technische universität dortmund

SFB 876 Providing Information by Resource-Constrained Data Analysis

Project A2

Algorithmic aspects of learning methods in embedded systems

Prof. Dr. Christian Sohler, Prof. Dr. Jens Teubner

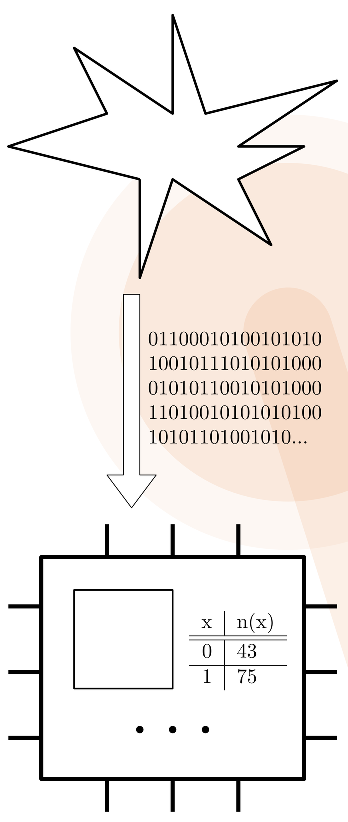
Problem

Streaming Algorithms

Often in continuous settings

- Big Data (Volume, Velocity)
- Little ressources (Memory, CPU)

Maintain up-to-date model



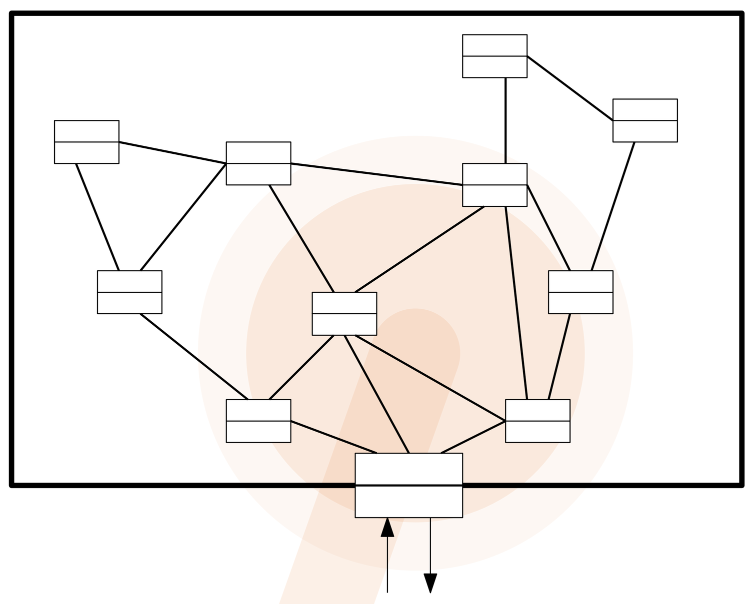
A1 B3 C3 C5

Distributed Algorithms

Often in larger amounts

- Big Data (Volume, Variety)
- No centrality

Establish central model



A4 B3 C4

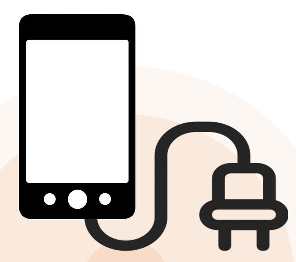
Energy Consumption

Power consumption as most important design constraint for computer systems

- Embedded systems and servers

Embedded systems

- Battery life
- Ubiquitous computing (sensors)



Servers / chip design

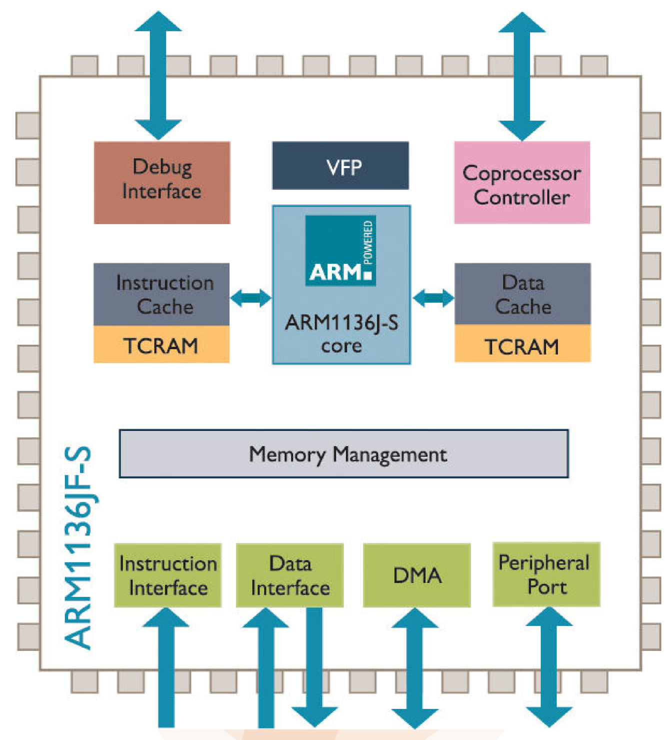
- High transistor densities cause **excessive heat**

→ **Specialised circuits** (data-parallel units, SFUs)

→ Power-off parts of the chip (**dark silicon**)

A4 C5

Computation and Communication Architecture



Smartphone processor architecture

- Memory bandwidth
- WiFi / network bandwidth
- Cache utilisation
- Data ↔ Memory

A3 B4 C3

Planned Research

Merge & Reduce Paradigm

Applicable in streaming and distributed settings

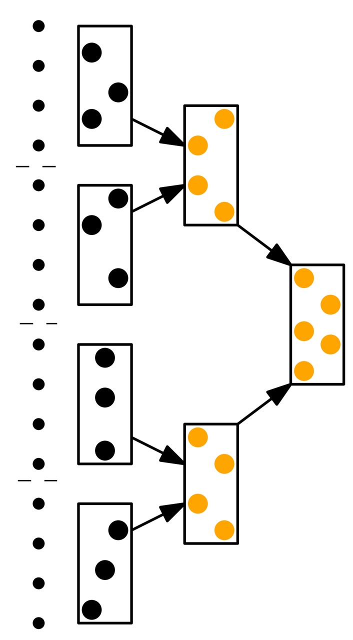
- Use of M&R dependent on problem, not input
- E.g. k-median, k-means

(Coresets for) logistic regression

- Rules out strongly sublinear space → no streaming
- **Strategy:** Make reasonable assumptions on the input

(Coresets in) distributed settings

- Input may (need to) have certain properties
- Subsets (on nodes) may not
  → Establish properties by exchanging points?



Efficient Use of Communication Resources

Balance demands for communication resources

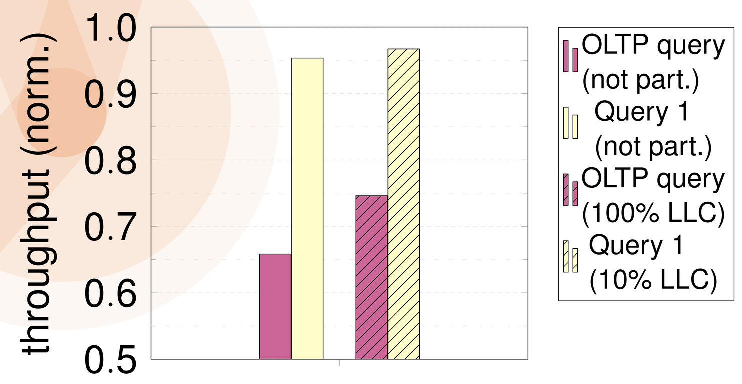
- Avoid spending energy for underutilised resources (e.g. frequency scaling)
- Devise resources only when there is a gain (E.g. can the algorithm benefit from higher bandwidth?)
- Lower execution times (static power)

Example: Cache partitioning

[Noll et al., ICDE'18]:

- Partition cache for concurrent workloads
- Avoid mututal negative effects between cache polluters and cache beneficiaries

→ **Improve cache utilisation**



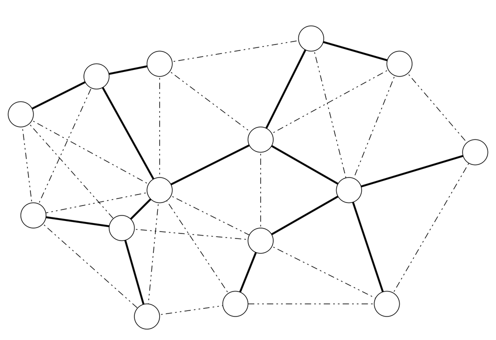
Partitioning reduces side-effects

How can we improve energy-efficiency with regard to different aspects of communication architectures (e.g. network, data placement) ?

Distributed Algorithms and Energy Consumption

Find efficient data summary trees in networks (e.g. WSNs)

- Various spanning tree and clustering algorithms available
- Empirical results, no (theoretical) energy model so far



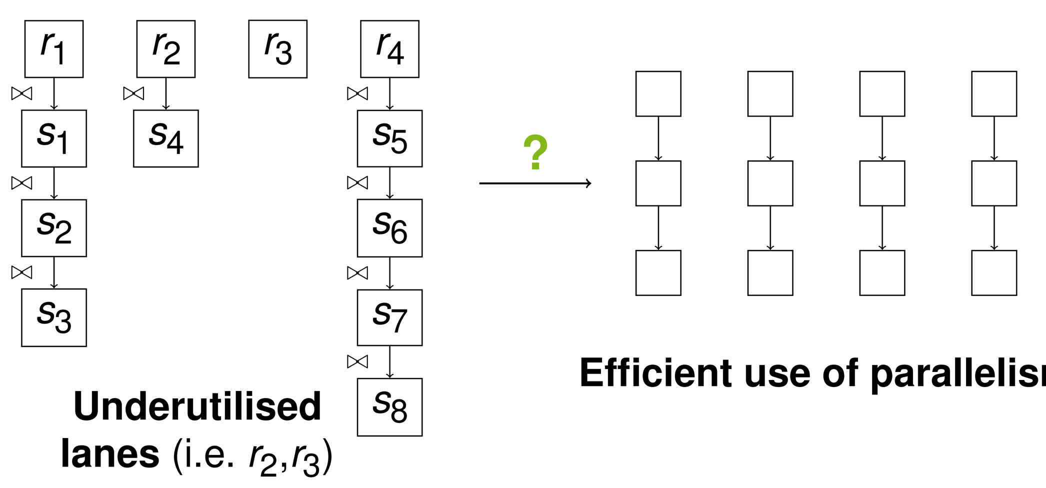
Strategy: Use task graph scheduling techniques

- Task graph ⇝ Merge & Reduce tree
- Costs measure from Bingham and Greenstreet 2008: $ET^{\alpha} = dist^{\alpha+1}$

Data-parallel Computation Architectures

Skewed distributions cause **control flow divergence** → inefficiency

Example: join skew for $R \bowtie S$ with 4 parallel lanes



Underutilised lanes (i.e. r_2,r_3)

Efficient use of parallelism

Streaming Algorithms and Computation Architecture

Analyse algorithms **adaptability** in ressource utilisation/demand

- Cache size (constants in $\mathcal{O}$-notation) → Cache allocation
- Single/special purpose processing units

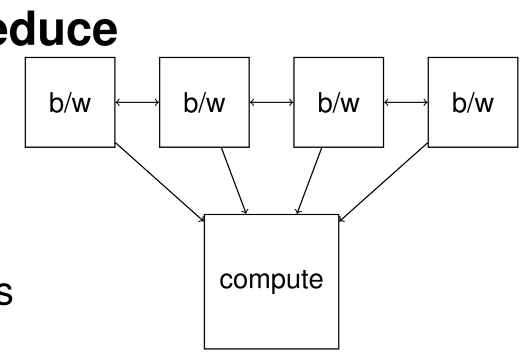
Communication Architectures for Merge & Reduce

- **Resource-aware architectures**

  Merge-phase: Communication/bandwidth hungry

  Reduce-phase: Compute hungry

  Strategy: Devise architecture along resource demands



Lehrstuhl Informatik 2
Effiziente Algorithmen und Komplexitätstheorie

Lehrstuhl Informatik 6
Datenbanken und Informationssysteme