Fine-grained Transaction Scheduling in Replicated Databases via Symbolic Execution

Pedro Raminhas

pedro.raminhas@tecnico.ulisboa.pt

Stage: 2nd Year PhD Student

Research Area: Dependable and fault-tolerant systems and networks

Advisors: Miguel Matos
Paolo Romano

Research Questions

- **Problem**:
  - Replication techniques incur non-negligible costs to maintain consistency among replicas
  - 2PC => Distributed *Deadlocks*
  - Classic SMR => Serial execution
  - Parallel SMR => Not to trivial to parallelize txs and maintain consistency

Research Questions

- **Problem**:
  - Replication techniques incur non-negligible costs to maintain consistency among replicas
  - 2PC => Distributed *Deadlocks*
  - Classic SMR => Serial execution
  - Parallel SMR => Not to trivial to parallelize txs and maintain consistency

- **Why is it a problem**:
  - Deadlocks
  - Automatic Conflict Class Prediction => Too Coarse Grained => Level of the table => Low Throughput
  - Manual Prediction => Hard and Not optimal
  - Avoid False negatives => Rollback of Txs, possibly inconsistencies among replicas

Research Questions

- **Problem**:
  - Replication techniques incur non-negligible costs to maintain consistency among replicas
  - 2PC => Distributed *Deadlocks*
  - Classic SMR => Serial execution
  - Parallel SMR => Not to trivial to parallelize txs and maintain consistency

- **Why is it a problem**:
  - Deadlocks
  - Automatic Conflict Class Prediction => Too Coarse Grained => Level of the table => Low Throughput
  - Manual Prediction => Hard and Not optimal
  - Avoid False negatives => Rollback of Txs, possibly inconsistencies among replicas

- **Approach**:

  Symbolic Execution => Recurring to Symbolic Execution, we are able to provide in a ***fine-grained*** and ***automatic*** way the set of objects/tuples accessed in a transaction.

Research Questions

- **Problem**:
  - Replication techniques incur non-negligible costs to maintain consistency among replicas
  - 2PC => Distributed *Deadlocks*
  - Classic SMR => Serial execution
  - Parallel SMR => Not to trivial to parallelize txs and maintain consistency

- **Why is it a problem**:
  - Deadlocks
  - Automatic Conflict Class Prediction => Too Coarse Grained => Level of the table => Low Throughput
  - Manual Prediction => Hard and Not optimal
  - Avoid False negatives => Rollback of Txs, possibly inconsistencies among replicas

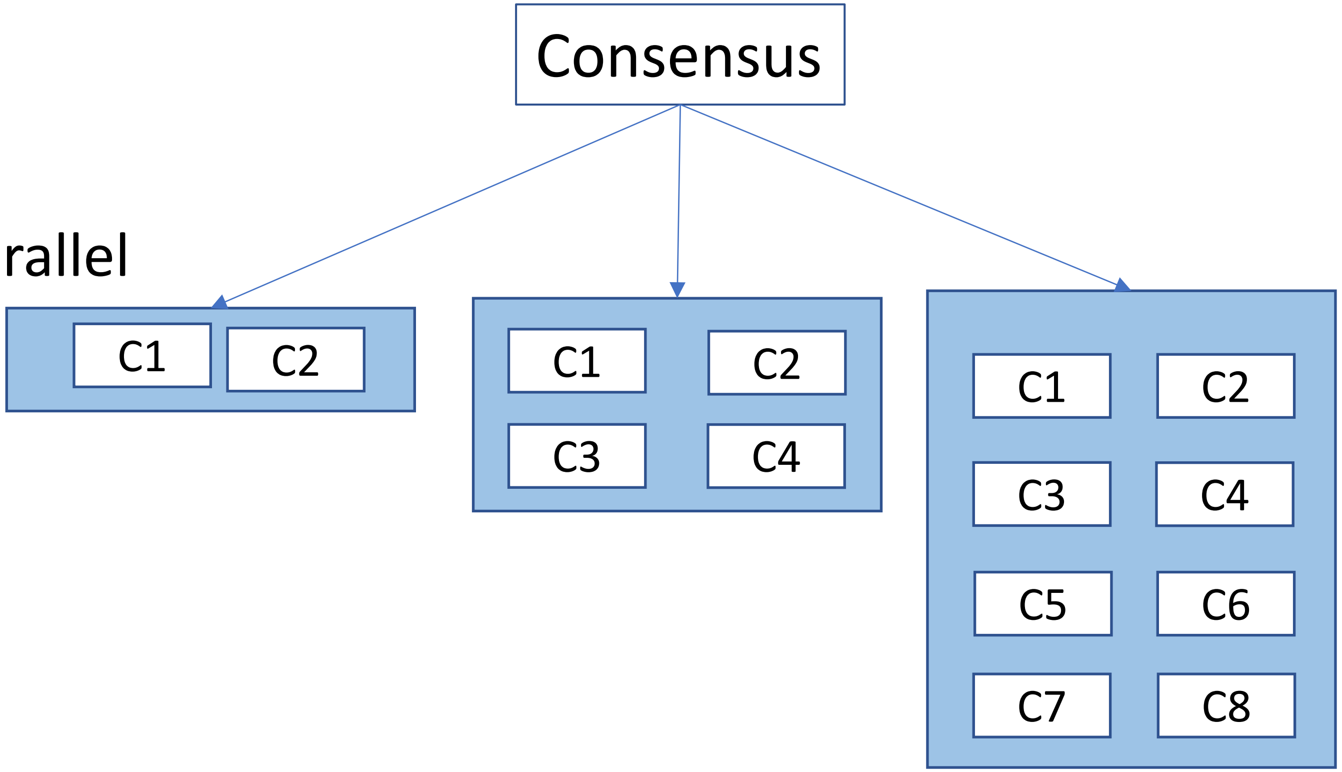
- **Approach**:

  Symbolic Execution => Recurring to Symbolic Execution, we are able to provide in a ***fine-grained*** and ***automatic*** way the set of objects/tuples accessed in a transaction.

- **Consequences:**
  - Better Parallelism
  - No Runtime Overhead => *offline analysis*
  - No False negatives
  - Determinist Scheduling algorithms benefit from fine-grained information

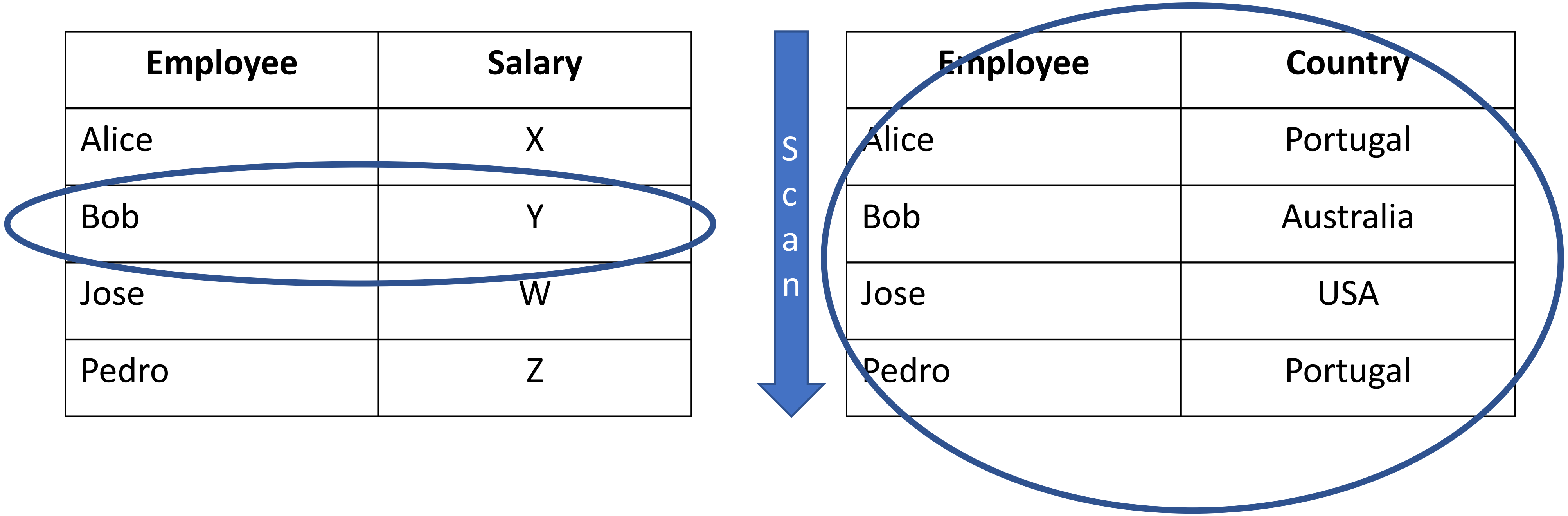
Approach

- *Most* CPUs are multicore!!!
  - Some Transactions can be run in Parallel
  - Replicas must remain consistent
  - Schedule must be deterministic



Approach

- *Compile-Time*
  - SE gives the set of keys/tuples accessed



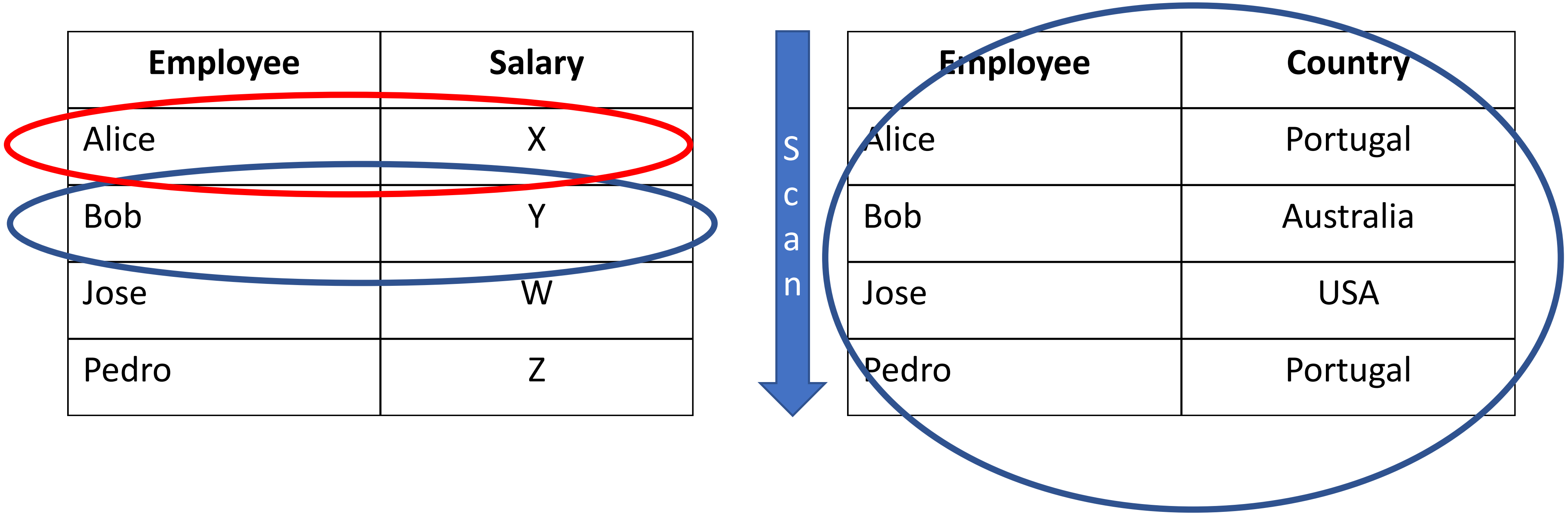
Employee	Salary
Alice	X
Bob	Y
Jose	W
Pedro	Z

Employee	Country
Alice	Portugal
Bob	Australia
Jose	USA
Pedro	Portugal

Tx A: Sum of Australia's Sallary : Entire Table Employee's Countries && Bob

Approach

- *Compile-Time*
  - SE gives the set of keys/tuples accessed



Employee	Salary
Alice	X
Bob	Y
Jose	W
Pedro	Z

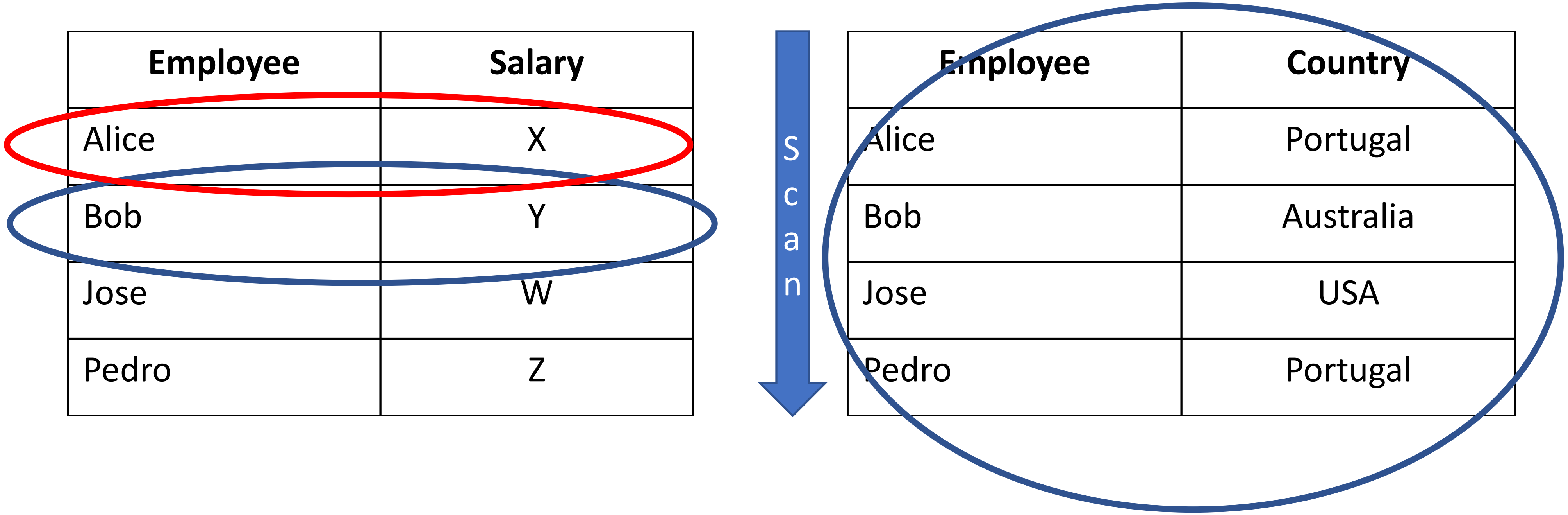
Employee	Country
Alice	Portugal
Bob	Australia
Jose	USA
Pedro	Portugal

Tx A: Sum of Australia's Sallary : Entire Table Employee's Countries && Bob

Tx B: Increase Alice Sallary : Alice

Approach

- ***Compile-Time***
  - SE gives the set of keys/tuples accessed



Employee	Salary
Alice	X
Bob	Y
Jose	W
Pedro	Z

Employee	Country
Alice	Portugal
Bob	Australia
Jose	USA
Pedro	Portugal

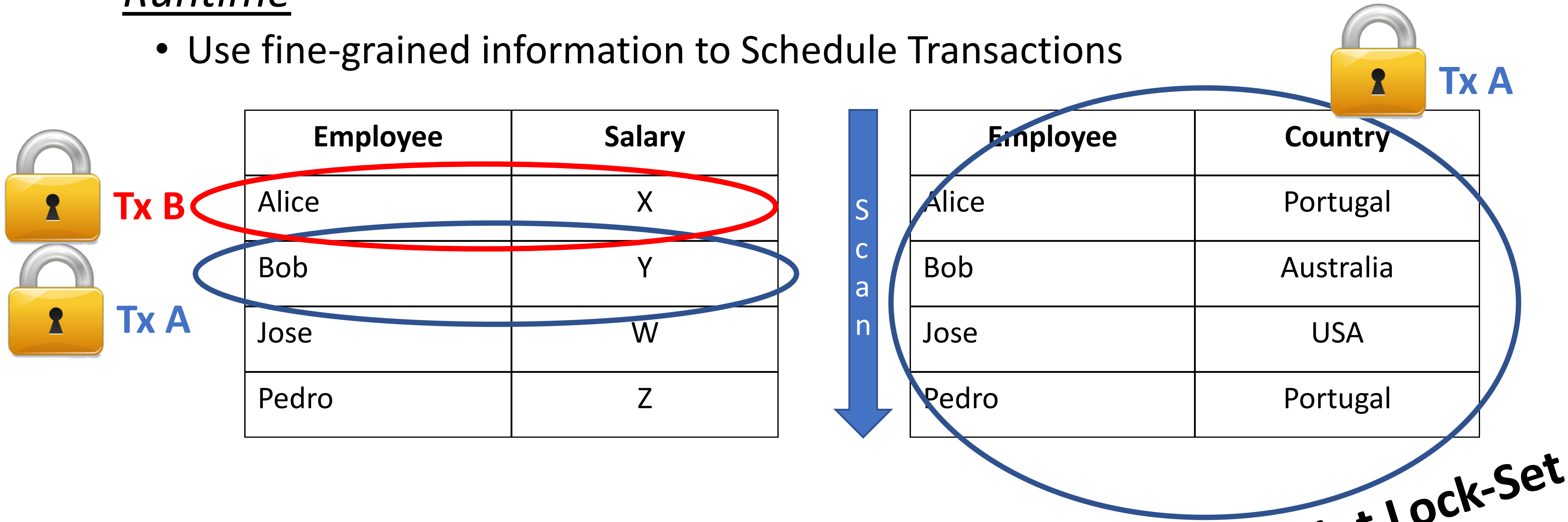
Tx A: Sum of Australia's Sallary : Entire Table Employee's Countries && Bob

Tx B: Increase Alice Sallary : Alice

Disjoint!!!!!

Approach

- *Runtime*
  - Use fine-grained information to Schedule Transactions



Employee	Salary
Alice	X
Bob	Y
Jose	W
Pedro	Z

Employee	Country
Alice	Portugal
Bob	Australia
Jose	USA
Pedro	Portugal

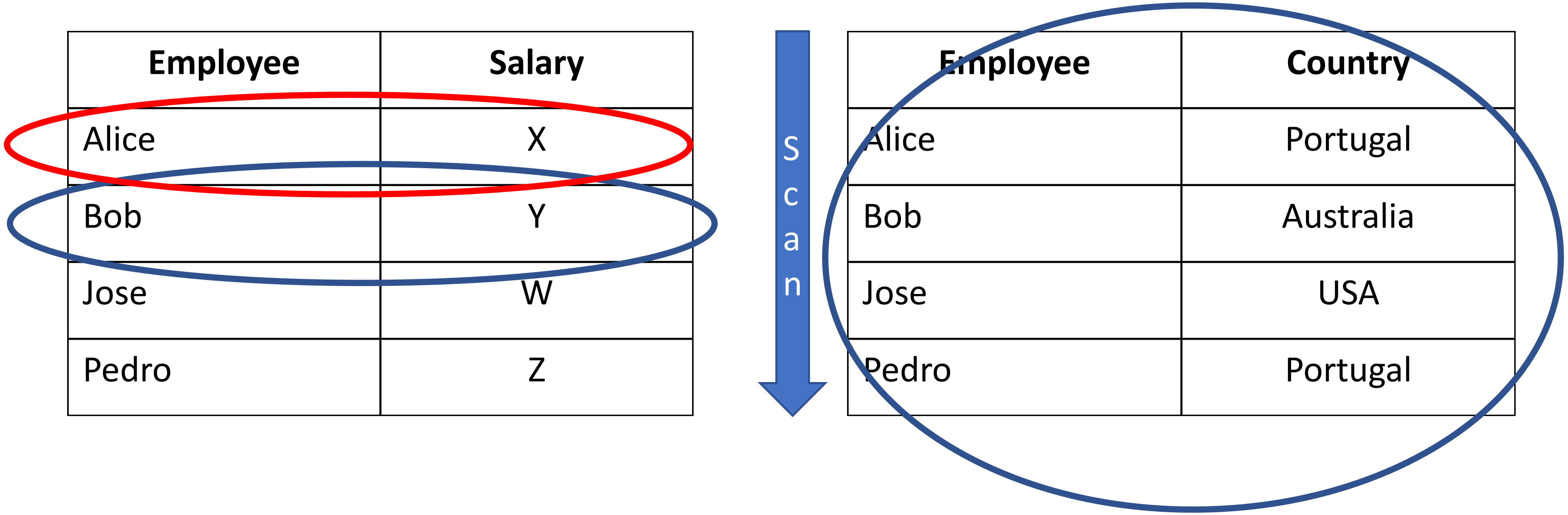
Tx A: Sum of Australia’s Sallary : Entire Table Employee’s Countries && Bob

Tx B: Increase Alice Sallary : Alice

Disjoint Lock-Set
Run in parallel

Approach

- *Runtime*
    - Use fine-grained information to Schedule Transactions



Employee	Salary
Alice	X
Bob	Y
Jose	W
Pedro	Z

Employee	Country
Alice	Portugal
Bob	Australia
Jose	USA
Pedro	Portugal

Tx A: Sum of Australia's Sallary : Entire Table Employee's Countries && Bob

Tx B: Increase Alice Sallary : Alice

Solver

Approach

Lock-based:

+ Simple

+ No runtime overhead

- Not trivial to parallelize lock table

Solver:

+ Provides the optimal schedule for batch

- Imposes runtime overhead

TÉCNICO LISBOA

inesc id lisboa

Thanks for the attention

Q&A