Staying CALM Beyond Deterministic Queries

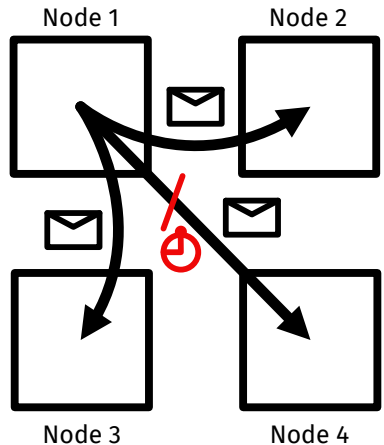
Tim Baccaert Bas Ketsman

Vrije Universiteit Brussel

Dutch-Belgian Database Day, 2022

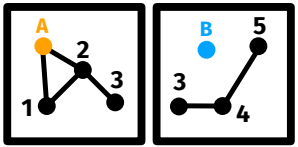
Shared-Nothing
State is local to each node, communication done through message passing.

Asynchronous Messaging
messages may arrive out-of-order and with arbitrary (but finite) delay.



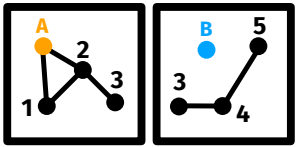
Pathfinding

Is there a path from A to B?



Pathfinding

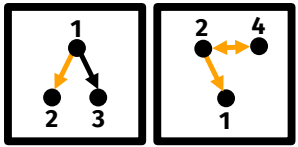
Is there a path from A to B?



Node 1 Node 2

Deadlock Detection [5]

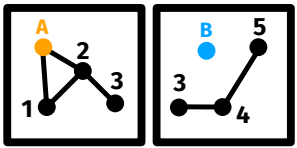
Is there a cycle in the waits-for graph?



Node 1 Node 2

Pathfinding

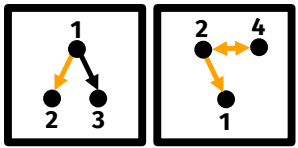
Is there a path from A to B?



Node 1 Node 2

Deadlock Detection [5]

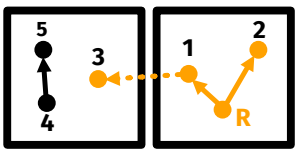
Is there a cycle in the waits-for graph?



Node 1 Node 2

Garbage Collection [5]

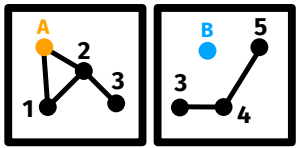
Is this memory object still referenced?



Node 1 Node 2

Pathfinding

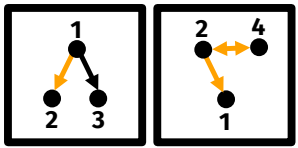
Is there a path from A to B?



Node 1 Node 2

Deadlock Detection [5]

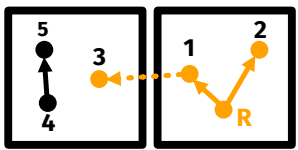
Is there a cycle in the waits-for graph?



Node 1 Node 2

Garbage Collection [5]

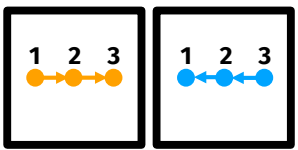
Is this memory object still referenced?



Node 1 Node 2

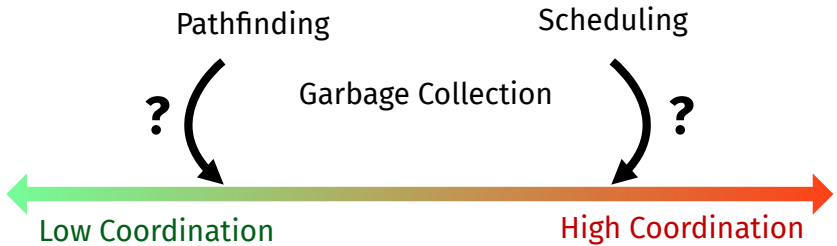
Scheduling

Do all nodes agree on an execution order?



Node 1 Node 2

Deadlock Detection



Nodes can run almost independent of each other

e.g., Conjunctive Queries, ...

Highly complex dependency relation between nodes

e.g., Paxos, 2PC, Raft, ...

CALM Theorem [1, 2, 4]

Consistency And Logical Monotonicity

CALM Theorem [1, 2, 4]

***C**onsistency **A**nd **L**ogical **M**onotonicity*

Monotonic Query

Nodes oblivious to other nodes a priori

No Coordination

CALM Theorem [1, 2, 4]

***C**onsistency **A**nd **L**ogical **M**onotonicity*

Monotonic Query	Nodes oblivious to other nodes a priori	No Coordination
$\subseteq$		
Semi/Disjoint-Monotonic Query	Nodes know data partitioning strategy (+ oblivious to nodes)	Light Coordination

Pathfinding in **Datalog** (*i.e.*, monotonic query language)

```
Path(x, y) :- Edge(x, y)
Path(x, z) :- Path(x, y), Edge(y, z)
```

Pathfinding in **Datalog** (*i.e.*, monotonic query language)

```
Path(x, y) :- Edge(x, y)
Path(x, z) :- Path(x, y), Edge(y, z)
```

Deadlock Detection in **Datalog**

```
WaitsFor(x, y) :- DirEdge(x, y)
WaitsFor(x, z) :- WaitsFor(x, y), DirEdge(y, z)
Deadlock() :- WaitsFor(x, y), WaitsFor(y, x)
```

Pathfinding in **Datalog** (*i.e.*, monotonic query language)

```
Path(x, y) :- Edge(x, y)
Path(x, z) :- Path(x, y), Edge(y, z)
```

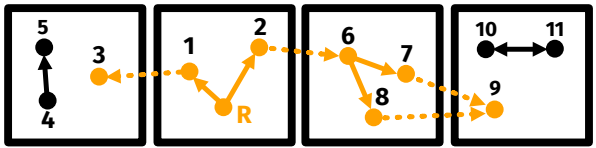
Deadlock Detection in **Datalog**

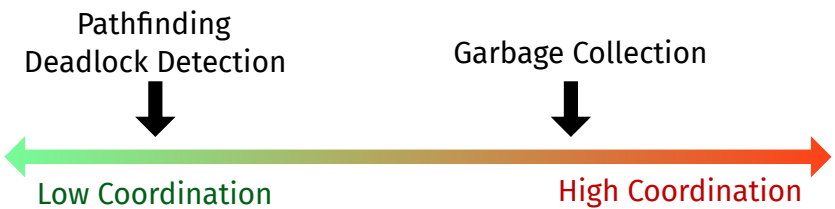
```
WaitsFor(x, y) :- DirEdge(x, y)
WaitsFor(x, z) :- WaitsFor(x, y), DirEdge(y, z)
Deadlock() :- WaitsFor(x, y), WaitsFor(y, x)
```

Garbage Collection [5]

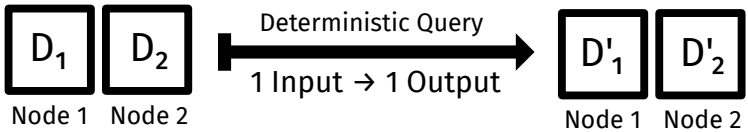
Nodes **cannot locally decide** if an object is garbage **until** it has seen the **entire graph**.

⇒ All nodes must have discovered each other

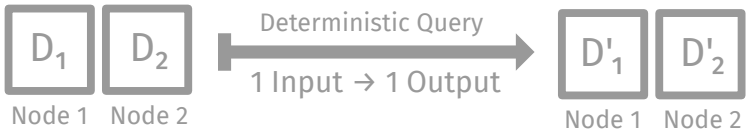




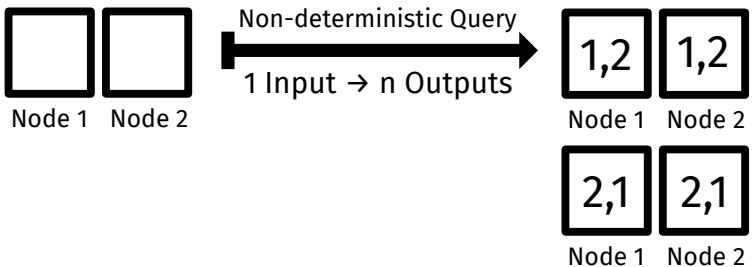
Examples until now...



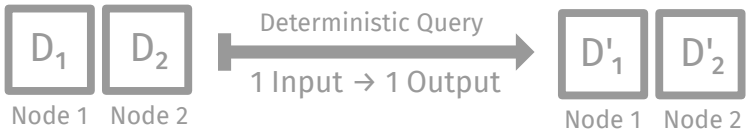
Examples until now...



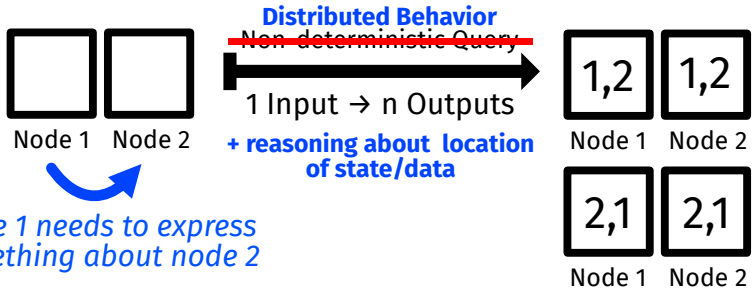
Selection



Examples until now...

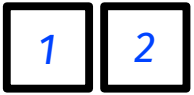


Selection



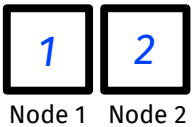
Constraints
Modeling Systems with Behaviors

Id: Nodes have a unique identity

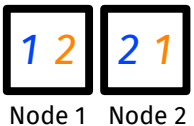


Constraints
Modeling Systems with Behaviors

Id: Nodes have a unique identity

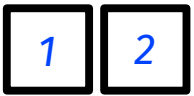


Id+All: each node knows the unique identity of all other nodes

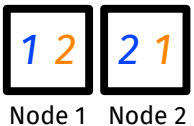


Constraints
Modeling Systems with Behaviors

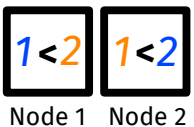
Id: Nodes have a unique identity

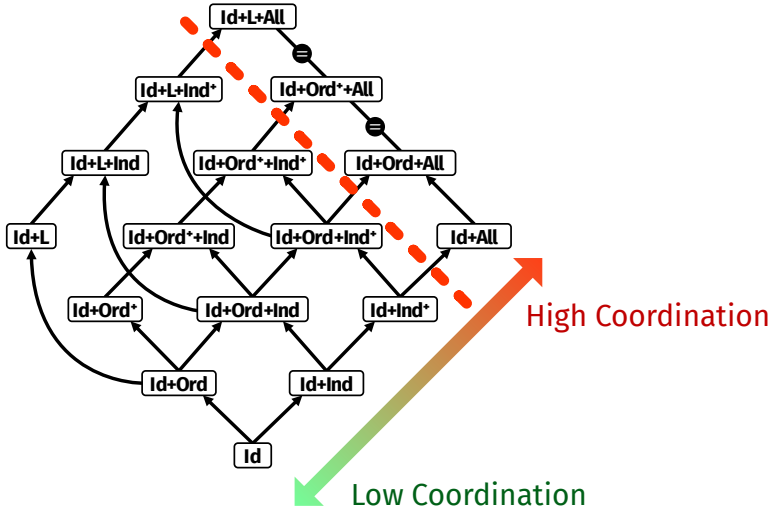


Id+All: each node knows the unique identity of all other nodes



Id+All+Ord: each node agrees on an order over all of its stored data/state



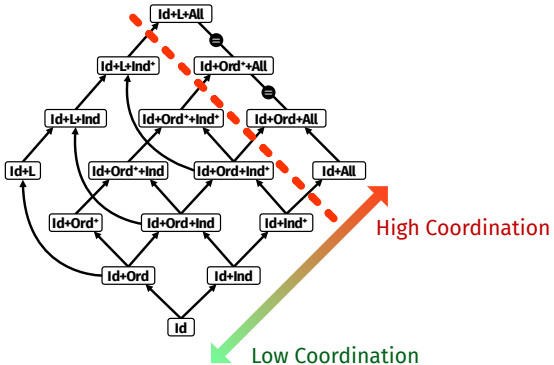

Id+L+All
Id+L+Ind+
Id+Ord+-All
Id+L+Ind
Id+Ord+-Ind+
Id+Ord+All
Id+L
Id+Ord+-Ind
Id+Ord+Ind+
Id+All
Id+Ord+
Id+Ord+Ind
Id+Ind+
Id+Ord
Id+Ind
Id
High Coordination
Low Coordination

CALM Theorem (Revisited) [3]

Id-Monotonic Query	**Id**-System	No Coordination
Id+Ind/Ind+-Monotonic Query	**Id+Ind/Ind+**-System	Light Coordination
C-Monotonic Behavior	System for which **C** holds	Depends on **C**

Scheduling requires a way for nodes to
make **globally consistent choices**
$\Longrightarrow$ **Ord**-constraints (*e.g.*, choices made by node w. max/min **Id**).

- CALM for languages with **non-determinism** and **data placement**.
- Generalized the CALM model to **arbitrarily configured data systems**.
- Introduced a **coordination spectrum**:



more powerful configurations = more inherent coordination

References

[1] Tom J. Ameloot, Frank Neven, and Jan Van den Bussche. 2013. Relational Transducers for Declarative Networking. *J. ACM.* 60, 2, 1-38.

[2] Tom J. Ameloot, Bas Ketsman, Frank Neven, and Daniel Zinn. 2016. Weaker Forms of Monotonicity for Declarative Networking: A More Fine-Grained Answer to the CALM-Conjecture. *ACM Trans. Database Syst.* 40, 4, 1-45.

[3] Tim Baccaert and Bas Ketsman. 2023. Distributed Consistency Beyond Queries (Accepted). *PODS 2023*.

[4] Joseph M. Hellerstein. 2010. The Declarative Imperative: Experiences and Conjectures in Distributed Logic. *SIGMOD Rec.* 39, 1, 5-19.

[5] Joseph M. Hellerstein and Peter Alvaro. 2020. Keeping CALM: When Distributed Consistency is Easy. *Commun. ACM.* 63, 9, 72-81.