# X-Device Query Processing by Bitwise Distribution

Holger Pirk
Stefan Manegold

Thibault Sellam
Martin Kersten

**CWI, Amsterdam, Netherlands**

CWI

monetdb

# The Problem

CWI

monetdb

# The Data

CWI

monetdb

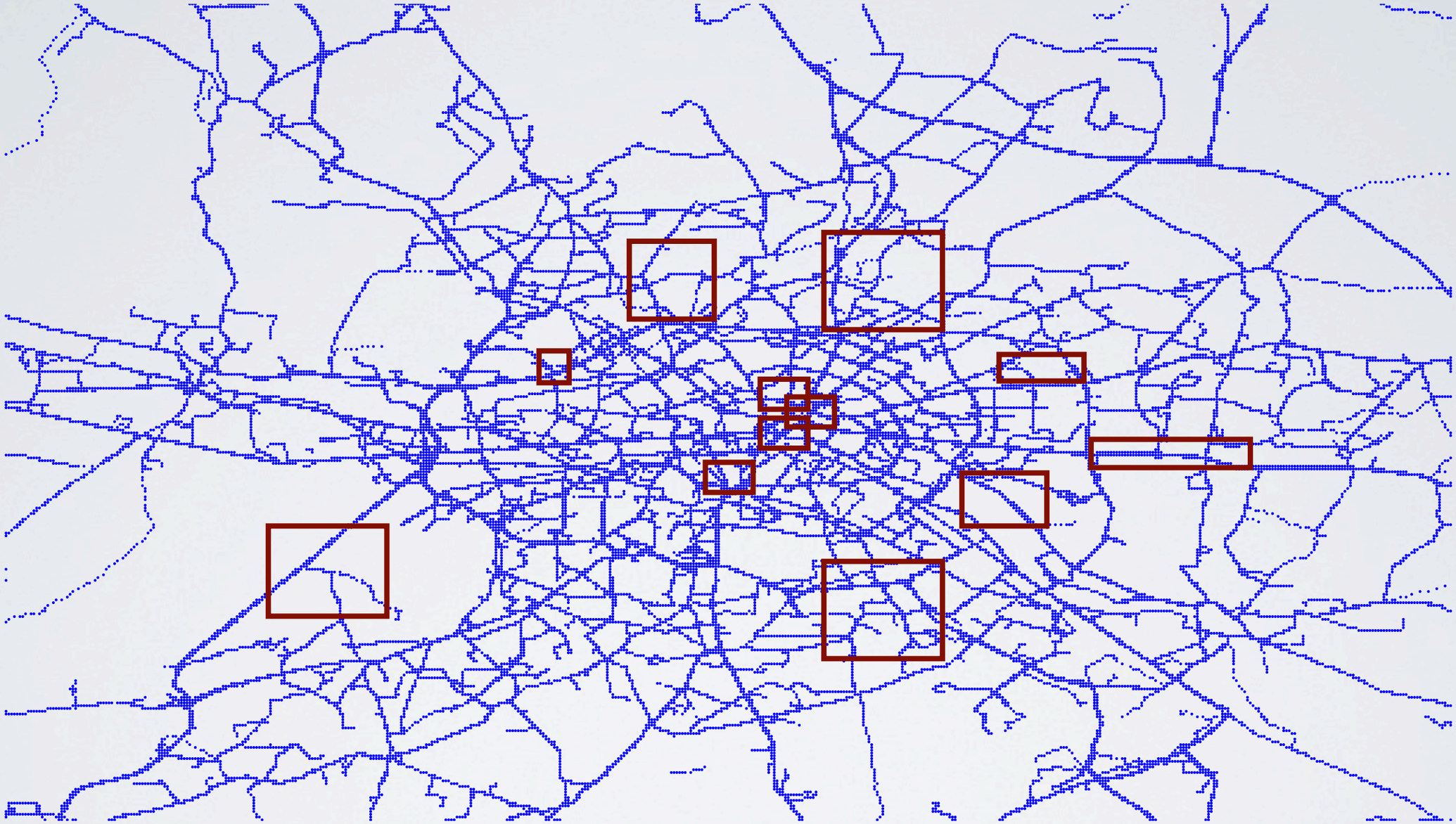
# The Data

- A moving objects database of vehicles in NW Europe
  - 240M Tuples (lat int, long int, trip bigint, timestamp bigint)
- Applications like traffic monitoring, forecasting or planing

CWI

monetdb

# The Queries



CWI

monetdb

# The Queries

- Lot's of spatial range queries:

  select * where x between $x_{low}$ and $x_{high}$ and

  y between $y_{low}$ and $y_{high}$

- Focus on Query Throughput

CWI

monetdb

# Let's kill it with Hardware

- GPUs have high bandwidth and are massively parallel

➡Many queries can be evaluated concurrently

- GPU memory is never large enough

# What can you do?

- Host-to-Device Streaming suffers from PCI bottleneck
- Horizontal Distribution leaves most data/costs on the CPU
- Sampling only yields approximate results
- Decomposed Storage is not enough

➡**The existing techniques just don't suffice**

CWI

monetdb

# Approach

CWI

monetdb

# Remember gifs through a modem connection?


CWI

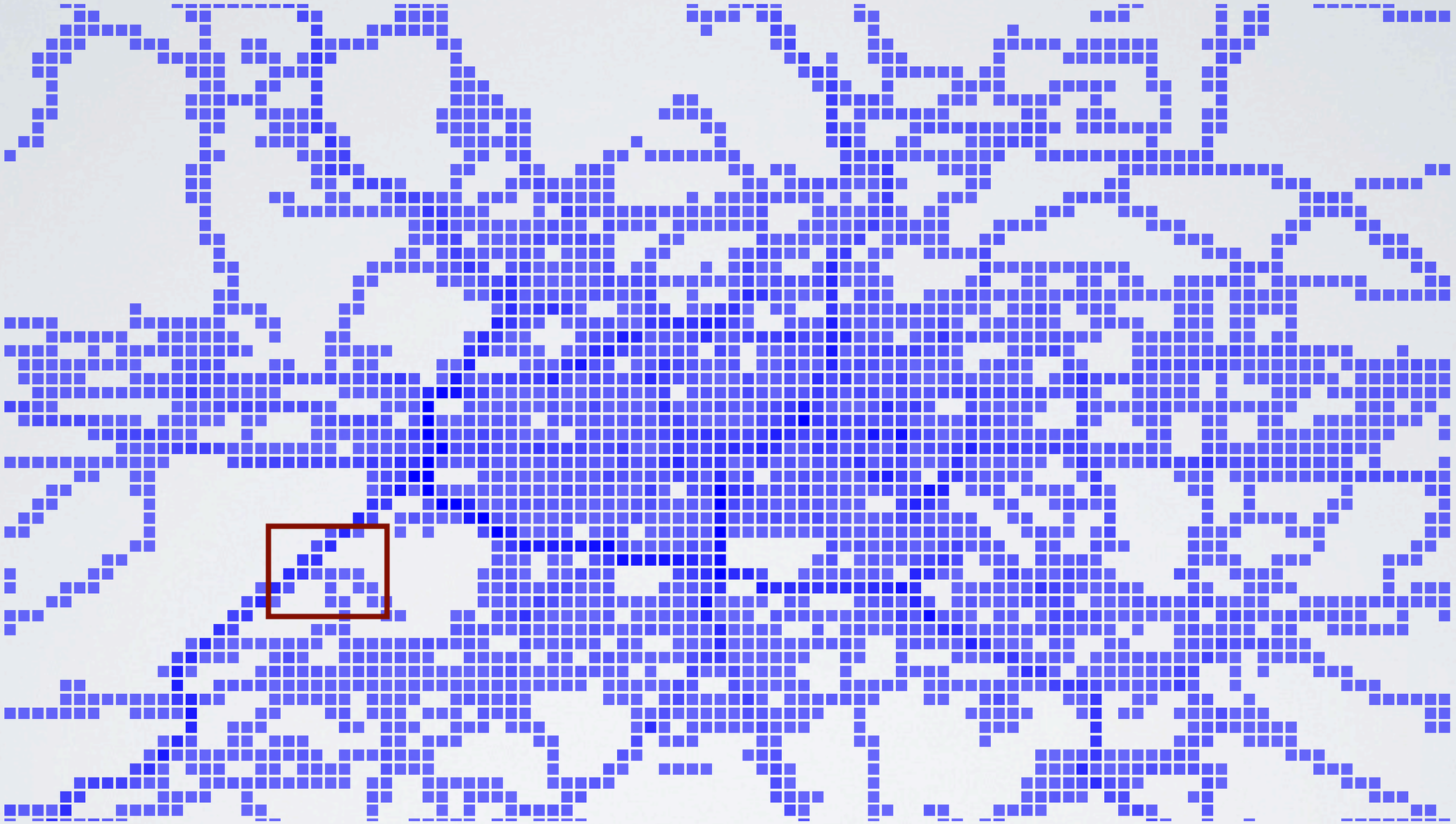
monetdb

# The Approach

CWI

monetdb

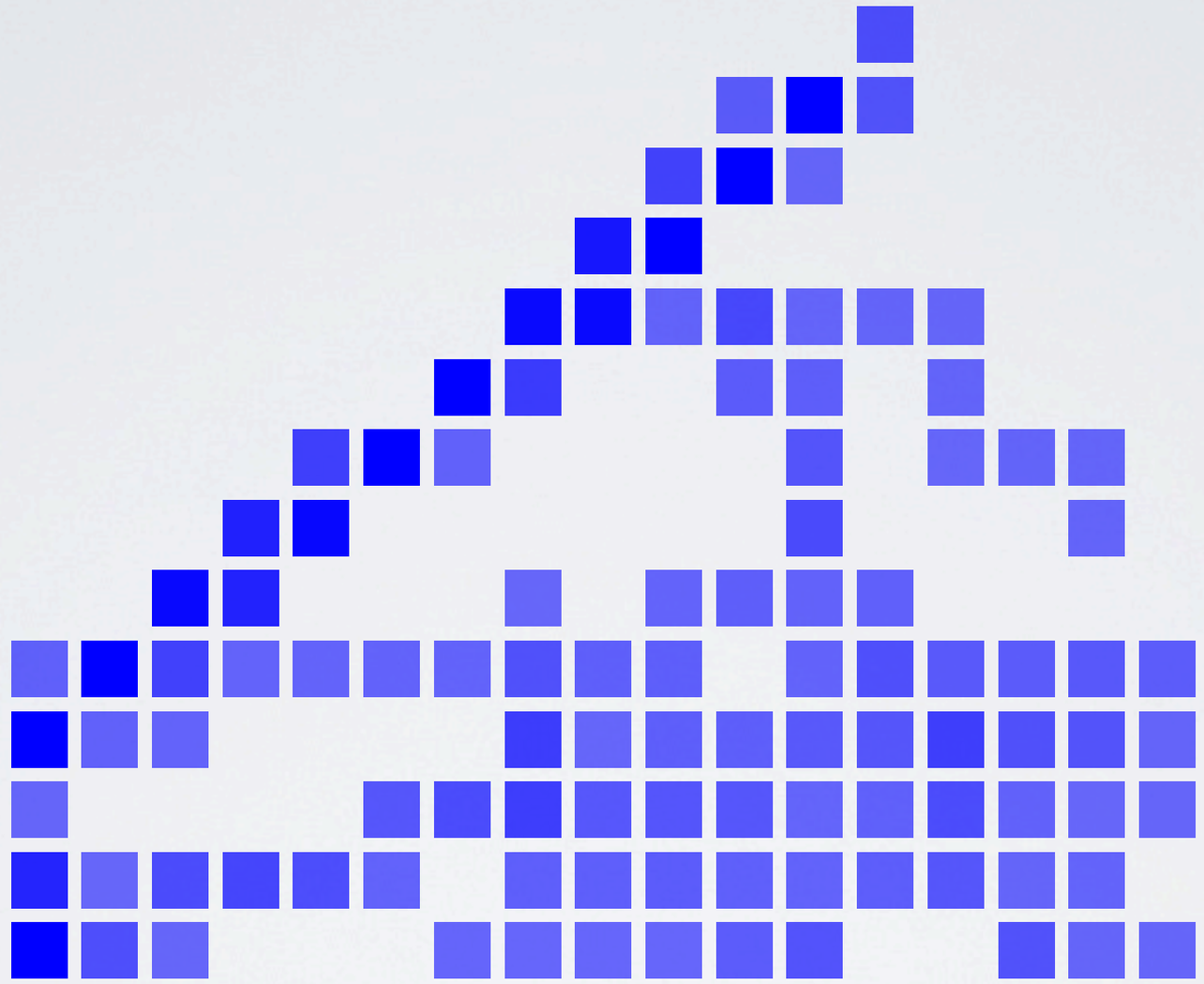
# The Approach



CWI

monetdb

# Approximate



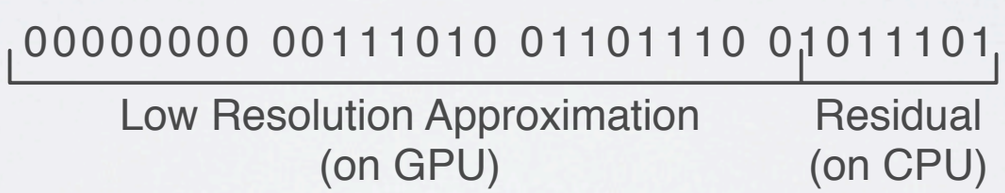
CWI

monetdb

# Refine

# Implementation

- Vertical Partitioning and Distribution
  - But with finer slices: Bitwise Decomposition & Distribution



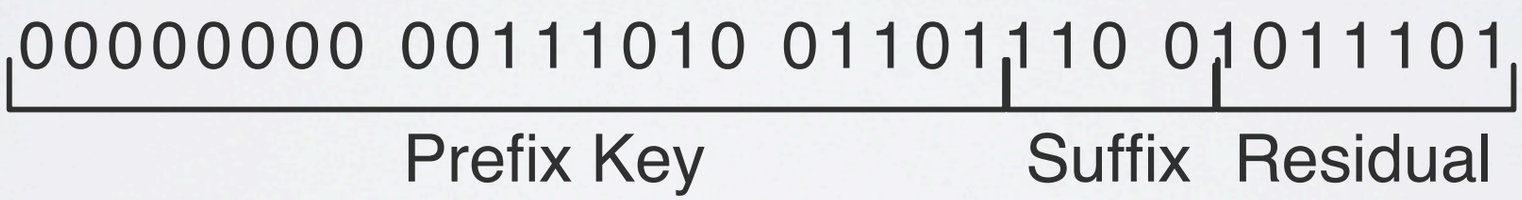

- Precision on GPU is selected appropriately (see paper)
- Low Resolution Data is well compressible

CWI

monetdb

# Compression

- Has to be Lightweight and parallel decompression friendly:
  - Prefix Compression

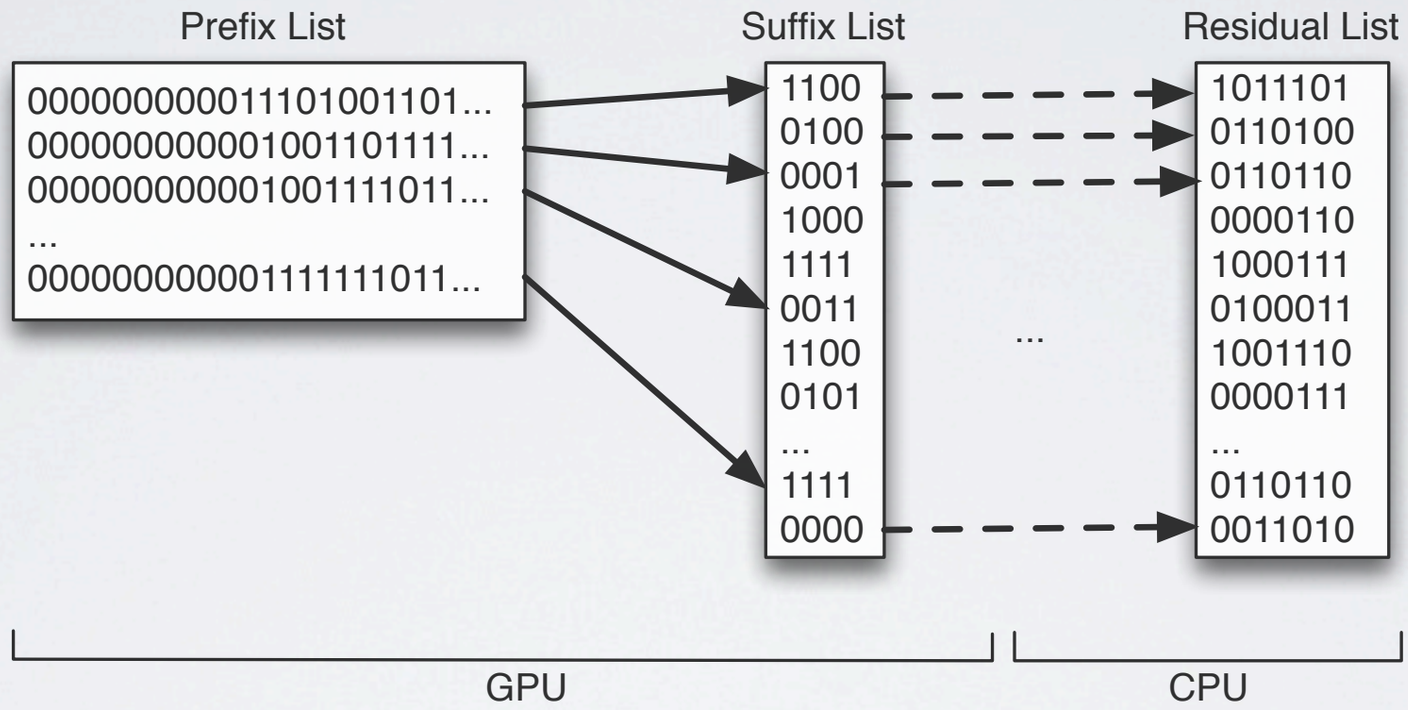



- Values are split into:
- Spatial Locality helps compression

CWI

monetdb

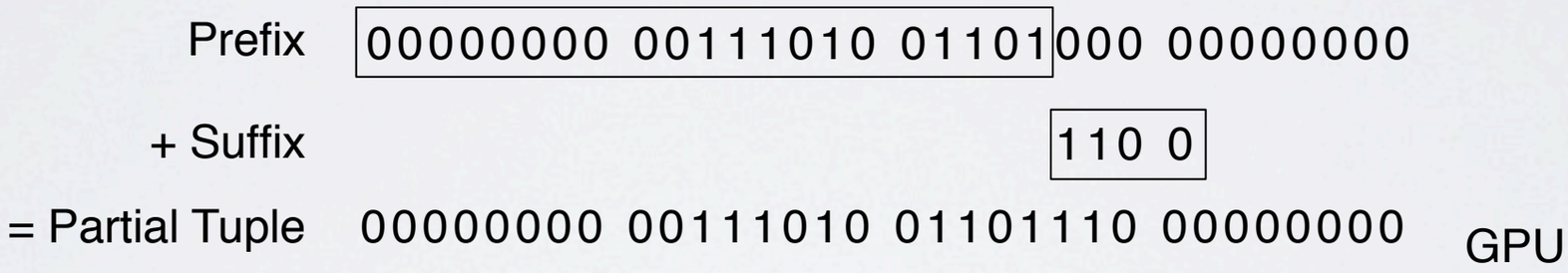
# Compression




- Radix-Clustering clustering improves compression further

# Query Evaluation

- Queries are evaluated in phases, tuples reconstructed lazily
- Every device does it's best with the available data

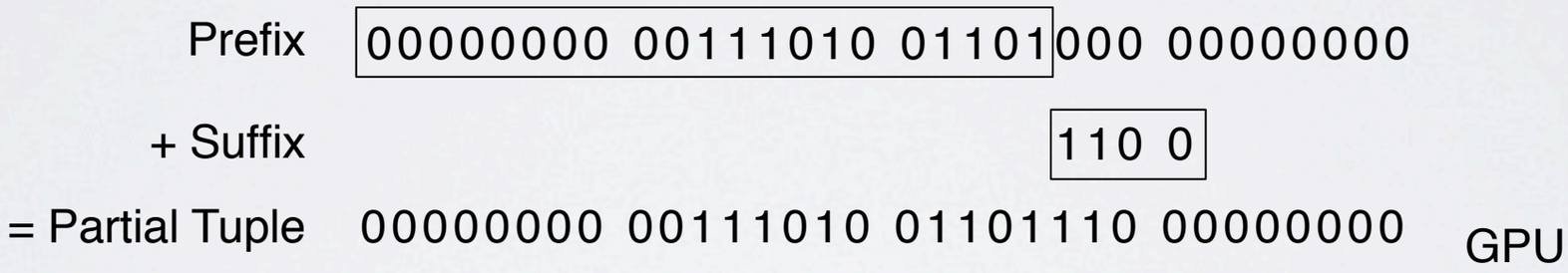


# GPU-Phase

- Geared towards throughput
- Parallelized over Queries and Prefixes
- Prefix-Clusters are skipped
  - This may cause thread divergence
- Output-Regions are overallocated

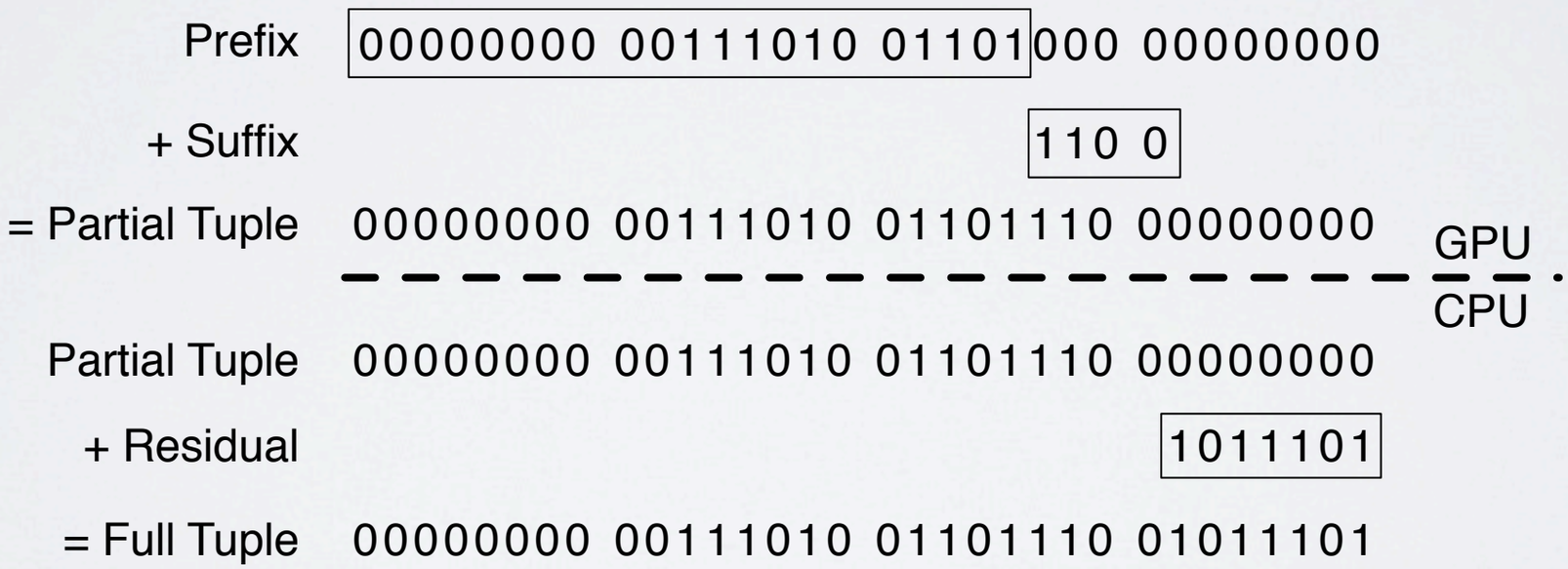
# Query Evaluation

- Queries are evaluated in phases, tuples reconstructed lazily
- Every device does it's best with the available data



# Query Evaluation

- Queries are evaluated in phases, tuples reconstructed lazily
- Every device does it's best with the available data




CWI

monetdb

# CPU-Phase

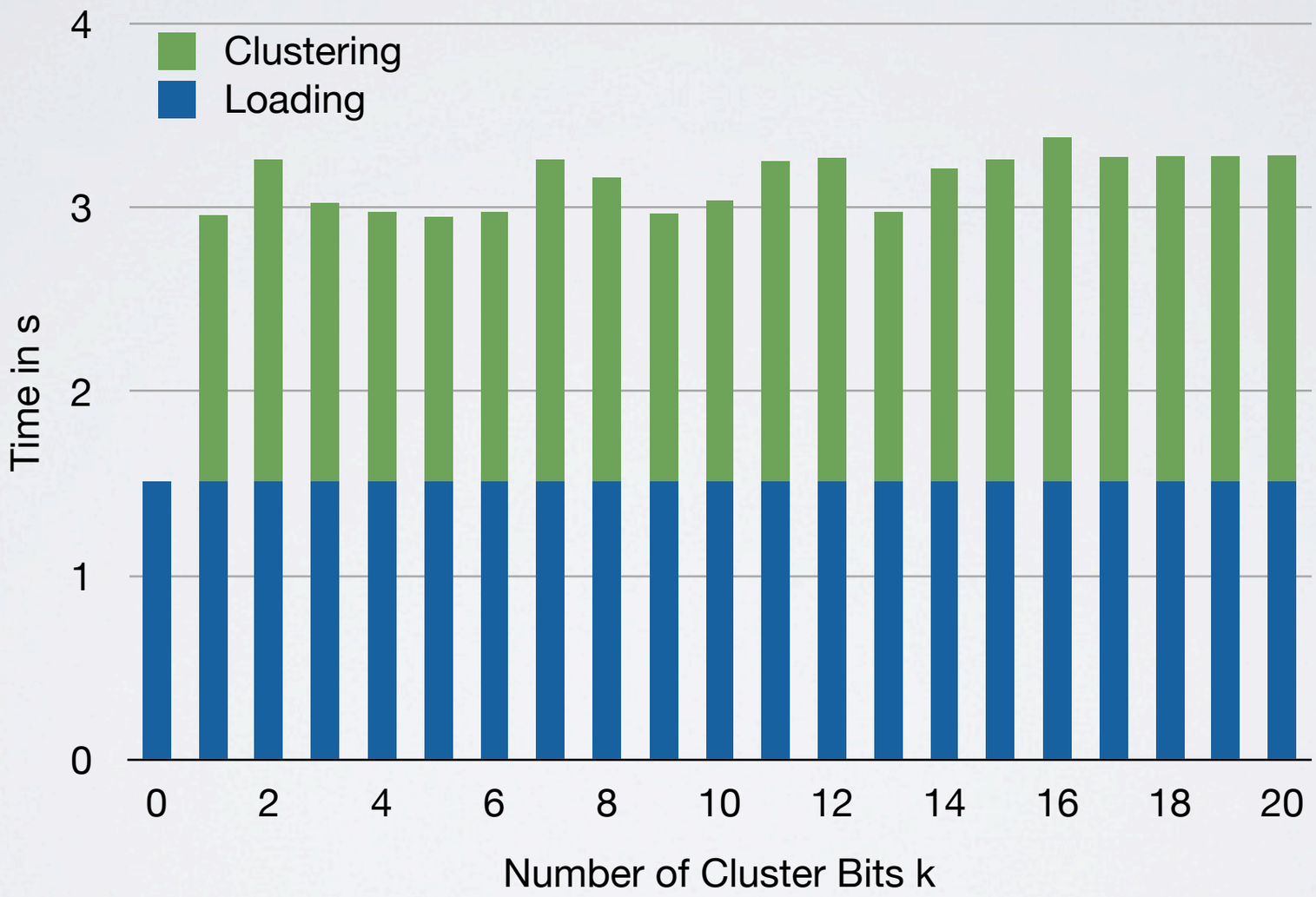
- Approximation may include false positives
- Post-filtering is done on the CPU
- **But**: they have to be transferred to the CPU
- We found around 25% false positives in the first phase

# Evaluation

CWI

monetdb

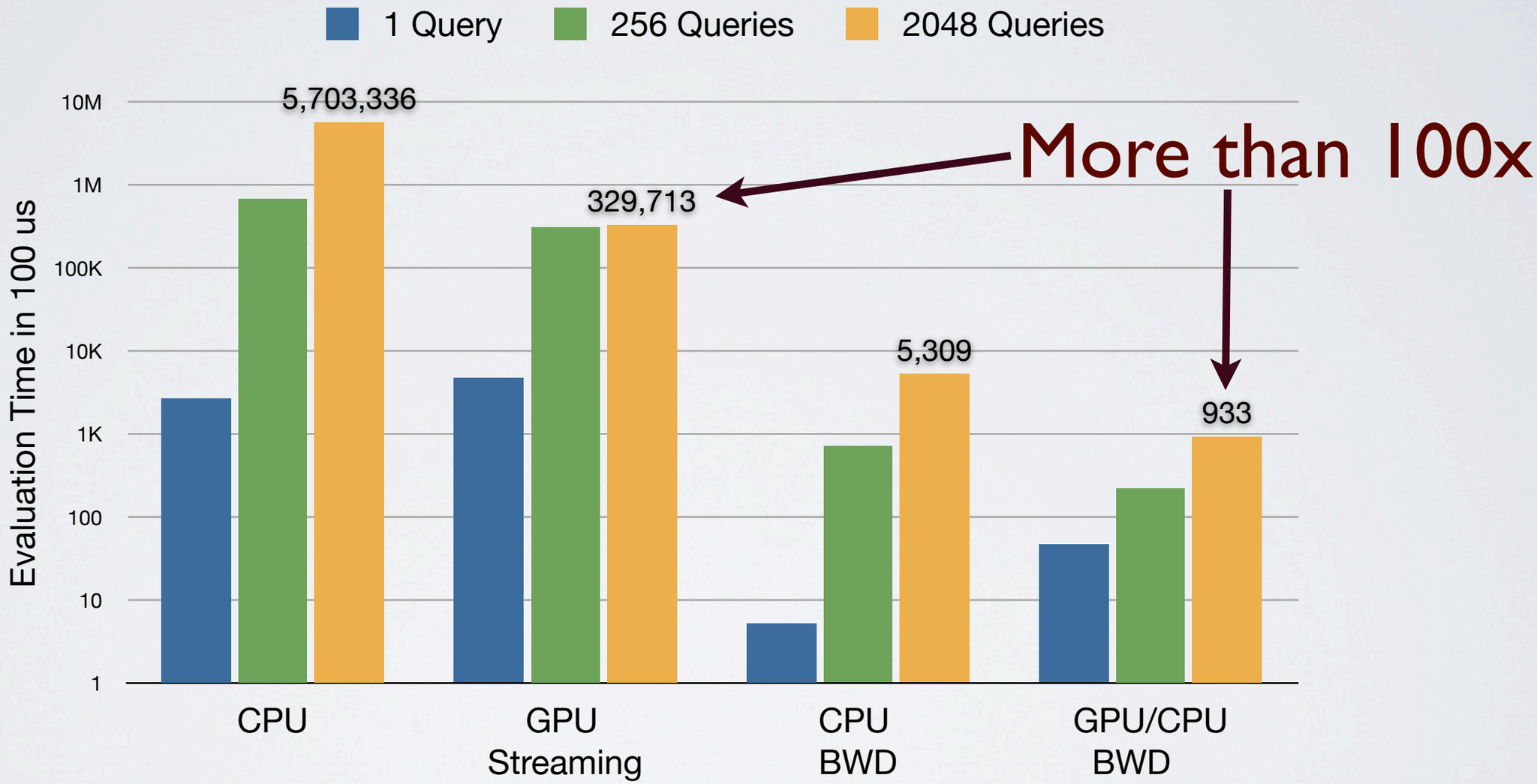
# Loading Performance




CWI

monetdb

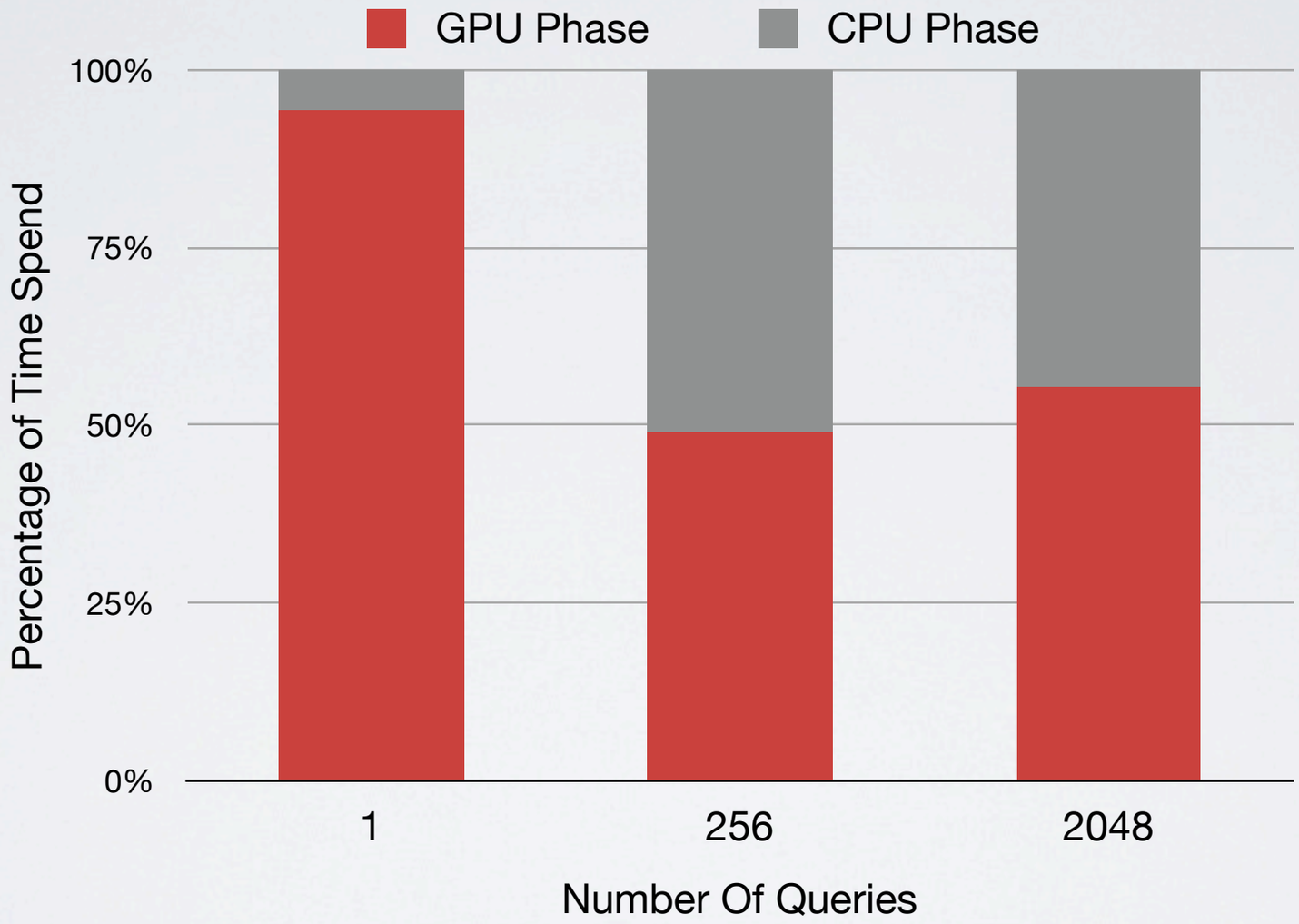
# Query Performance




CWI

monetdb

# Load Balancing




CWI

monetdb

# Conclusions and Outlook

CWI

monetdb

# Conclusion

- **Bitwise Decomposition** offers high query throughput on devices with different characteristics like GPUs and CPUs
- Largely targeted at scans
  - Doesn't really compete with indices in latency

CWI

monetdb

# Outlook

- Other X-Device Setups:
  - SSD-HDD *(remember the Keynote?)*
  - Mobile - Server
  - GPU - CPU - SSD - HDD - Server
- Other Applications:
  - Relational Queries (Joins, Aggregations, ...)
- Other Data: Categorical, Float, ...

CWI

monetdb

# Thank you !

Visit us at the poster session on Wednesday 2pm

CWI

monetdb

# Questions?

Visit us at the poster session on Wednesday 2pm

CWI

monetdb