



JOHANNES GUTENBERG  
UNIVERSITÄT MAINZ

# MetaCache-GPU: Ultra-Fast Metagenomic Classification

Robin Kobus<sup>1</sup>, A. Müller<sup>1</sup>, D. Jünger<sup>1</sup>, C. Hundt<sup>2</sup>, B. Schmidt<sup>1</sup>

<sup>1</sup>Institute for Computer Science Johannes Gutenberg University Mainz, Germany

<sup>2</sup>NVIDIA AI Technology Center Luxembourg, Luxembourg

ICPP'21, Chicago (Argonne National Lab), Illinois, USA

- Background
- MetaCache-GPU Workflow
- Performance Evaluation
- Conclusion

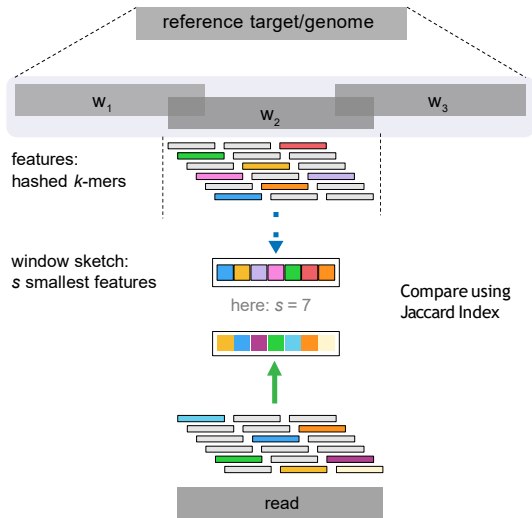
## Background – Metagenomic Classification

- High-throughput sequencers produce massive amounts of **short DNA reads**
- **Metagenomic Classification:** Determine most likely origin of each read in set of reference genomes
- Common strategy:
  - **Store and index** reference genome sequences as sets of  $k$ -length substrings (*k-mers*)
  - **Query for  $k$ -mers** in a read to find exact matches
  - Evaluate matches to **classify read**

# Background – MetaCache Basics

## MetaCache basic concepts:

- Windowing
- Minhashing
- Hash Table
- Jaccard Index



# Background – Why GPUs

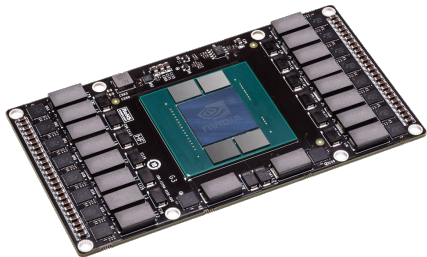
## CPUs

- Multi-core (<100)
- Memory size > 1 TB
- Memory bandwidth < 100 GB/s



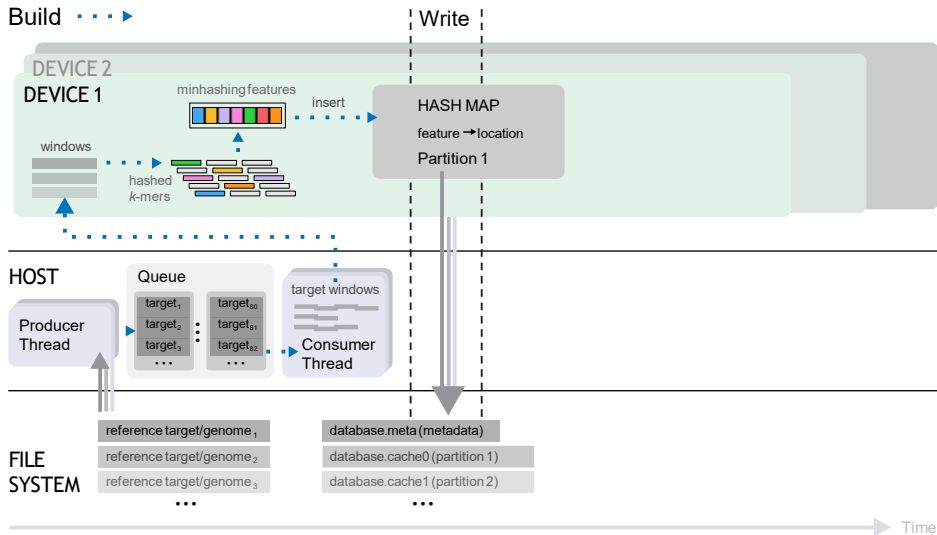
## GPUs

- Many-core (> 1000)
- Memory size up to 80 GB
- Memory bandwidth up to 2 TB/s

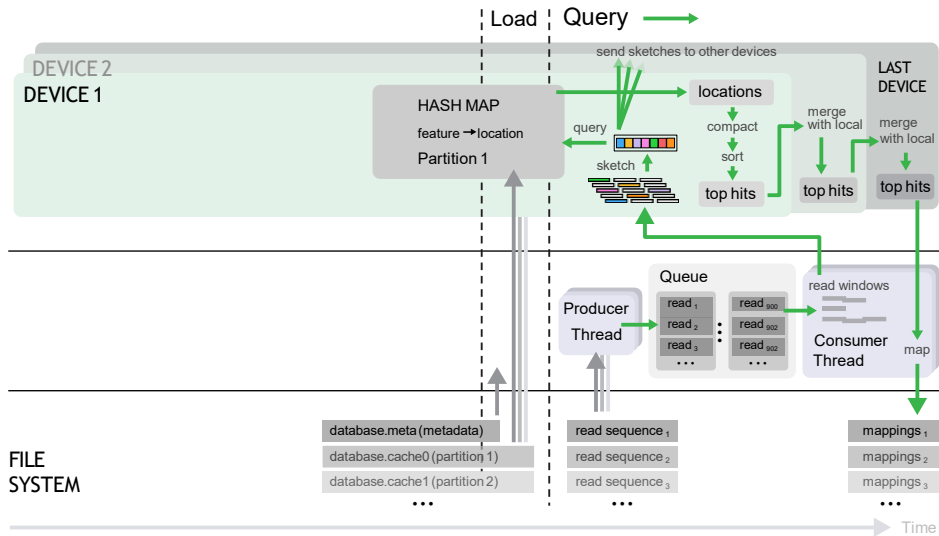


- **Novel multi-value hash table** for construction and querying of  $k$ -mer indices.
- **GPU-based minhashing and top candidate generation** for data-parallel read classification.
- **Index distribution across multiple GPUs** to support larger reference indices.
- **In-memory index construction** that allows for on-the-fly classification pipelines avoiding intermediate disk I/O.

# MetaCache-GPU – Build

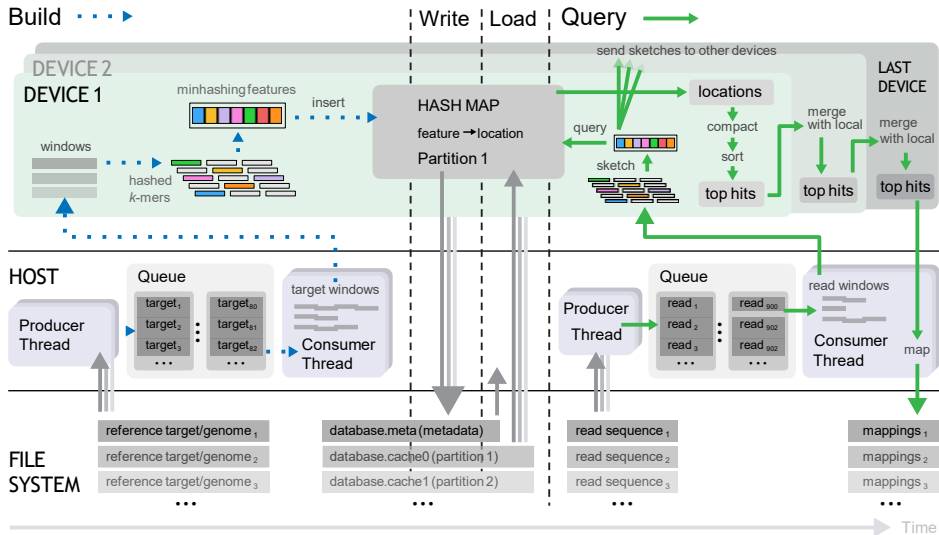


# MetaCache-GPU – Query

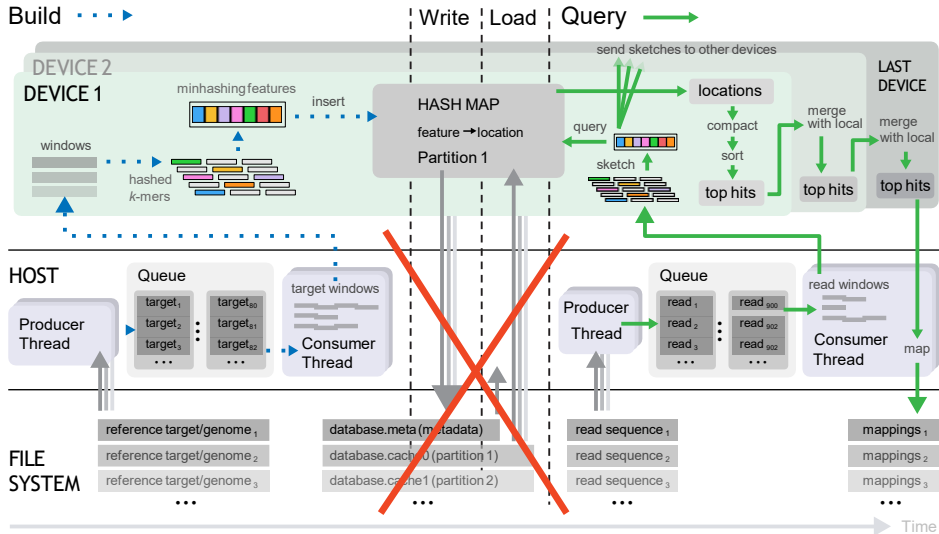




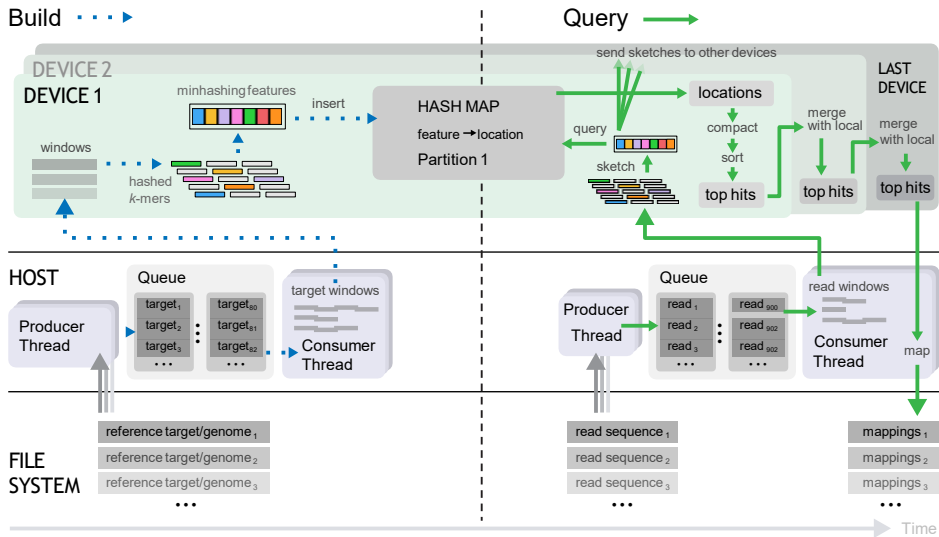
# MetaCache-GPU – Whole Pipeline



# MetaCache-GPU – Avoid Disk I/O

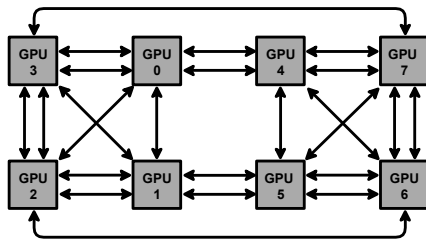


# MetaCache-GPU – On-The-Fly



## DGX-1 System:

- **CPU:** 2x Intel Xeon E5-2698 v4 @2.20 GHz
- **RAM:** 512 GB DDR4 RAM
- **GPU:** 8x NVIDIA Tesla V100 with 32 GB HBM2
- **Storage:** 4x 1.92 TB SSD RAID 0



## Performance Evaluation – Datasets

Reference genome sets  
used for databases:

Database	Species	Size on disk
RefSeq 202	15,461	74 GB
All-Food-Seq 31	31	77 GB
AFS 31 + RefSeq 202	15,492	151 GB

Metagenomic read  
datasets:

Dataset	Format	Sequences	Avg. Len.
HiSeq	FASTA single	10,000,000	92
MiSeq	FASTA single	10,000,000	157
KAL_D	FASTQ paired	26,114,376	101

## Performance Evaluation – Build

Method	Build time	Total time	DB size	RAM
RefSeq 202 database:				
Kraken2	-	72 min	40 GB	46 GB
MC CPU	67 min	69 min	51 GB	71 GB
MC 4 GPUs	10.4 s	<b>59.6 s</b>	88 GB	1 GB
MC 8 GPUs	<b>9.7 s</b>	67.0 s	97 GB	1 GB
AFS 31 + RefSeq 202 database:				
Kraken2	-	256 min	110 GB	160 GB
MC CPU	194 min	201 min	127 GB	194 GB
MC 8 GPUs	<b>42.7 s</b>	<b>3 min 31 s</b>	176 GB	30 GB

Build performance for different databases. Total time includes build time and time for writing DBs to files.

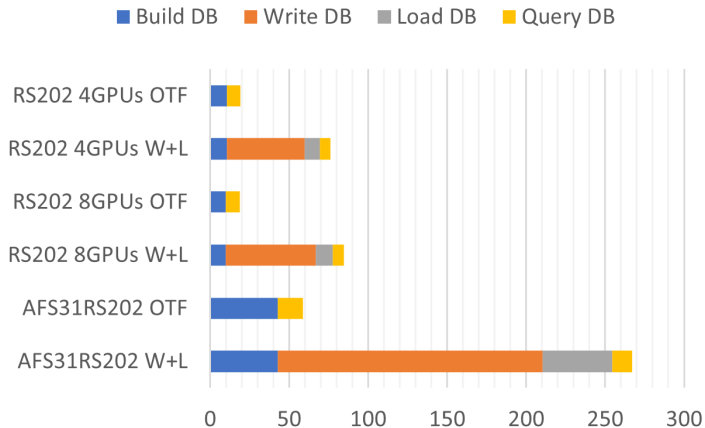
## Performance Evaluation – Query

Dataset	Method	RefSeq 202		AFS31+RefSeq202	
		Time	Speed	Time	Speed
HiSeq	Kraken2	4.6 s	130	4.1 s	147
	MC CPU	11.4 s	53	107.4 s	5.6
	MC 4 GPUs	2.1 s	292	– <sup>1</sup>	– <sup>1</sup>
	MC 8 GPUs	<b>2.0 s</b>	<b>305</b>	<b>2.0 s</b>	<b>298</b>
KAL_D	Kraken2	42.6 s	74	37.8 s	83
	MC CPU	38.9 s	81	237.0 s	13
	MC 4 GPUs	<b>6.9 s</b>	<b>454</b>	– <sup>1</sup>	– <sup>1</sup>
	MC 8 GPUs	7.2 s	435	<b>12.6 s</b>	<b>249</b>

Query performance against different databases. Query speed in million reads per minute.

<sup>1</sup>4 V100 GPUs do not provide enough memory for AFS31+RefSeq202.

# Performance Evaluation – On-The-Fly



Runtime comparison of our on-the-fly (OTF) mode to separate build and query execution (W+L).



## Performance Evaluation – Accuracy

Dataset	Method	Species		Genus	
		Prec.	Sens.	Prec.	Sens.
HiSeq	Kraken2	82.52%	58.39%	99.09%	<b>88.46%</b>
	MC CPU	<b>89.41%</b>	<b>63.68%</b>	99.20%	81.36%
	MC 4 GPUs	88.70%	62.61%	<b>99.36%</b>	82.32%
	MC 8 GPUs	88.81%	62.63%	<b>99.36%</b>	82.40%
MiSeq	Kraken2	<b>77.91%</b>	48.53%	<b>99.38%</b>	93.25%
	MC CPU	72.28%	60.67%	99.21%	93.23%
	MC 4 GPUs	73.07%	61.55%	99.37%	93.82%
	MC 8 GPUs	73.53%	<b>61.99%</b>	99.37%	<b>93.92%</b>

Classification accuracy using RefSeq 202 database.

## MetaCache-GPU

- Alignment-free method for **metagenomic read classification on CUDA-enabled GPUs**
- Ultra-Fast **parallel construction and querying** using a novel hash table structure for  $k$ -mers
- **On-the-fly mode**
  - Enables **rapidly updatable classification pipelines** to make use of the latest reference genomes or use custom reference genome sets
  - Achieves **two orders-of-magnitude speedup** compared to Kraken2 and the CPU version of MetaCache

# Future Work

- Scaling out to larger numbers of GPUs on cluster systems
- Adopt methods to related NGS tasks such as read mapping and long-read-to-long-read alignment



# Thank you.

Our implementation is publicly available at  
<https://github.com/muellan/metacache>