# Declarative Networking

**Tom J. Ameloot**

Frank Neven
Jan Van den Bussche

Hasselt University & Transnational University of Limburg

# Introduction

### Declarative

- high-level languages
- hide technical details, do optimization

### Networking

- distributed programs
- cloud computing, increase parallellism

### Recently combined

# Introduction

**Asynchronous communication**

unpredictable message delays

**(Eventual) consistency**

output the same regardless of delays

## CALM Conjecture

# Problem Description

Desire to increase parallellism

**Monotone operations**

embarrassingly parallel, coordination-free

# Problem Description

Desire to increase parallellism

**Monotone operations**
embarrassingly parallel, coordination-free

**[Hellerstein] CALM Conjecture:**

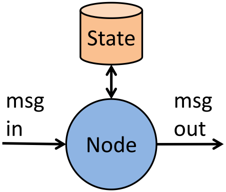
> *A program has an eventually consistent, coordination-free execution strategy if and only if it is expressible in (monotonic) Datalog.*

**Verify conjecture**
formalize: program, eventual consistency, coordination-freeness

# Programs

**Transducer $\Pi$**

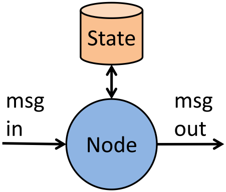
- Local state: input, output, memory, system
- Messages
- Driven by database queries



# Programs

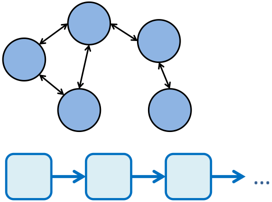
### Transducer $\Pi$

- Local state: input, output, memory, system
- Messages
- Driven by database queries



### Transducer Network

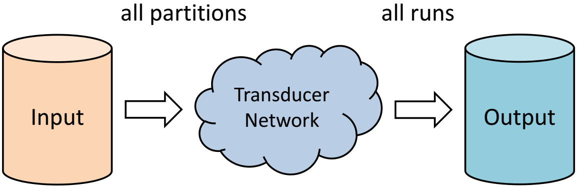
- Homogeneous
- Input distributed database
- Runs: nondeterministic, asynchronous, fair



# Consistency

**Output** of run: union of local outputs

**Consistent** transducer network:



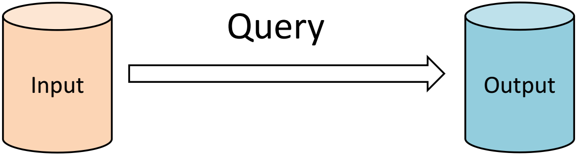
Transducer $\Pi$ **distributedly computes** query $\mathcal{Q}$ if

*all transducer networks for $\Pi$ express $\mathcal{Q}$*

# Consistency

**Output** of run: union of local outputs

**Consistent** transducer network:



Transducer $\Pi$ **distributedly computes** query $\mathcal{Q}$ if

*all transducer networks for $\Pi$ express $\mathcal{Q}$*

# Coordination-freeness

**Coordination**

distributed consensus, communication, waiting

Transducer network is **coordination-free** if

*for all input instances I, the output can already be produced without communication on some partition of I*

Transducer $\Pi$ is **coordination-free** if

*all transducer networks for $\Pi$ are coordination-free*

# CALM Conjecture

*A program has an eventually consistent, coordination-free execution strategy if and only if it is expressible in (monotonic) Datalog.*

$\Downarrow$

*A query can be distributedly computed by a coordination-free transducer if and only if it is expressible in (monotonic) Datalog.*

# CALM Conjecture

*A program has an eventually consistent, coordination-free execution strategy if and only if it is expressible in (monotonic) Datalog.*

$$\Downarrow$$

*A query can be distributedly computed by a coordination-free transducer if and only if it is expressible in (monotonic) Datalog.*

**If direction: holds**

All monotone queries computable by coordination-free transducer

**Only-if direction: fails**

Monotone queries outside Datalog (Afrati et al. 1995)

# CALM Property

The following are equivalent for any query $\mathcal{Q}$:

1. $\mathcal{Q}$ is **monotone**
2. $\mathcal{Q}$ can be distributedly computed by a **coordination-free** transducer

# CALM Property

The following are equivalent for any query $\mathcal{Q}$:

1. $\mathcal{Q}$ is **monotone**
2. $\mathcal{Q}$ can be distributedly computed by a **coordination-free** transducer
3. $\mathcal{Q}$ can be distributedly computed by a transducer that is given no node identifiers (obliviousness)

# Directions for Further Work

**Quantify coordination**

**Data initialization strategies**
reduce need for coordination

Thank you