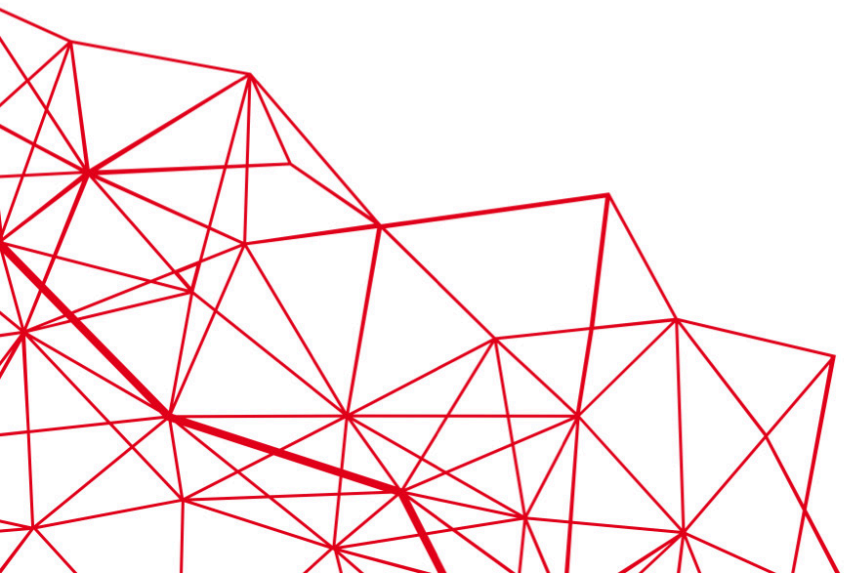


ISC High Performance

REINVENTING HPC

MAY 12 – 16, 2024 | HAMBURG, GERMANY

Multi-GPU processing of unstructured data for machine learning

ISC'24 High Performance Computing, Hamburg, Germany,
May 12 – 16, 2024

Joel Ratsaby, Alexander Timashkov
Ariel University
ratsaby@ariel.ac.il

Introduction

Unstructured-data is ubiquitous ($> 80\%$ of data),

primitive no pre-processing

no fixed input dimensionality

hybrid character, binary, text, XML, ...

Want Machine-Learning (ML) from *raw da*ta (with basic parsing or none, without feature extraction)

LZ-complexity Measure of string complexity based on universal data compression

LZ-distance distance $d(x, y)$ for strings x and y (sequences), may be 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 LZD algorithm to compute *LZ-distance matrix*

Implement CUDA on multi-GPU computer

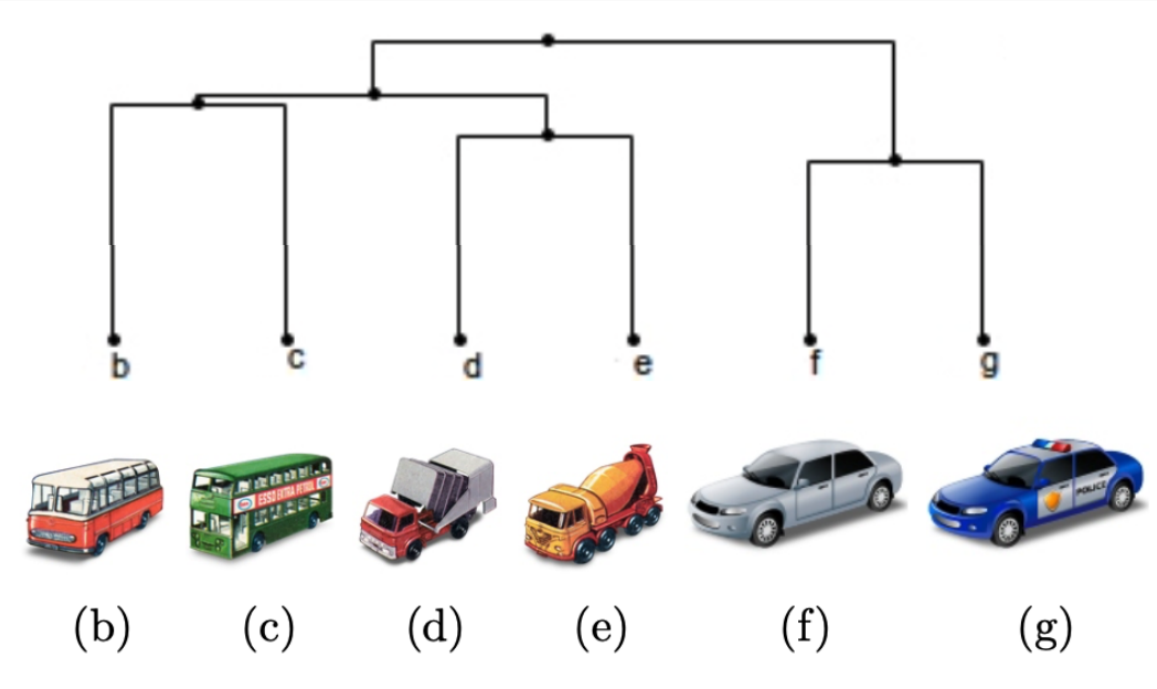
High-Speedup ~ 500 (for LZ-distance of 1Mb strings)

ML by LZD

Distance-matrix representation of data $[d(x_i, x_j)]_{i,j=1}^{N}$

Learn from raw data, time series, event logs, ...

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



Unstructured Data

Time-series sequential data

Electric current drive signals, to classify defective motor components.

A data instance is a *sequence* of real numbers, with possibly different lengths for different instances

time	1	2	3	4	5	6	7	8	9
data 1	-3.0146E-07	8.2603E-06	-1.1517E-05	-2.3098E-06	-1.4386E-06	-2.1225E-05	0.031718	0.03171	0.031721
data 2	2.9132E-06	-5.2477E-06	3.3421E-06	-6.0561E-06	2.7789E-06	-3.7524E-06	0.030804		

Sequential Event Log data

Hadoop Application data set.

A data instance is a *sequence* of event IDs

time	1	2	3	4	5	6	7	8	9	10	...																						
data 1	50	66	73	42	61	43	50	66	73	42	36	61	43	50	73	67	68	16	69	52	45	32											
data 2	50	66	73	42	61	43	50	66	73	42	61	43	50	66	73	42	61	43	50	66	73	42	61	43	50	66	73	42	61	43	50	66	36	73
data 3	50	66	73	42	61	43	50	66	73	42	61	43	50	66	73	42	61	43	50	66	73	42	61	43	50	66								
data 4	50	66	73	42	61	43	50	66	73	42	61	43	50	66																				

Multivariate time series sequential data

Chemical Sensors data.

A data instance is a *sequence* of real numbers, with possibly different lengths for different instances

Time (s)	CO (ppm)	Humidity (%r.h.)	Temperature (C)	Flow rate (mL/min)	Heater voltage (V)	R1 (MOhm)	R2 (MOhm)	R3 (MOhm)	R4 (MOhm)
0.0000	0.0000	48.4700	24.6200	247.4926	0.2000	0.6831	0.6992	1.1940	16.0255
0.3090	0.0000	48.4700	24.6200	243.8282	0.1998	0.6649	0.6976	1.1514	15.2863
0.6180	0.0000	48.4700	24.6200	243.0668	0.2000	0.6481	0.6863	1.1133	14.3923

Recordings of people performing different activities

Sequence Name	Tag identificator	timestamp	date	x coordinate of the tag	y coordinate of the tag	z coordinate of the tag	activity {walking,falling,'lying down',lying,'sitting down',sitting,'standing up from lying','on all fours','sitting on the ground','standing up from sitting','standing up from sitting on the ground'}
A01	010-000-024-033	633790226051280329	27.05.2009 14:03:25:127	4.062931060791020	1.892434239387510	0.507425427436829	walking
A01	020-000-033-111	633790226051820913	27.05.2009 14:03:25:183	4.291953563690190	1.781140446662900	1.344495296478270	walking
A01	020-000-032-221	633790226052091205	27.05.2009 14:03:25:210	4.359101295471190	1.826455712318420	0.968820989131928	walking
A01	010-000-024-033	633790226052361498	27.05.2009 14:03:25:237	4.087835311889650	1.879998683929440	0.466983407735825	walking

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

LZ-complexity equals 8

LZ distance

Given two sequences S_1, S_2

Distance $d(S_1, S_2) := \frac{LZ(S_1 S_2) - \min\{LZ(S_1), LZ(S_2)\}}{\max\{LZ(S_1), LZ(S_2)\}}$

Parallel LZD (Distance Matrix)

Input: strings S_1, S_2, ..., S_N

// Pool of available GPUs

// stage 1

for $1 \leq i \leq N$

$c_i := \textbf{ParallelLZ}(S_i)^{\dagger}$ // on available GPU

// stage 2

for $1 \leq i \leq N$, $1 \leq j \leq N$

$S_{i,j} := S_i \cdot S_j$ // Concatenate

$c_{i,j} := \textbf{ParallelLZ}(S_{i,j})$ // on available GPU

$d_{i,j} := \frac{c_{i,j} - \min\{c_i, c_j\}}{\max\{c_i, c_j\}}$

// LZD matrix

$\mathbf{D} := [d_{i,j}]_{i,j=1}^{N}$

Output: D

[†] Ratsaby & Timashkov (ICPADS'23)

Speedup

Given N strings of length $1M$ bytes

Compute $N \times N$ LZD matrix

Compare serial versus parallel algorithm

Comparison

N=4	TS	TP
Stage 1 execution time		
Total (4 1M-strings)	16.31 min	3 sec
Average per string	4.08 min	2.75 sec
Stage 2 execution time		
Total (16 2M-strings)	3.22 hr	27 sec
Average per string	12.07 min	1.69 sec
Total time: Stages 1 + 2	**3.49 hr**	**30 sec**
Speed Factor: 419		

N=8	TS	TP
Stage 1 execution time		
Total (8 1M-strings)	32.57 min	6 sec
Average per string	4.07 min	3 sec
Stage 2 execution time		
Total (64 2M-strings)	14.28 hr	1.7 min
Average per string	13.39 min	1.59 sec
Stages 1 + 2	**14.82 hr**	**1.8 min**
Speed Factor: 494		

N=16	TS	TP
Stage 1 execution time		
Total (16 1M-strings)	1.09 hr	12 sec
Average per string	4.07 min	2.63 sec
Stage 2 execution time		
Total (256 2M-strings)	59.98 hr	6.73 min
Average per string	14.06 min	1.58 sec
Stages 1 + 2	**61.07 hr**	**6.93 min**
Speed Factor: 528		

Machine Learning

Raw-Data Set of N data instances, each a sequence (character or binary), e.g., multi-dimensional time-series[4]

LZD-representation Matrix, D $:= [d_{i,j}]_{i,j=1}^{N}$, $d_{i,j}$ is distance between i^{th} sequence and j^{th} sequence

	1	2	3	$\cdots$	N
1	0.24	0.11	0.02	$\cdots$	0.51
2	0.89	0.28	0.7	$\cdots$	0.98
3	0.31	0.19	0.58	$\cdots$	0.13
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
N	0.44	0.39	0.89	$\cdots$	0.31

Train standard Machine Learning algorithms (supervised, unsupervised)

Experiments

Trained 30 Machine Learning Algorithms on the LZD-representation of the data[4]

Including

- **k-NN** based on Euclidean distance (k = 1, 3, 5, 7, 10)
- **Naive-Bayes**
- **C5.0** classification tree learning
- **Multi-layer** perceptrons (Neural networks)
- **SVM** with two different kinds of kernels

Results

Compare LZD-representation vs. standard TFIDF-representation (word based)

Result almost all of the 30 algorithms consistently achieve significantly higher learning accuracy based on LZD data representation (details in the paper)

Conclusions

Parallel algorithm for processing unstructured data for any standard Machine Learning algorithm

Computing the LZD matrix is automatic, based only on the original (raw) data, no feature extraction

Useful for small-shot learning (up to a few thousand data instances)

Inference can also be implemented on edge-computing device (e.g., nVidia Jetson boards) and done locally close to data source (no need for large centralized inference machine)

Thanks for your attention.

Appendix

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

`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

}

GPU

Kernel uses global memory

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

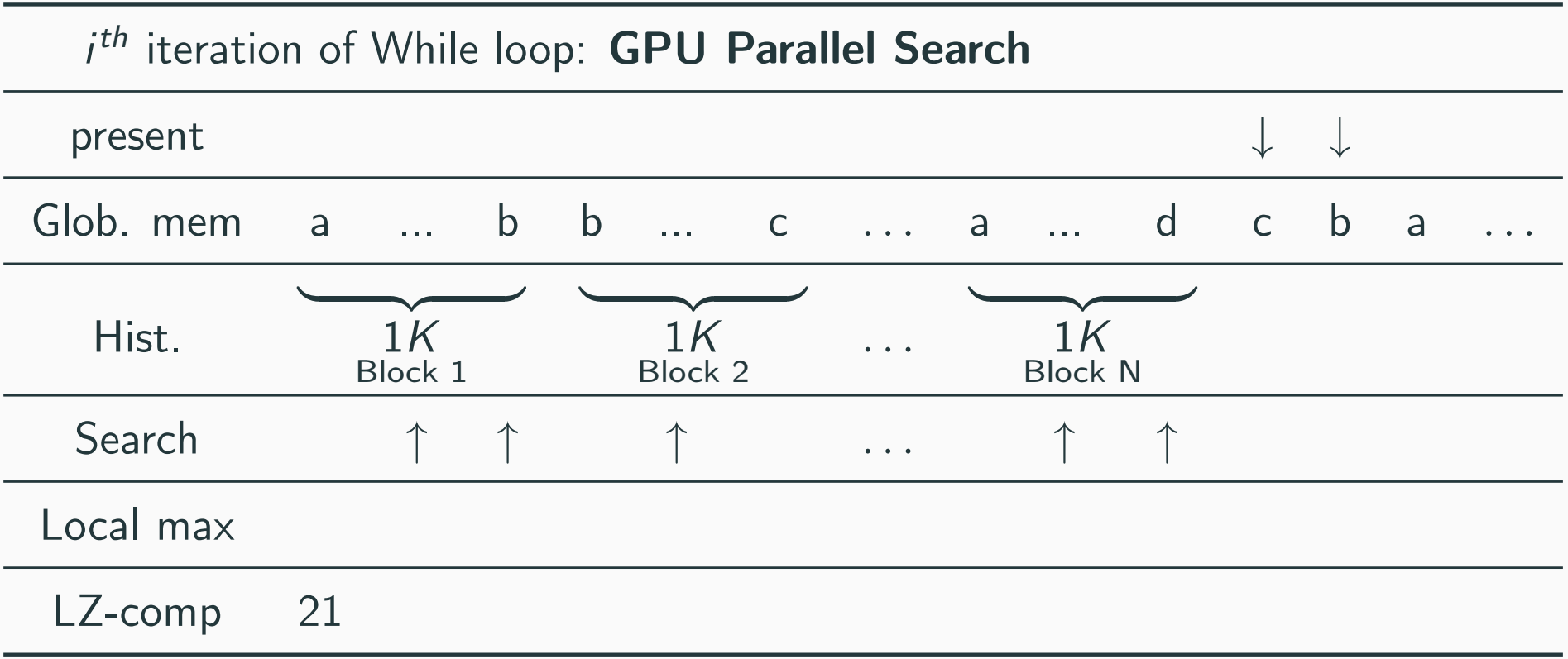
Kernel

i^{th} iteration of While loop: **Parallel Launch**																
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															

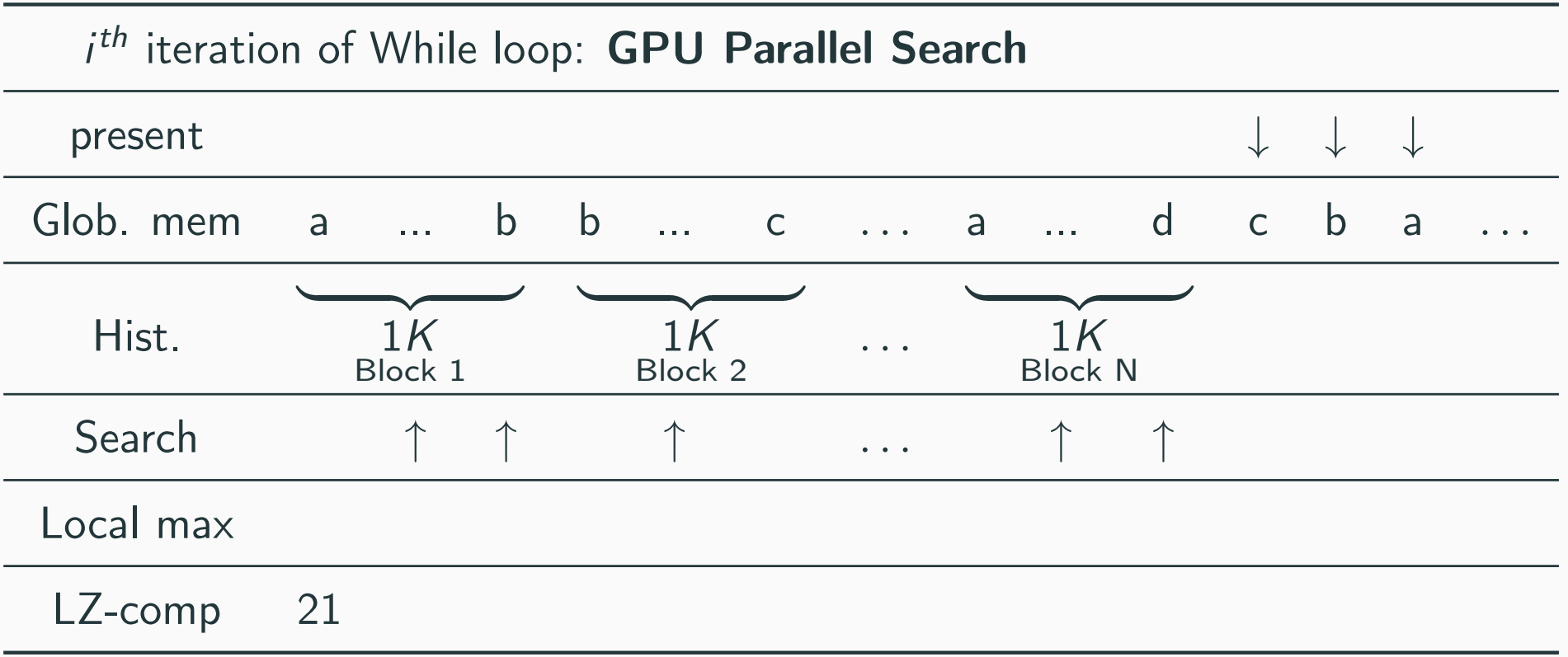
Kernel

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

Kernel


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


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

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.	$\overbrace{\quad}^{1K}$ Block 1			$\overbrace{\quad}^{1K}$ Block 2			...	$\overbrace{\quad}^{1K}$ Block N							
Search			↑			↑	...			↑					
Local max[(1)]			3			1				2					
LZ-comp	21														

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

Kernel

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

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

Kernel

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