…

Is it useful to have a system that converges in 10 sec when we update the value every 5 ms ? Probably not.

➡ Huge delays…

Interesting relation to explore between these two notions !


Update speed (operations / sec)
Good fast update
High delays
Useless traffic
Good slow update
Convergence time (sec)