Holistic Indexing in Main-memory Column-stores

Eleni Petraki
CWI

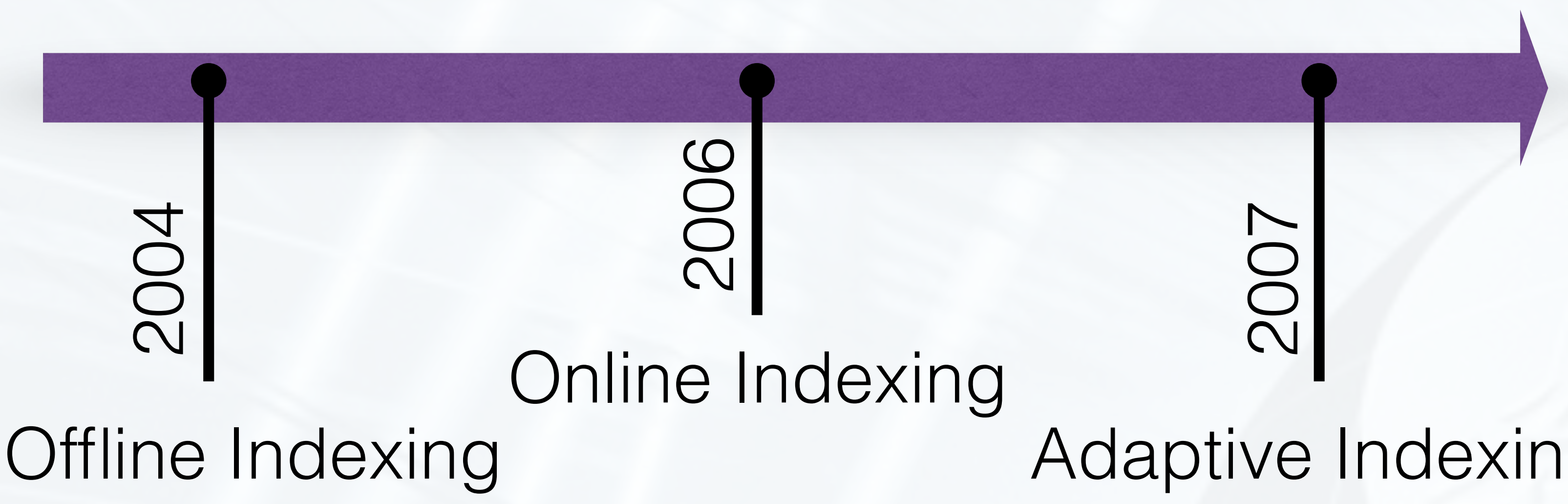
Stratos Idreos
Harvard University

Stefan Manegold
CWI

1. Automatic Indexing is Complex

How can we exploit **idle CPU resources** efficiently to better prepare the **physical design** for fully dynamic and exploratory workloads?

2. State of the Art



2004 Offline Indexing

2006 Online Indexing

2007 Adaptive Indexing

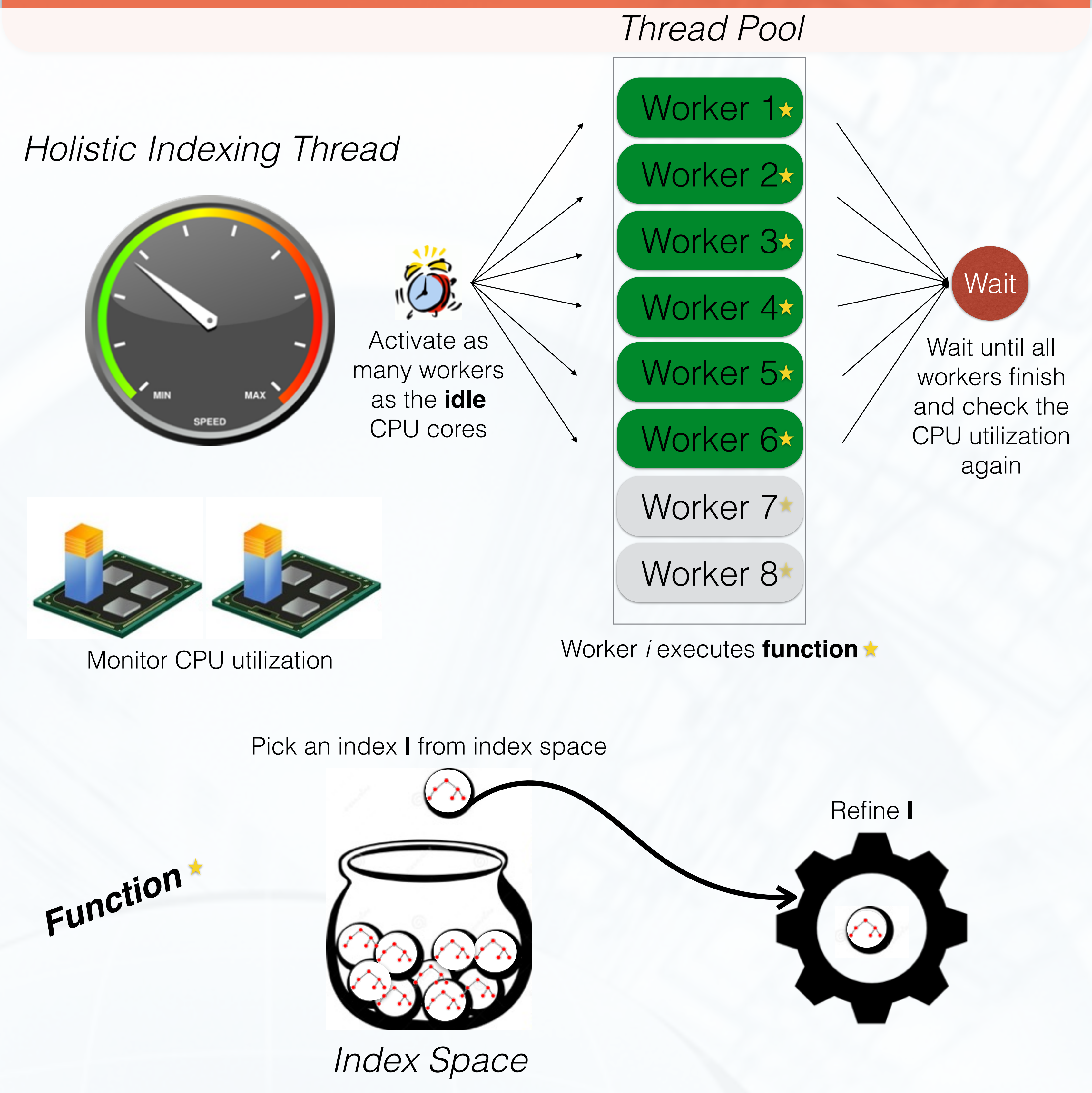
Indexing	Statistical Analysis	IR *before* QP	IR *during* QP	Index	Updates Cost	Workload
Offline	√	√	x	full	high	static
Online	√	x	√	full	high	dynamic
Adaptive	x	x	x	partial	low	dynamic
Holistic	√	√	√	partial	low	dynamic

*IR: Exploitation of Idle Resources
*QP: Query Processing

3. Desirable Properties

- ✦ ***Continuous*** workload monitoring and analysis
- ✦ ***Instant*** adjustment to workload changes
- ✦ Exploitation of ***idle CPU*** resources
- ✦ ***Lightweight*** indexing
- ✦ ***No*** external ***tool*** or ***human*** administration
- ✦ Indexing ***embedded*** in the database kernel

4. Holistic Indexing



Thread Pool

Holistic Indexing Thread

Activate as many workers as the **idle** CPU cores

Worker 1★
Worker 2★
Worker 3★
Worker 4★
Worker 5★
Worker 6★
Worker 7★
Worker 8★

Wait

Wait until all workers finish and check the CPU utilization again

Monitor CPU utilization

Worker *i* executes **function**★

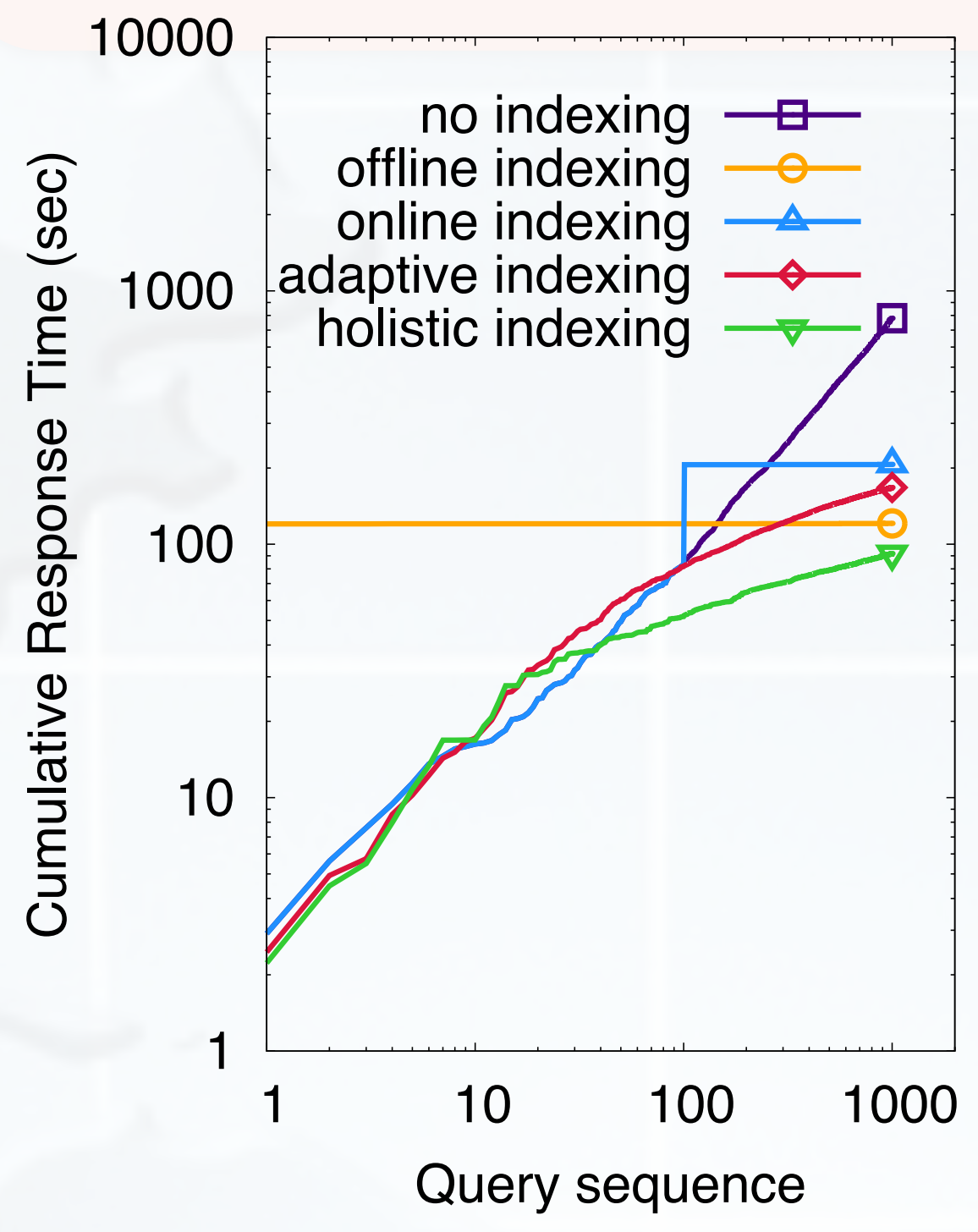
Pick an index **I** from index space

Refine **I**

Function★

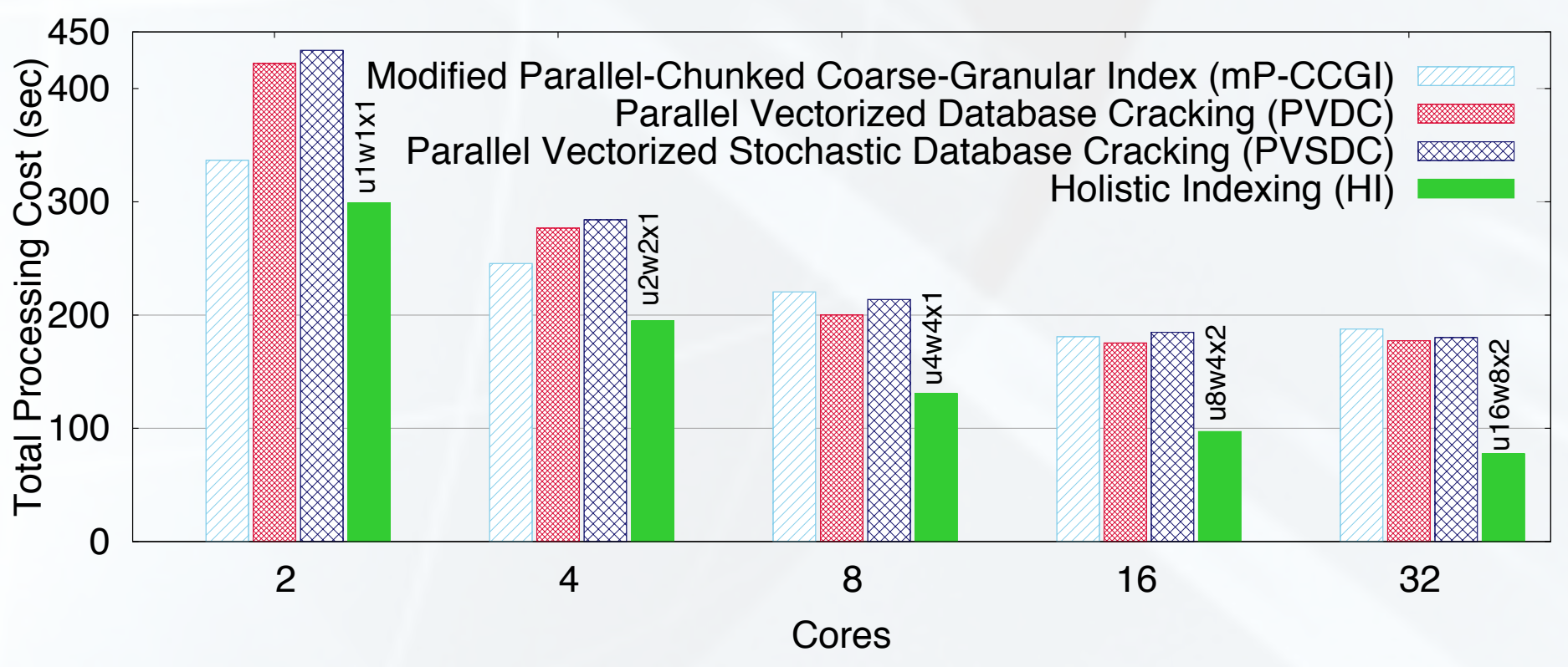
Index Space

5. Holistic Indexing Utilizes Idle CPU Resouces Efficiently



Holistic indexing improves performance by utilizing idle CPU resources more efficiently than other indexing approaches.

Holistic indexing utilizes available resources more efficiently than state of the art parallel adaptive indexing approaches.



6. Future Work

Row-stores, hybrids
Alternate indexes
Energy efficiency