Accelerating the LZ-complexity algorithm

2023 IEEE Cybermatics Congress is held in Ocean Flower Island, Hainan, China, December 17-21, 2023

Joel Ratsaby & Alexander Timashkov
Ariel University

Introduction

Unstructured-data is ubiquitous ($> 80\%$ of data), hybrid (text, images, ...), not pre-processed, no fixed input dimensionality

Want Machine-Learning (ML) from *raw data* (without pre-processing, without feature extraction)

LZ-complexity can define string-distance function $d(x, y)$, where x and y can have different lengths (useful for unstructured data)

Computation of LZ-complexity is inherently sequential (serial) process, time-complexity $O\left(n^2\right)$, n length of data, impractical for ML

Introduce Parallel LZ-complexity algorithm

Implement in CUDA on GPU

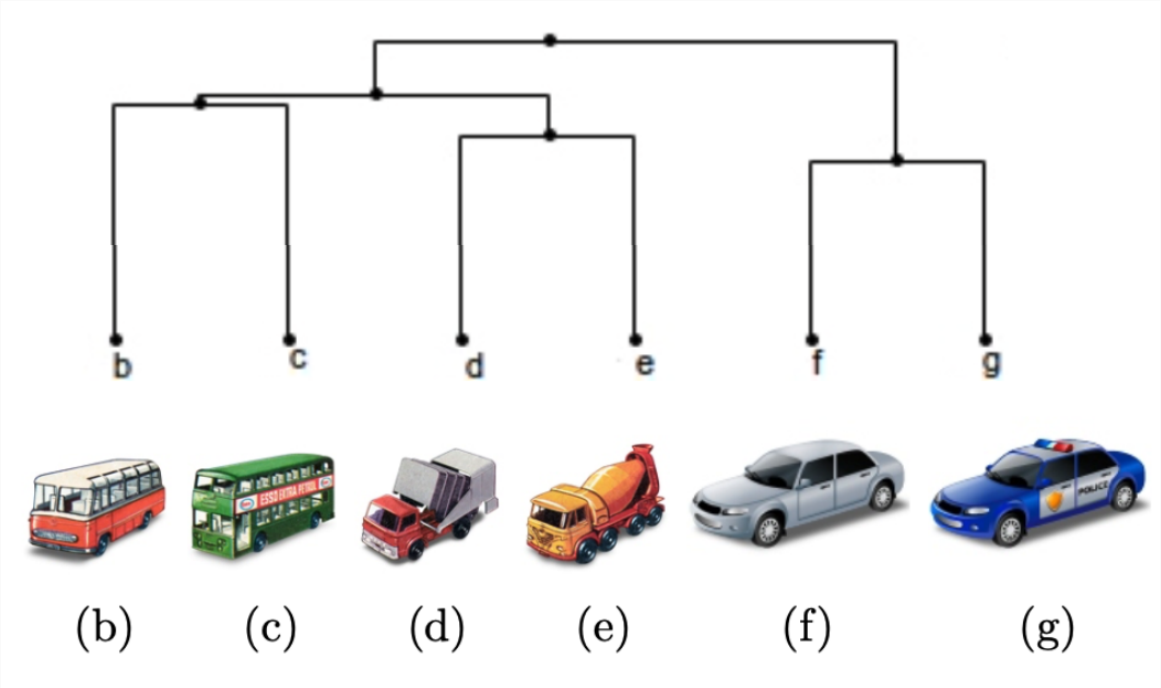
GPU-Utilization 90%

High-Speedup 150 (for 2Mb strings)

ML by LZ

Distance-matrix representation of data $[d(x_i, x_j)]_{i,j=1}^{N}$, no need for feature extraction, learn from raw data, e.g., data streams, time series, images, ...

Example LZ-distance for *clustering images* (converted to strings)



LZ-complexity

Lempel-Ziv complexity of a finite string S, basis of many LZ compression algorithms, LZ77, LZ78 , LZSS, ...

Minimal number of substrings of S that are sufficient to reproduce S via a *copy* operation

String $S = abdbbcdddddaddc$ can be reproduced from a dictionary of substrings

a	b	d	bb	c	dd	$ddda$	ddc

therefore its LZ-complexity equals 8

Serial LZ algorithm

present		↓													
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a														
search															
dictionary	a														
max															
LZ-complexity	1														

Serial LZ algorithm

present		↓													
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a														
search	↑														
dictionary	a														
max															
LZ-complexity	1														

Serial LZ algorithm

present			↓												
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b													
search															
dictionary	a	b													
max															
LZ-complexity	2														

Serial LZ algorithm

present			↓												
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b													
search	↑														
dictionary	a	b													
max															
LZ-complexity	2														

Serial LZ algorithm

present			↓												
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b													
search		↑													
dictionary	a	b													
max															
LZ-complexity	2														

Serial LZ algorithm

present				↓											
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d												
search															
dictionary	a	b	d												
max															
LZ-complexity	3														

Serial LZ algorithm

present				↓											
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d												
search	↑														
dictionary	a	b	d												
max															
LZ-complexity	3														

Serial LZ algorithm

present				↓											
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d												
search		↑													
dictionary	a	b	d												
max	1														
LZ-complexity	3														

Serial LZ algorithm

present				↓	↓										
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d												
search		↑	↑												
dictionary	a	b	d												
max	2														
LZ-complexity	3														

Serial LZ algorithm

present						↓									
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b										
search															
dictionary	a	b	d	bb											
max															
LZ-complexity	4														

Serial LZ algorithm

present						↓									
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b										
search	↑														
dictionary	a	b	d	bb											
max															
LZ-complexity	4														

Serial LZ algorithm

present						↓									
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b										
search		↑													
dictionary	a	b	d	bb											
max															
LZ-complexity	4														

Serial LZ algorithm

present						↓									
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b										
search			↑												
dictionary	a	b	d	bb											
max															
LZ-complexity	4														

Serial LZ algorithm

present						↓									
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b										
search				↑											
dictionary	a	b	d	bb											
max															
LZ-complexity	4														

Serial LZ algorithm

present						↓									
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b										
search					↑										
dictionary	a	b	d	bb											
max															
LZ-complexity	4														

Serial LZ algorithm

present							↓								
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c									
search															
dictionary	a	b	d	bb	c										
max															
LZ-complexity	5														

Serial LZ algorithm

present							↓								
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c									
search	↑														
dictionary	a	b	d	bb	c										
max															
LZ-complexity	5														

Serial LZ algorithm

present							↓								
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c									
search		↑													
dictionary	a	b	d	bb	c										
max															
LZ-complexity	5														

Serial LZ algorithm

present							↓								
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c									
search			↑												
dictionary	a	b	d	bb	c										
max	1														
LZ-complexity	5														

Serial LZ algorithm

present							↓	↓							
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c									
search			↑	↑											
dictionary	a	b	d	bb	c										
max	2														
LZ-complexity	5														

Serial LZ algorithm

present							↓								
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c									
search				↑											
dictionary	a	b	d	bb	c										
max	2														
LZ-complexity	5														

Serial LZ algorithm

present							↓								
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c									
search					↑										
dictionary	a	b	d	bb	c										
max	2														
LZ-complexity	5														

Serial LZ algorithm

present							↓								
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c									
search						↑									
dictionary	a	b	d	bb	c										
max	2														
LZ-complexity	5														

Serial LZ algorithm

present								↓							
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d							
search															
dictionary	a	b	d	bb	c	dd									
max															
LZ-complexity	6														

Serial LZ algorithm

present									↓						
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d							
search	↑														
dictionary	a	b	d	bb	c	dd									
max															
LZ-complexity	6														

Serial LZ algorithm

present									↓						
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d							
search		↑													
dictionary	a	b	d	bb	c	dd									
max															
LZ-complexity	6														

Serial LZ algorithm

present								↓							
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d							
search			↑												
dictionary	a	b	d	bb	c	dd									
max	1														
LZ-complexity	6														

Serial LZ algorithm

present									↓	↓					
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d							
search			↑	↑											
dictionary	a	b	d	bb	c	dd									
max	2														
LZ-complexity	6														

Serial LZ algorithm

present								↓							
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d							
search				↑											
dictionary	a	b	d	bb	c	dd									
max	2														
LZ-complexity	6														

Serial LZ algorithm

present									↓						
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d							
search					↑										
dictionary	a	b	d	bb	c	dd									
max	2														
LZ-complexity	6														

Serial LZ algorithm

present									↓						
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d							
search						↑									
dictionary	a	b	d	bb	c	dd									
max	2														
LZ-complexity	6														

Serial LZ algorithm

present								↓							
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d							
search							↑								
dictionary	a	b	d	bb	c	dd									
max	2														
LZ-complexity	6														

Serial LZ algorithm

present									↓	↓					
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d							
search							↑	↑							
dictionary	a	b	d	bb	c	dd									
max	2														
LZ-complexity	6														

Serial LZ algorithm

present									↓	↓	↓				
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d							
search							↑	↑	↑						
dictionary	a	b	d	bb	c	dd									
max	2														
LZ-complexity	6														

Serial LZ algorithm

present									↓	↓	↓	↓			
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d							
search							↑	↑	↑	↑					
dictionary	a	b	d	bb	c	dd									
max	4														
LZ-complexity	6														

Serial LZ algorithm

present													↓		
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d	d	d	d	a			
search															
dictionary	a	b	d	bb	c	dd	ddda								
max															
LZ-comp	7														

Serial LZ algorithm

present													↓		
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d	d	d	d	a			
search	↑														
dictionary	a	b	d	bb	c	dd	ddda								
max															
LZ-comp	7														

Serial LZ algorithm

present													↓		
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d	d	d	d	a			
search		↑													
dictionary	a	b	d	bb	c	dd	ddda								
max															
LZ-comp	7														

Serial LZ algorithm

present													↓		
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d	d	d	d	a			
search			↑												
dictionary	a	b	d	bb	c	dd	ddda								
max	1														
LZ-comp	7														

Serial LZ algorithm

present													↓	↓	
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d	d	d	d	a			
search			↑	↑											
dictionary	a	b	d	bb	c	dd	ddda								
max	2														
LZ-comp	7														

Serial LZ algorithm

present													↓		
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d	d	d	d	a			
search				↑											
dictionary	a	b	d	bb	c	dd	ddda								
max	2														
LZ-comp	7														

Serial LZ algorithm

present													↓		
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d	d	d	d	a			
search					↑										
dictionary	a	b	d	bb	c	dd	ddda								
max	2														
LZ-comp	7														

Serial LZ algorithm

present													↓		
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d	d	d	d	a			
search						↑									
dictionary	a	b	d	bb	c	dd	ddda								
max	2														
LZ-comp	7														

Serial LZ algorithm

present													↓		
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d	d	d	d	a			
search							↑								
dictionary	a	b	d	bb	c	dd	ddda								
max	2														
LZ-comp	7														

Serial LZ algorithm

present													↓	↓	
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d	d	d	d	a			
search							↑	↑							
dictionary	a	b	d	bb	c	dd	ddda								
max	2														
LZ-comp	7														

Serial LZ algorithm

present													↓	↓	↓
input string	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d	d	d	d	a			
search							↑	↑	↑						
dictionary	a	b	d	bb	c	dd	ddda								
max	3														
LZ-comp	8														

Serial LZ algorithm

present															
input	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
history	a	b	d	b	b	c	d	d	d	d	d	a	d	d	c
search															
dictionary	a	b	d	bb	c	dd	ddda	ddc							
max															
LZ-comp	8														

Parallel LZ algorithm

`While (`$present < n$`)` **{** // On CPU

Launch (on GPU) parallel kernel, with current *present* pointer ↓ to **search history** for maximal word component; each block finds (local) maximum word component (from its 1,024 *starting positions* in history)

Compute maximum of the local maxima

Increment LZ-complexity

Increment *present* pointer ↓ (by length of maximal word component), history grows accordingly

}

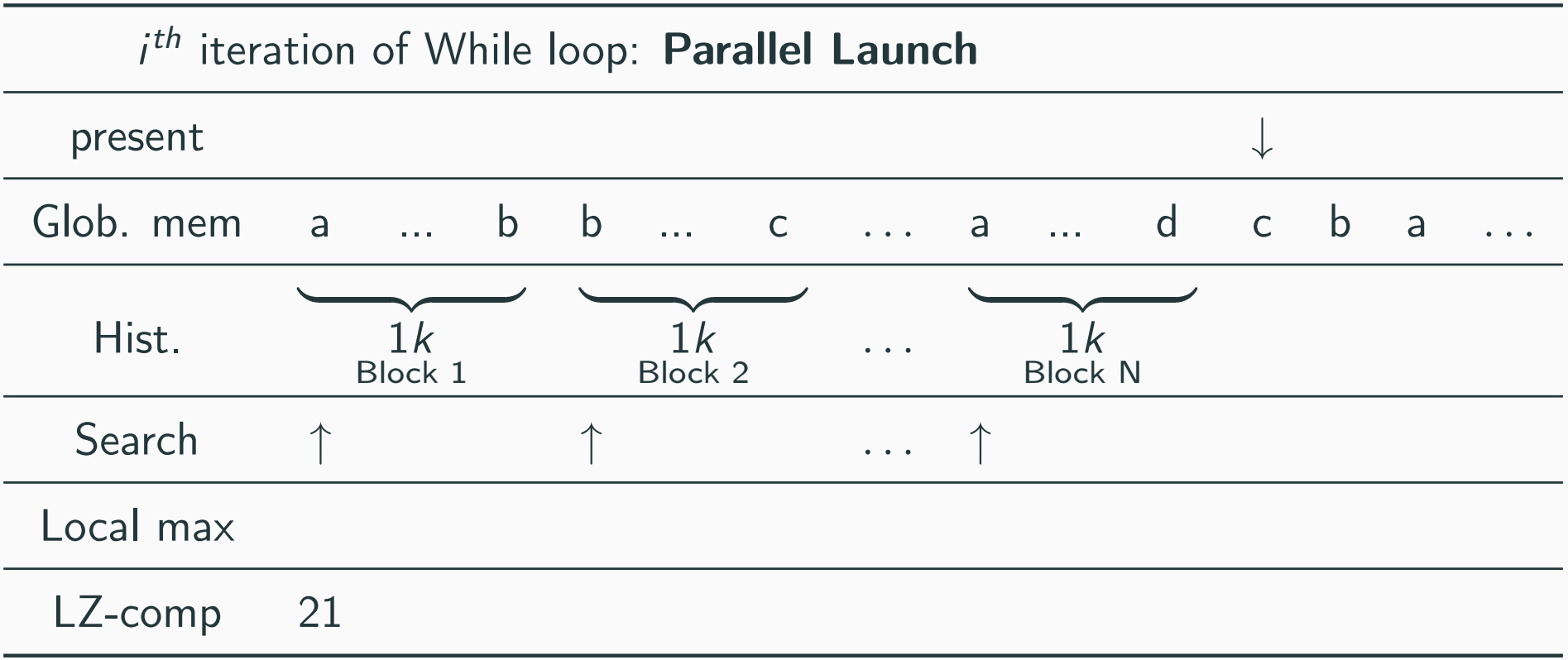
Two kernels

Kernel_global uses only global memory

Kernel_shared uses global and shared for history search

V100-GPU 80 Streaming Multiprocessors (SM), each SM has 64 cores, each core executes a thread-warp (32 threads), so at any instant, $2^{11} = 2048 = 2$ thread blocks execute. GPU has 96k of shared memory (fast)

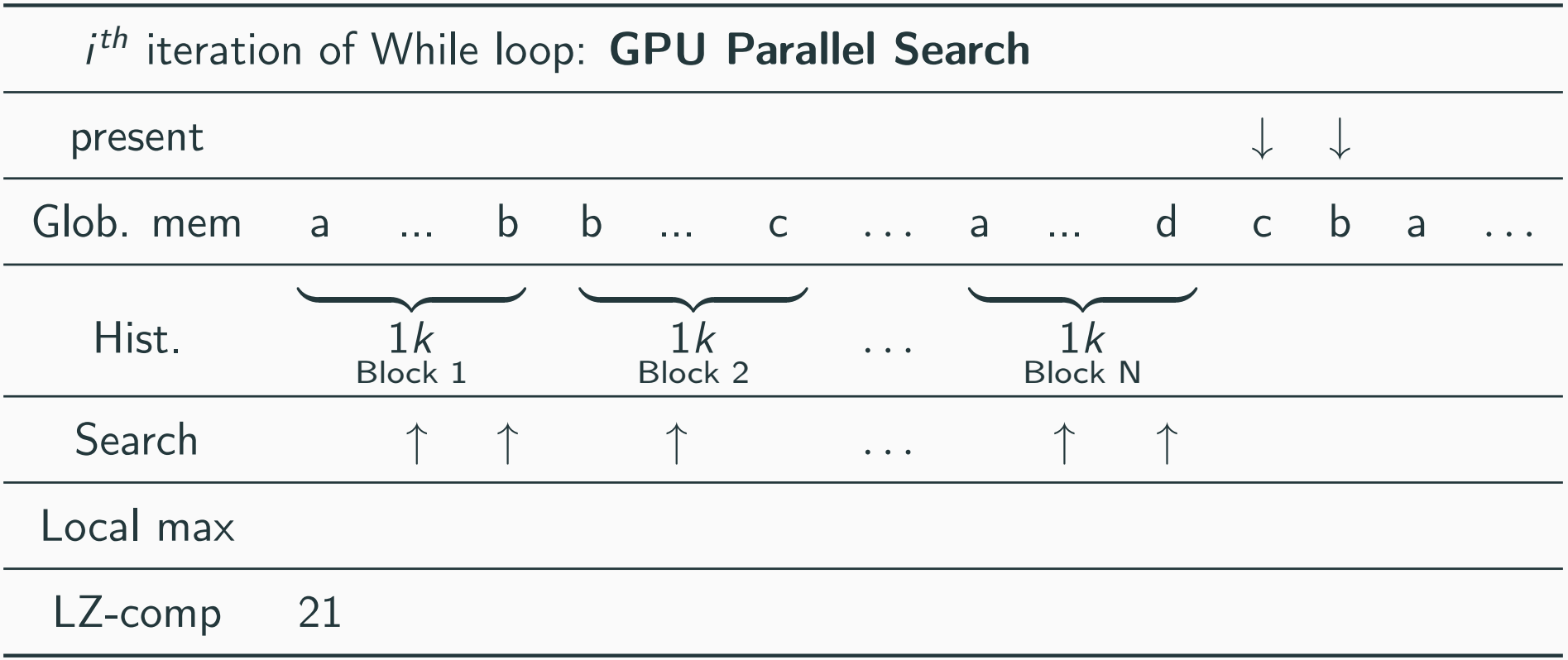
Kernel_global



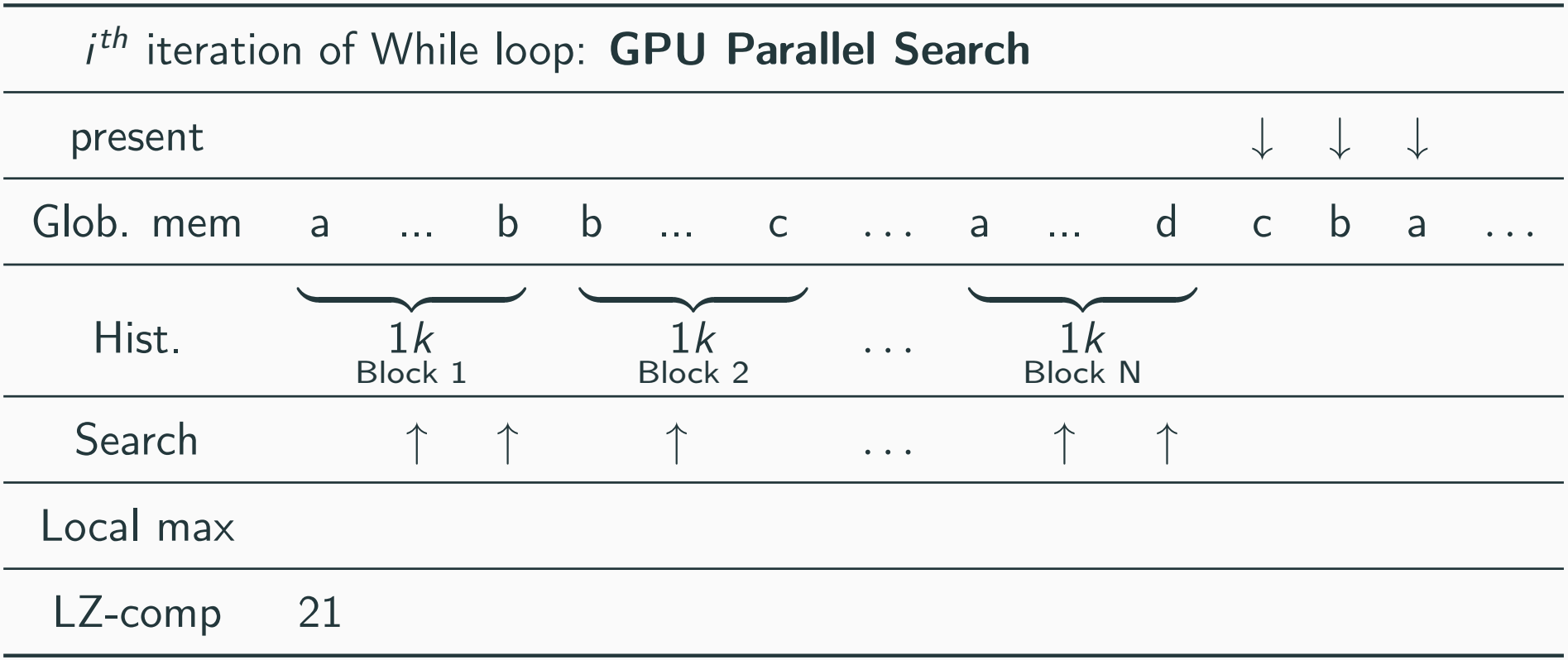
Kernel_global

i^{th} iteration of While loop: **GPU Parallel Search**															
present												↓			
Glob. mem	a	...	b	b	...	c	...	a	...	d	c	b	a	...	
Hist.	$1k$ Block 1			$1k$ Block 2			...	$1k$ Block N							
Search	...			...			...	...							
Local max															
LZ-comp	21														

Kernel_global



Kernel_global



Kernel_global

i^{th} iteration of While loop: **GPU Local maxima** (use parallel reduction, do loop unrolling on last warp)															
present												↓			
Glob. mem	a	...	b	b	...	c	...	a	...	d	c	b	a	...	
Hist.	$1k$ Block 1			$1k$ Block 2			...	$1k$ Block N							
Search			↑			↑	...			↑					
Local max[(1)]			3			1				2					
LZ-comp	21														

[(1)] Implemented in managed memory model (CPU and GPU share)

Kernel_global

i^{th} iteration of While loop: **CPU Compute max**															
present												↓			
Glob. mem	a	...	b	b	...	c	...	a	...	d	c	b	a	...	
Hist.	$1k$ Block 1			$1k$ Block 2			...	$1k$ Block N							
Search			↑			↑	...			↑					
Local max			3			1				2					
Max	3														
LZ-comp	21														

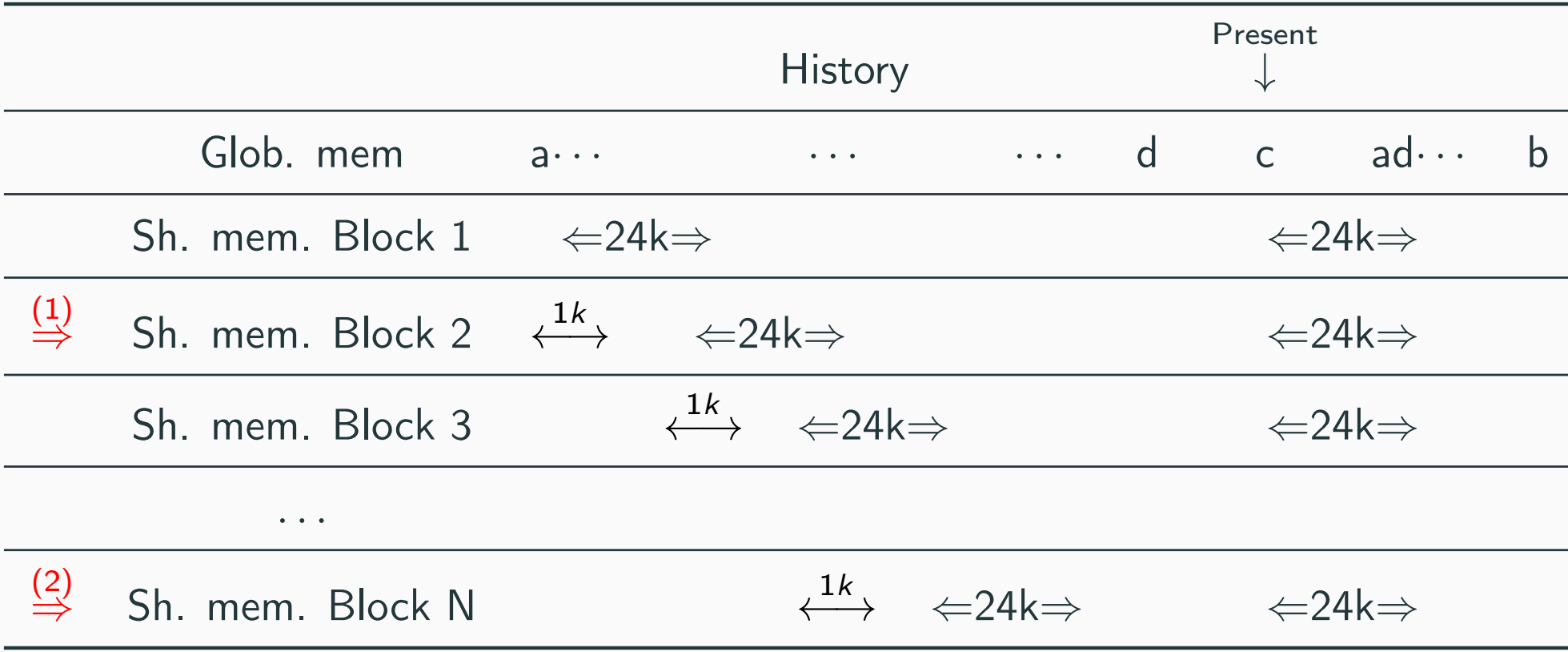
Kernel_global

i^{th} iteration of While loop: **CPU Increment LZ**																
present													↓			
Glob. mem	a	...	b	b	...	c	...	a	...	d	c	b	a	...		
Hist.	$1k$ Block 1			$1k$ Block 2			...	$1k$ Block N								
Search																
Local max																
Max	3															
LZ-comp	22															

Kernel_global

i^{th} iteration of While loop: **CPU Increment present**															
present															↓
Glob. mem	a	...	b	b	...	c	...	a	...	d	c	b	a	...	
Hist.	$1k$ Block 1			$1k$ Block 2			...	$1k$ Block N							
Search															
Local max															
Max	3														
LZ-comp	22														

Kernel_shared



(1,2) Two blocks execute at any given time, each using 48k shared mem. (total 96k on GPU)

Results

Kernel	Test	Result
global	random binary strings of lengths n= 50k,100k,...,2M	Speed Factor (SF) = $O(n^{2/3})$ SF=150 for n= 2M
global	Profiling, n= 1M random strings	High efficiency: 90% kernel execution 10% for data transfers
global vs. shared	random word documents 50k, ..., 2M (from dictionary of 50,000 Dutch words)	[(1)] Kernel_global performed better than Kernel_shared (except for the 50k documents)

[(1)] Tiling not cost-effective (perhaps, because random word documents have shorter word-components, search finishes quicker)

Conclusions

Algorithm ParallelLZ computes LZ complexity for arbitrarily long strings (up to GPU mem size 32G)

- **obtains** Speedup Factor 150 for strings of length at least up to 2M (Kernel_global)
- **scalable** number of blocks grows with length of string
- **Good** option for machine learning based on raw (unstructure) data

Significant improvement in speed:

- **Static** pointers on device, instead of passing dynamically allocated
- **Reduction** for max calculation with large stride
- **Loop** unrolling for last executing warp
- **Managed** memory used for sharing max calculation between CPU & GPU

Thanks for your attention.