Big Sequence Management
Scaling Up and Out

Karima Echihabi
Mohammed VI Polytechnic University

Kostas Zoumpatianos
Snowflake Computing
work done while at Harvard University & University of Paris

Themis Palpanas
University of Paris & French University Institute (IUF)

International Conference on Extending Data Base Technology (EDBT), March 2021

Questions This Tutorial Answers

- how important are data series nowadays?

- what does data series analysis involve?
- how can we speed up such an analysis?

- what are the different kinds of similarity search?
- what are the state-of-the-art data series indexes for similarity search?
- how can these indexes parallelize/distribute their operations?
- can these indexes be used for general high-d vector similarity search?

- what data series management systems exist?
- what is special about them?

- what are the open research problems in this area?
- what are the connections to deep learning?

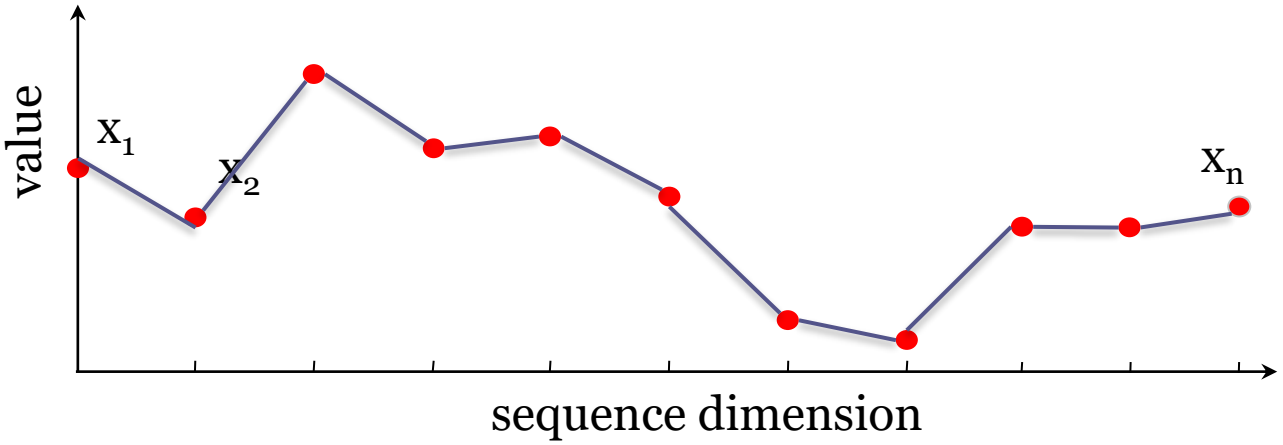
Acknowledgements

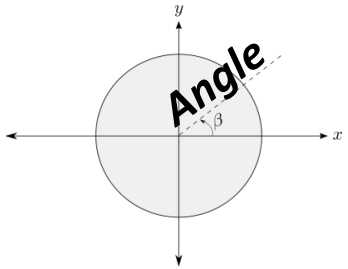
- thanks for slides to
  - Michail Vlachos
  - Eamonn Keogh
  - Panagiotis Papapetrou
  - George Kollios
  - Dimitrios Gunopulos
  - Christos Faloutsos
  - Panos Karras
  - Peng Wang
  - Liang Zhang
  - Reza Akbarinia
  - Georgios Chatzigeorgakidis

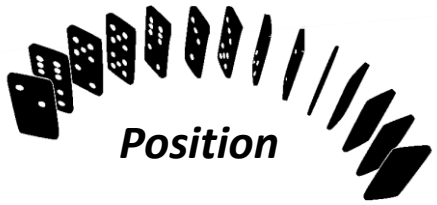
Introduction, Motivation

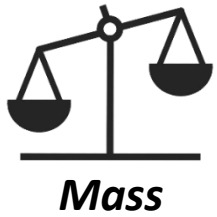
Data series

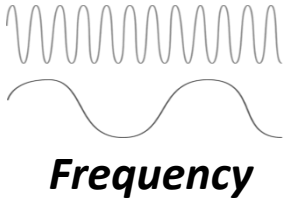
- Sequence of points ordered along some dimension





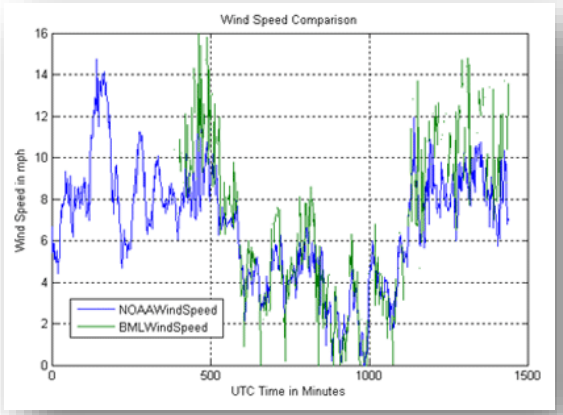






Scientific Monitoring

- meteorology, oceanography, astronomy, finance, sociology, …



Wind speed

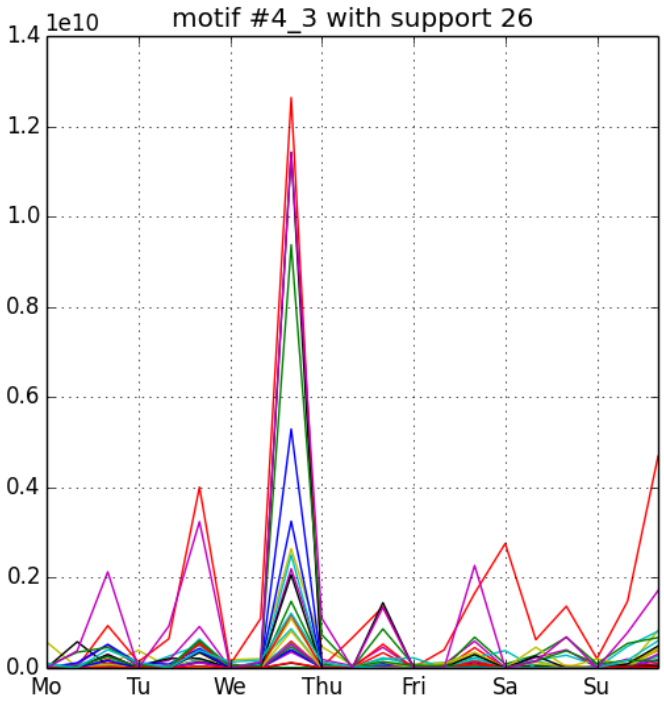
From ocean observing node project
http://bml.ucdavis.edu/boon/wind.html

Historical stock quotes

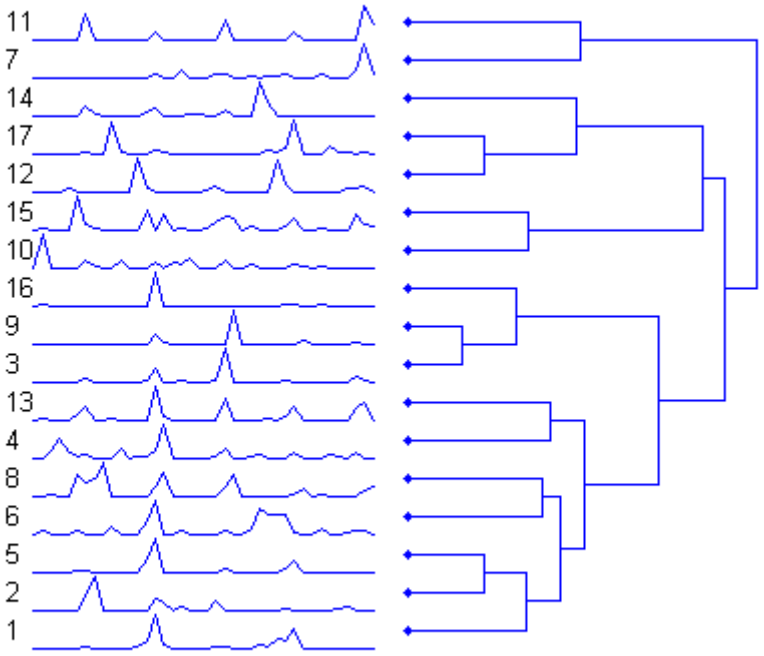
http://money.cnn.com/2012/04/23/markets/walmart_stock/index.htm

Home Networks

- temporal usage behavior analysis of home networks
  - Portugal Telecom



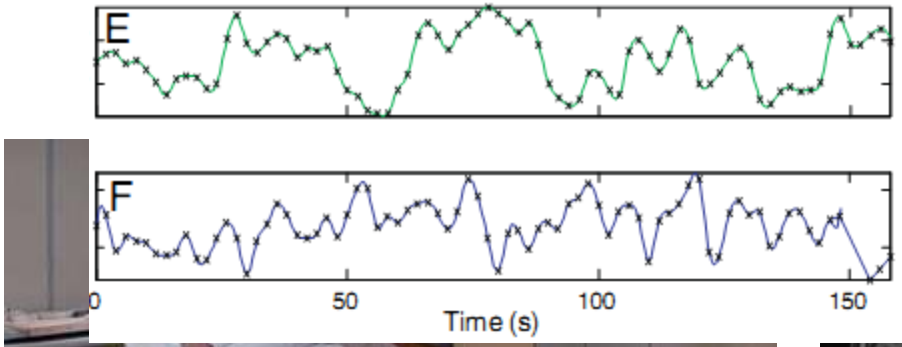
(previously unknown) frequent behavior pattern

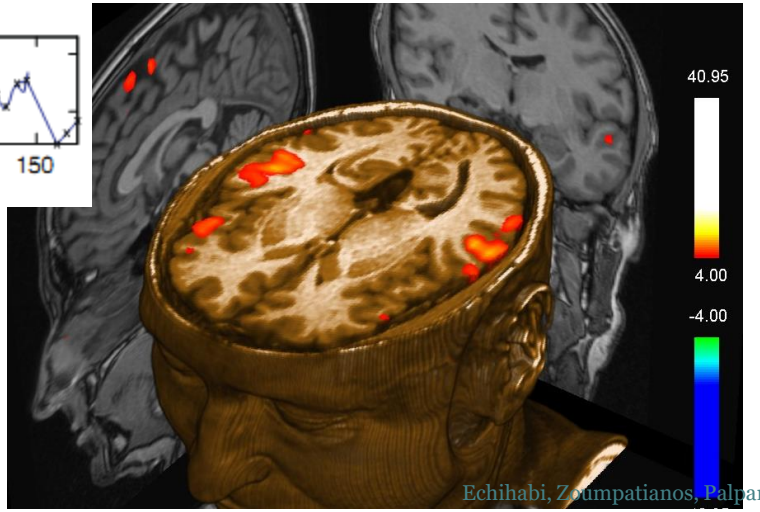


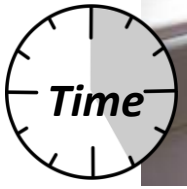
clustering based on user activity patterns

Neuroscience

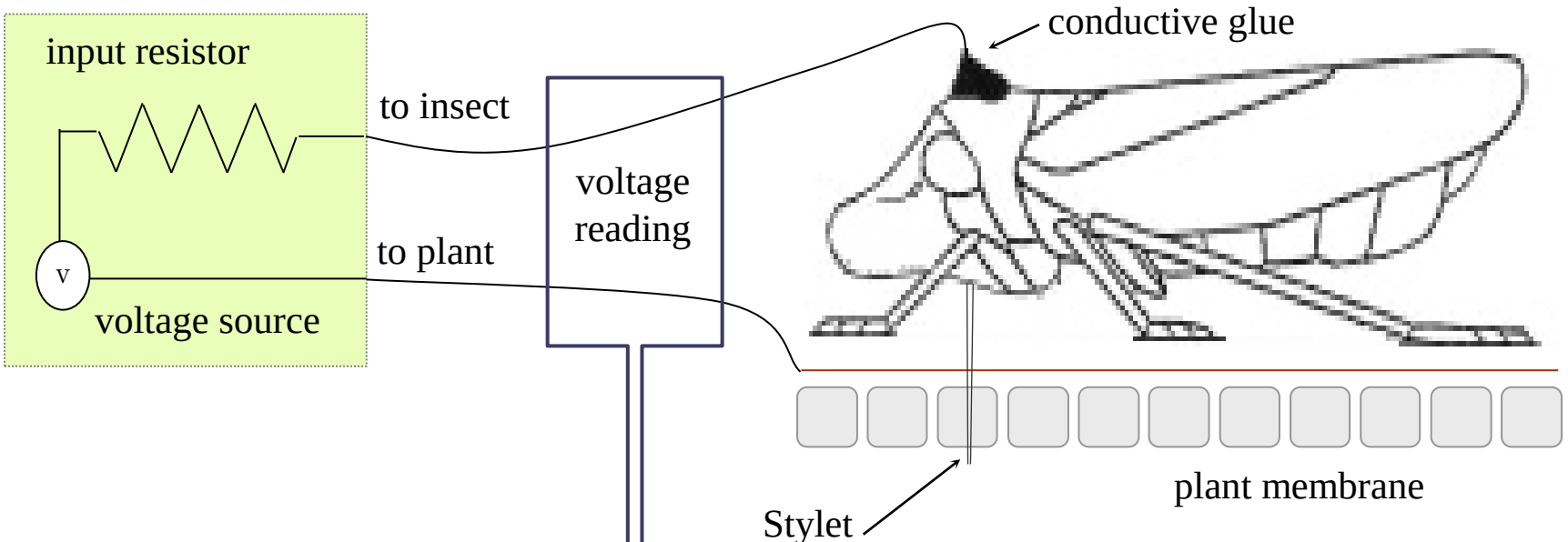
- functional Resonance Magnetic Imaging (fMRI) data
  - primary experimental tool of neuroscientists
  - reveal how different parts of brain respond to stimuli

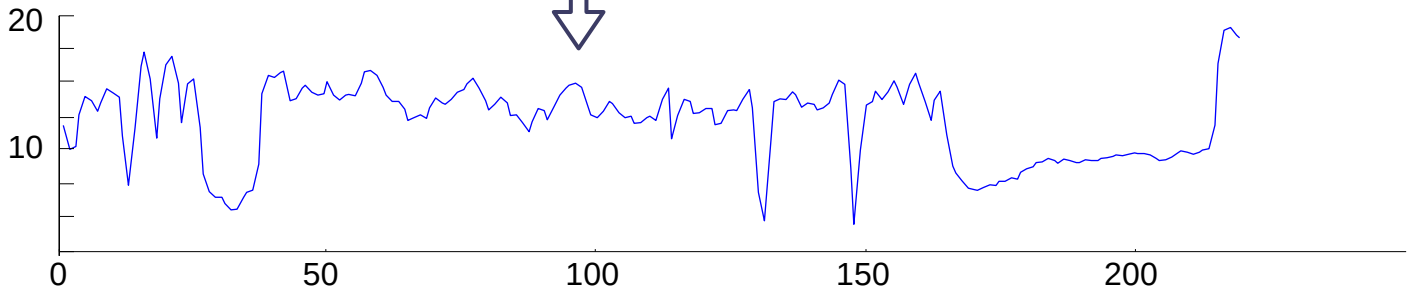




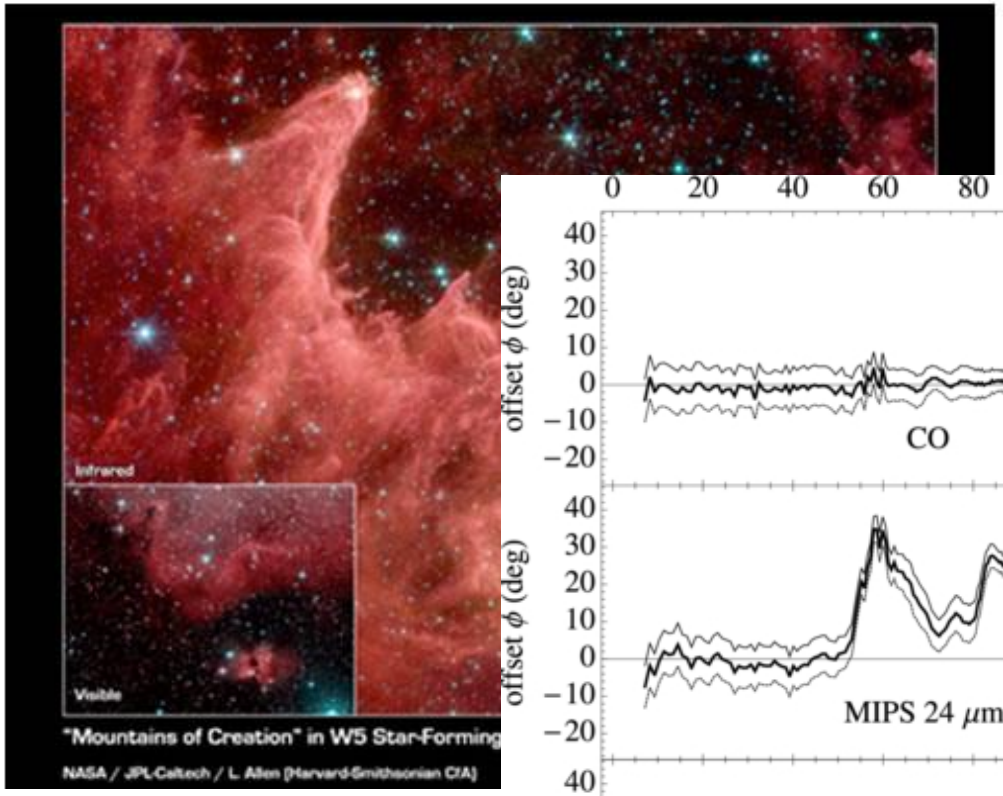


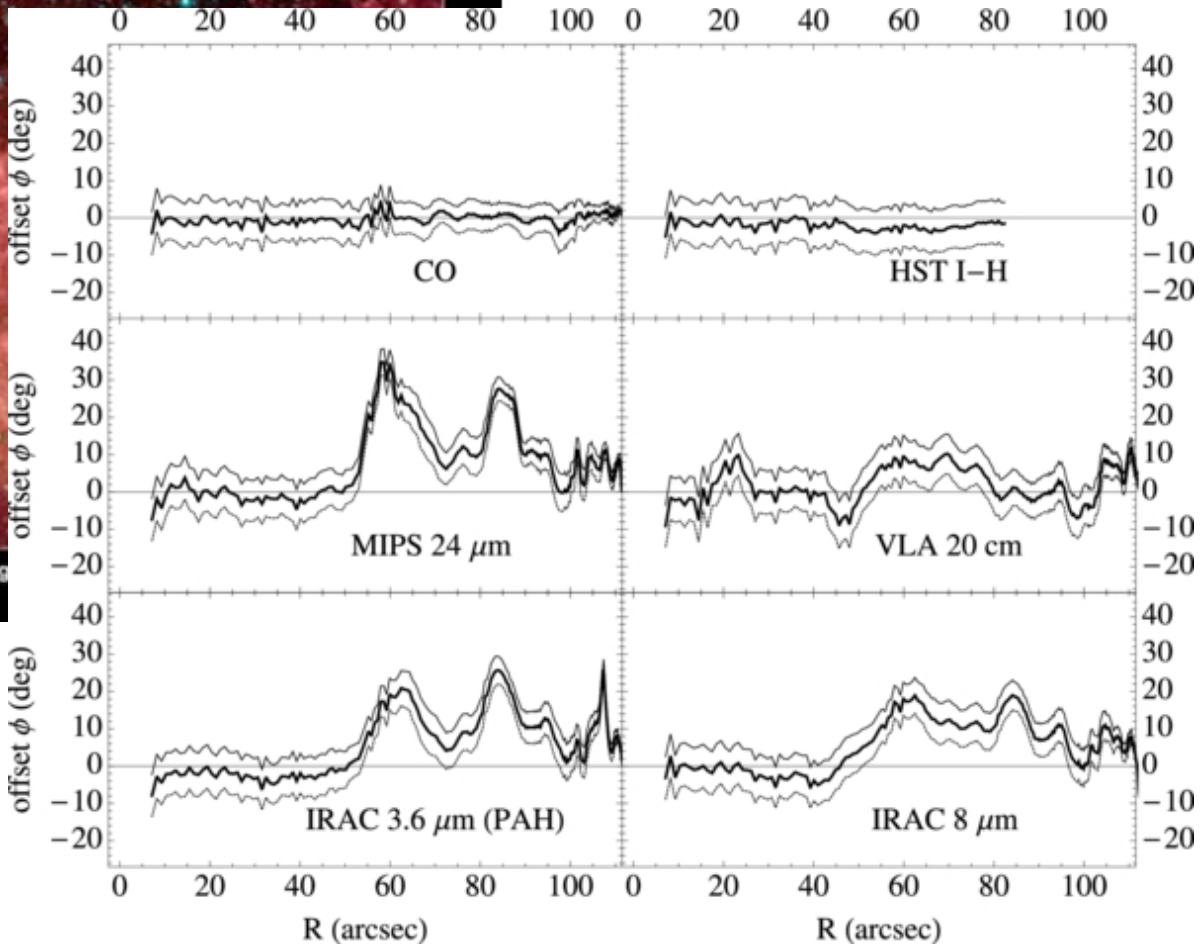
Entomology

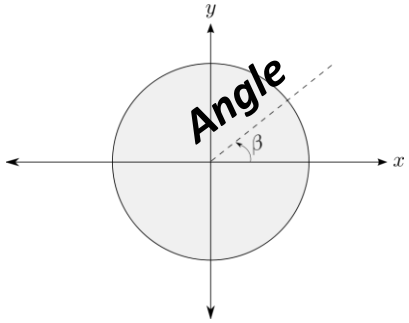

input resistor
to insect
to plant
V
voltage source
voltage reading
conductive glue
plant membrane
Stylet


20
10
0
50
100
150
200


Time

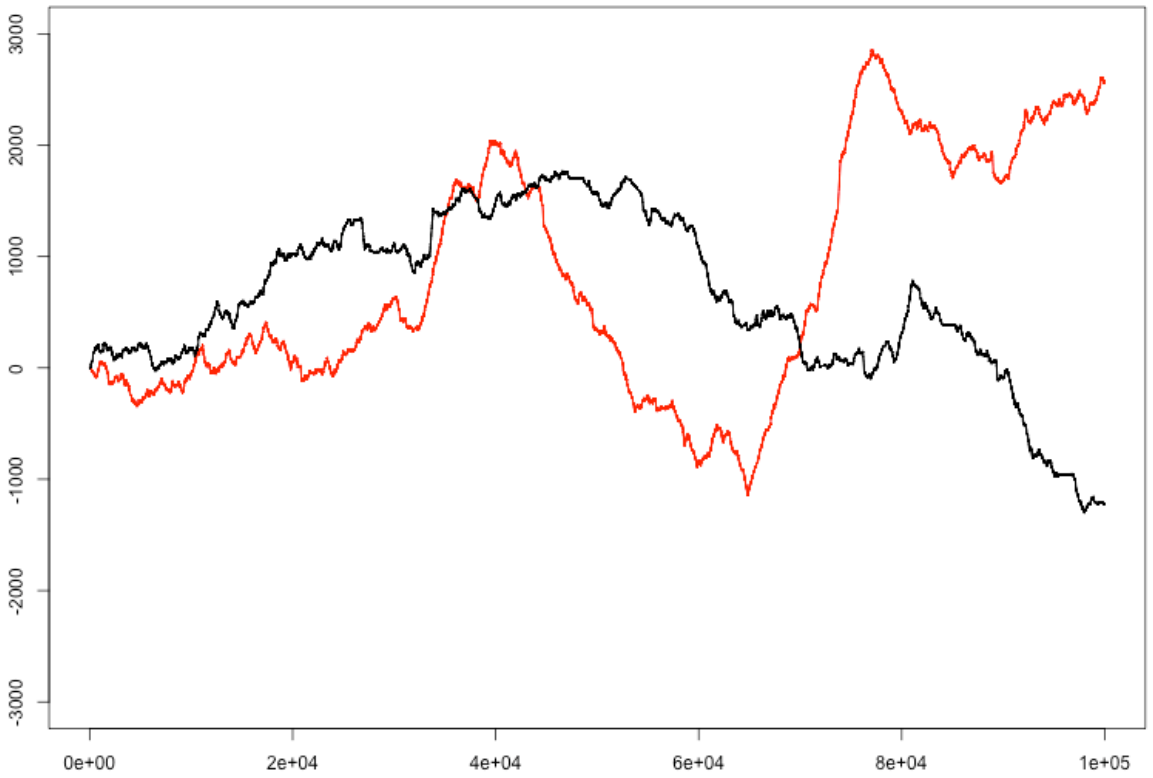


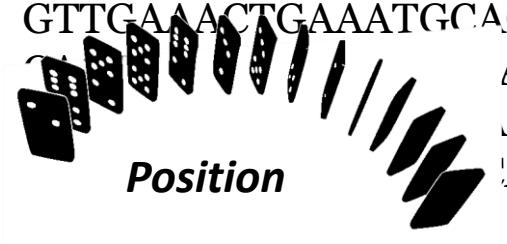




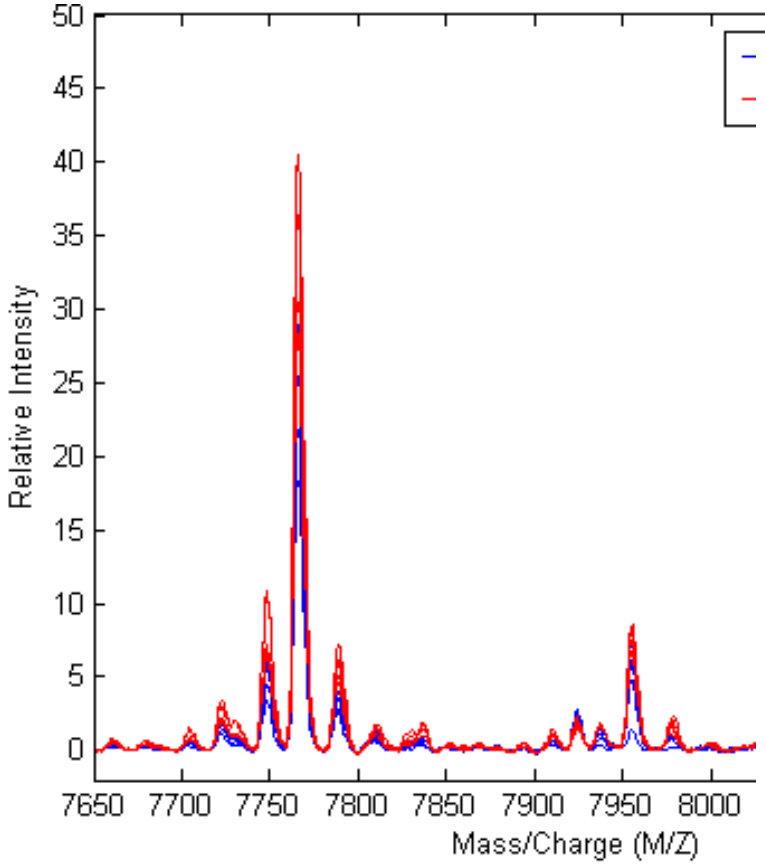
Schinnerer et al.

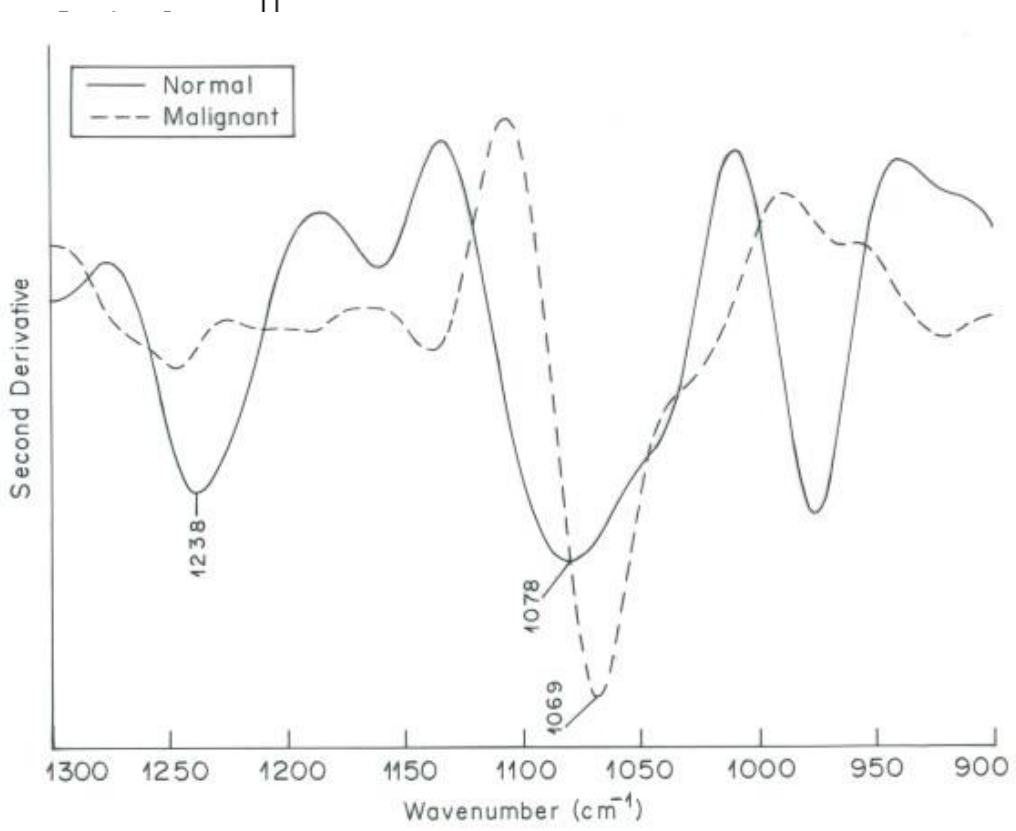
GTCAATGGCCAGGATATTAGAACAGTACTCTGTGAACCCTATTTATGGTGGCACCCCTTAGACTAA
GATAACACAGGGAGCAAGAGGTTGACAGGAAAGCCAGGGGAGCAGGGAAGCCTCCTGTAAAGAG
AGAAGTGCTAAC[illegible]TTGACTAAAGC
CCAAGGGATTG[illegible]AGTCAGATTAA
TATGTGGGTGC[illegible]TGAGGAGAAA
CTCCTCCACAGC[illegible]GCCAGAGGATC
TGTAAAAGAGA[illegible]ACAAGCAAAA
AAGAGCACTCA[illegible]GGCCACCAAG
CCAGTTGTCAAT[illegible]GAGTGTCTTCT
ATTTCTTCCAGC[illegible]CCTTTACAATT
GTTATTTATGTC[illegible]CATACTGTCTT
GGCCAAGCATCC[illegible]TGCACGTGGT
TATTTATTAGTC[illegible]AAGAGCAGAAT
CAATGAGCCGC[illegible]AGATGAGCATT
GAAGAGGATGC[illegible]CAGGTTGTTGC
CATGTTTGTTTT[illegible]TTAGAACAGTA
CTCTGTGAACCC[illegible]AGAGGTTGACA
GGAAAGCCAGG[illegible]TTTCTAAGGCA
CATGATGGATC[illegible]TAATCCGATTC
TTGGCAGAAGA[illegible]ATAAACAAATA
ATTGAATAATCC[illegible]AGGCAGAACCG
GTTGAAACTGAAATGC[illegible]ACCCGCTGCCAGATTTATTAGTCACCCAAGCATGTATTTTGCATGTCCAT
[illegible]AAGAACAGAATCAATGAGCCGCTGCAGATGCAGACATAGCAGCCCCTTG
[illegible]GATGAGCATTGAAGAGGATGCACAAGCCCGGTAGCCCGGGAAATGGCA
[illegible]AGGTTGTTGCCATGTTTGTTTTTGCAACTTGTCTTTTAAACAGATTTGA



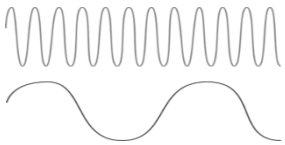


Medicine





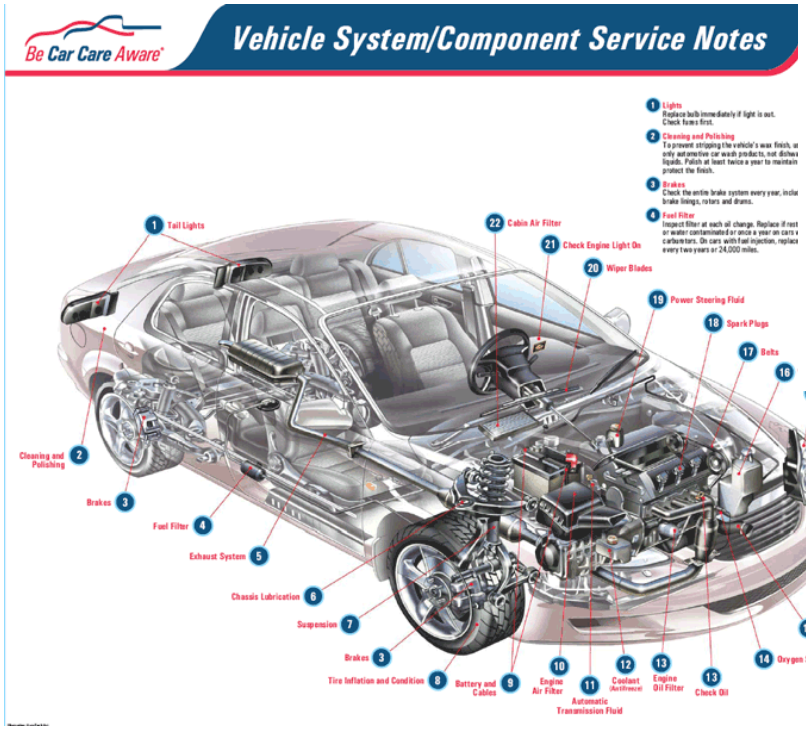
Mass

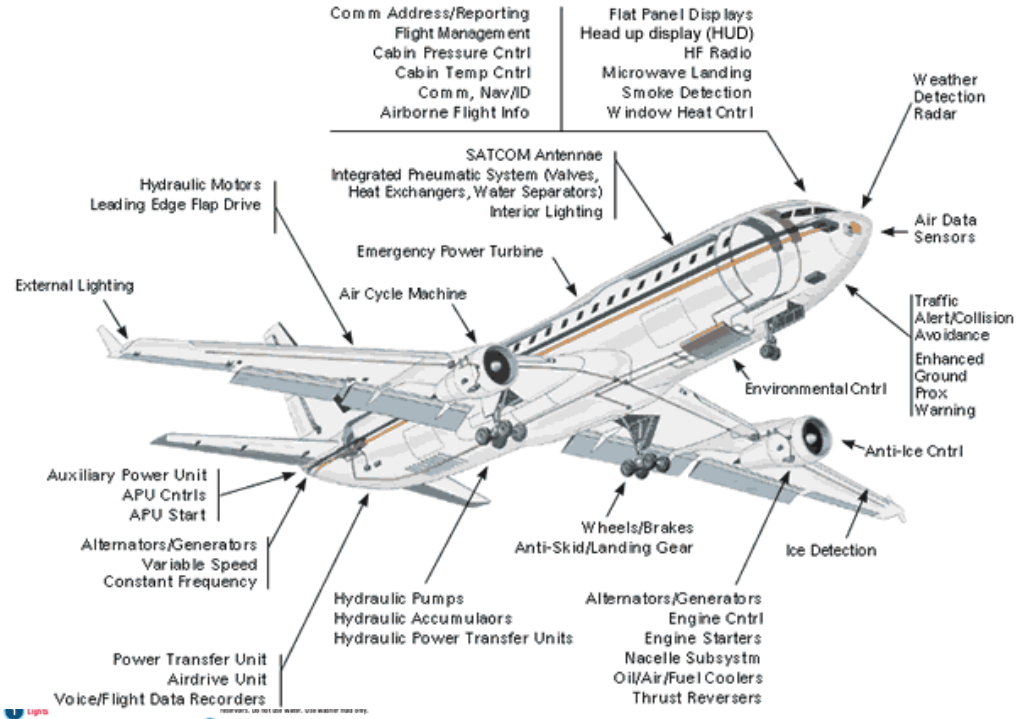


Frequency

IoT: Production Control Systems

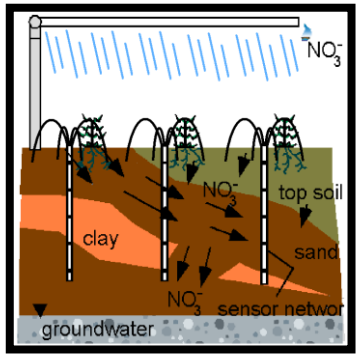
IoT: Monitoring Vehicle Operation

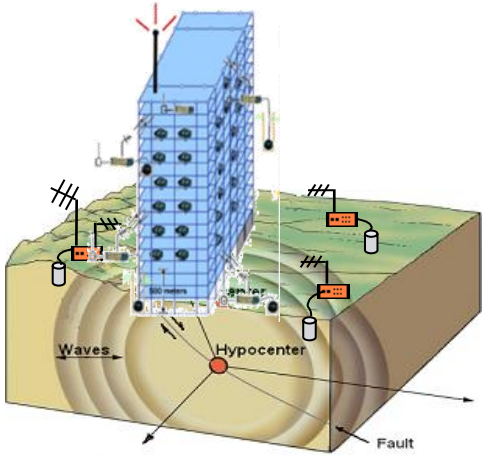




IoT: Monitoring the Environment

- the sensors era
  - ubiquitous, small, inexpensive sensors
  - applications that bridge physical world to information technology
- sensors unveil previously unobservable phenomena





Data as a Set
Data as a Sequence

- streaming data
  - window of interest
    - landmark window
    - sliding window (shifting window)

- may treat streaming data as a set, or as a sequence
  - depends on whether sequence is important

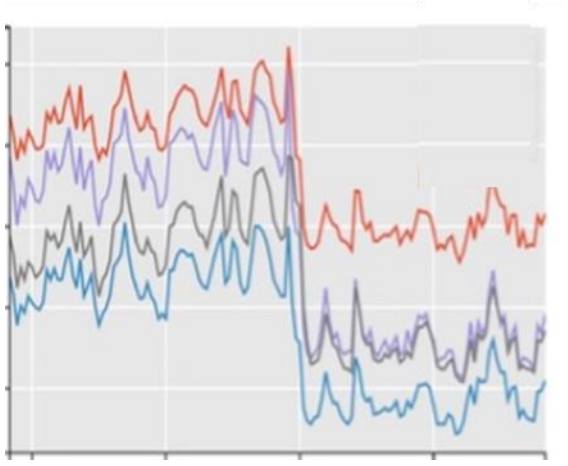
Data Series Anomalies Problem

- develop anomaly detection techniques based on sequences (data series), not on individual values
  - individual values can be normal, but their sequence can be abnormal!

Data Series Anomalies Problem

- develop anomaly detection techniques based on sequences (data series), not on individual values

Sequence S



sequences are abnormal

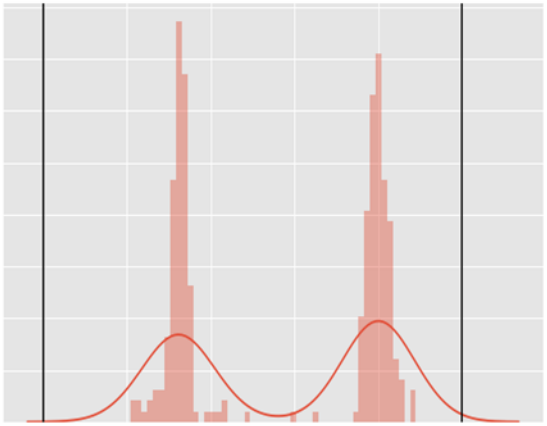
Data Series Anomalies Problem

- develop anomaly detection techniques based on sequences (data series), not on individual values
  - individual values can be normal, but their sequence can be abnormal!

150 points in a sequence S

Minimal critical value

Maximal critical value



sequences are abnormal

values are not outside critical thresholds
individual values are normal

Data Series (Signal) Processing
Data Series Management

- lots of literature on data series processing
  - periodicity detection
  - data series modeling and forecasting
    - ARMA, ARIMA
  - point outlier detection
    - focuses on next value

- instead, we will focus on
  - sequences as first class citizens
  - very large collections of data series
  - fast and scalable similarity search

Objectives

- get introduced to the data series data type
  - characteristics, properties, peculiarities
- learn about
  - data series representations
  - data series similarity matching
  - data series indexing
  - systems for data series management
  - challenges and open problems

Data Series Representations

Introduction

- lots of work on data series representations
  - techniques for representing/storing data series

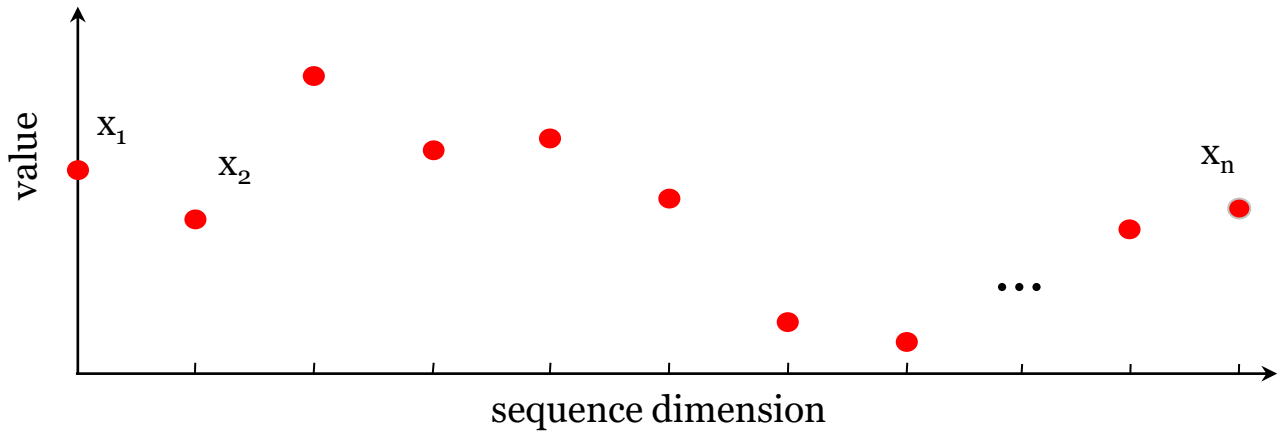
- main goal
  - summarize data series
  - render subsequent processing more efficient

Outline

- terminology and definitions
- motivation
- pre-processing tasks
- data series representation techniques

Data series

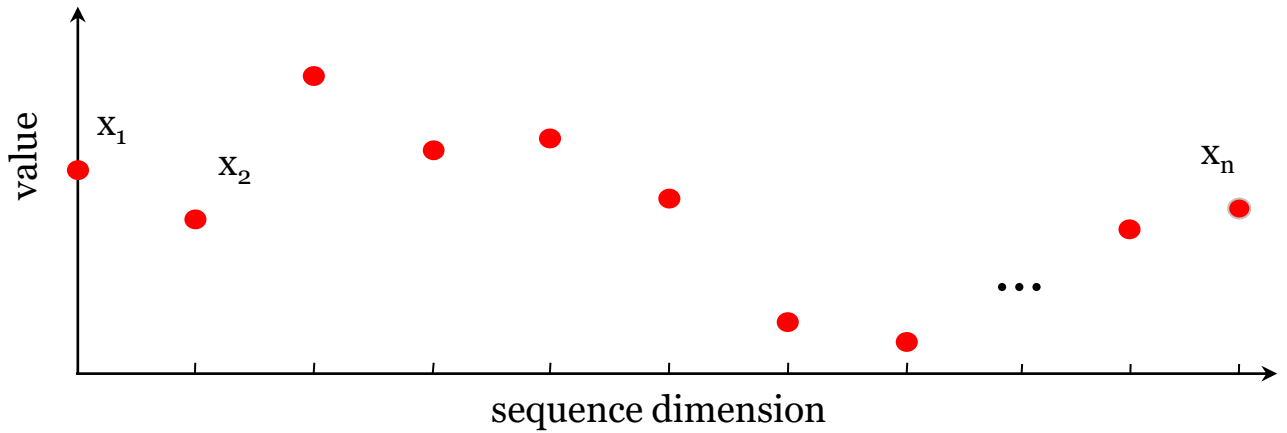
- Sequence of points ordered along some dimension



- terminology: we will use interchangeably
  - data series, time series, data sequence, sequence

Data series

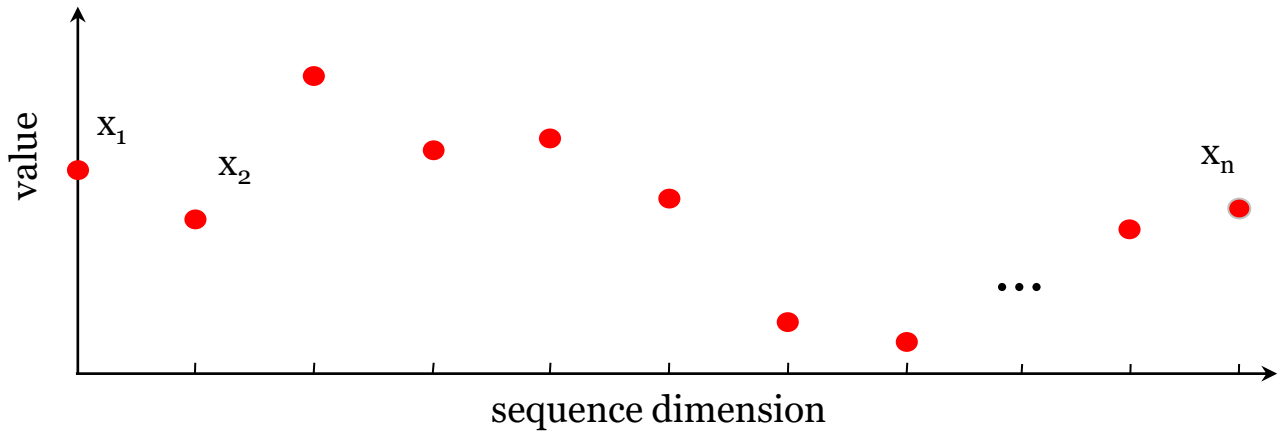
- Sequence of points ordered along some dimension



- number of data series values, n
  - length, or dimensionality

Data series

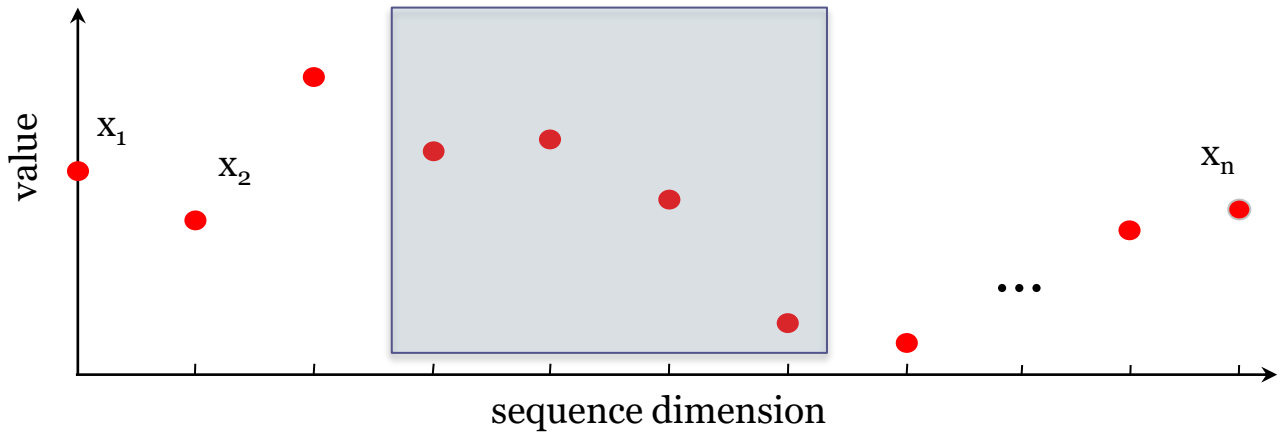
- Sequence of points ordered along some dimension



- subsequence
  - subset of contiguous values

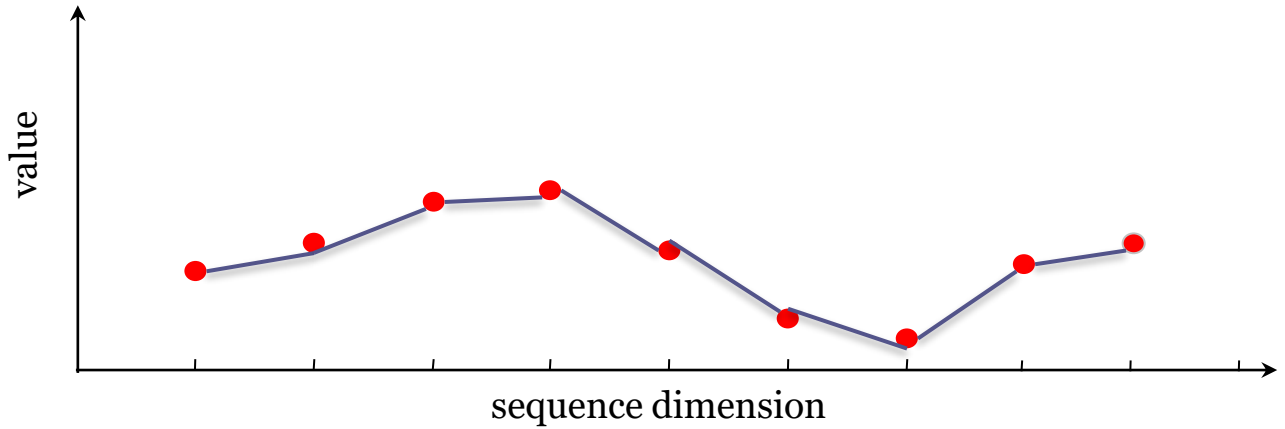
Data series

- Sequence of points ordered along some dimension

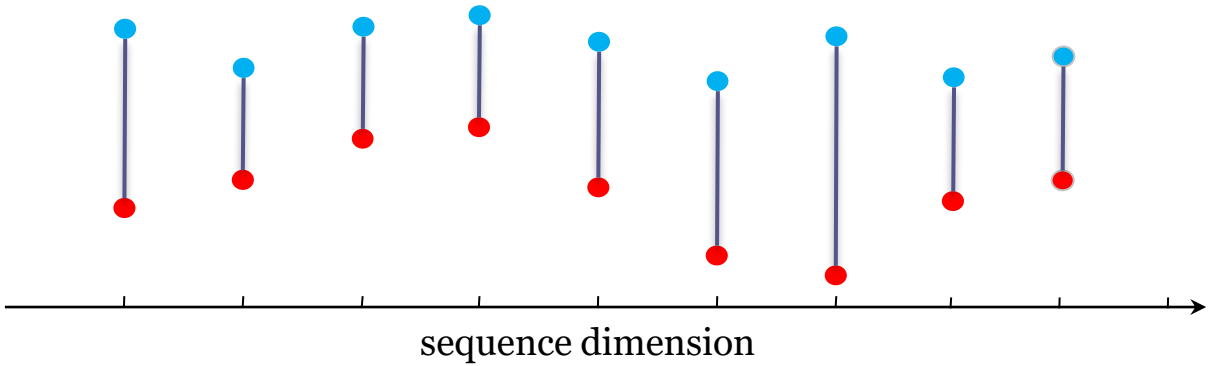


- subsequence
  - subset of contiguous values
  - eg, subsequence of length (dimensionality) 4

Data series Distance

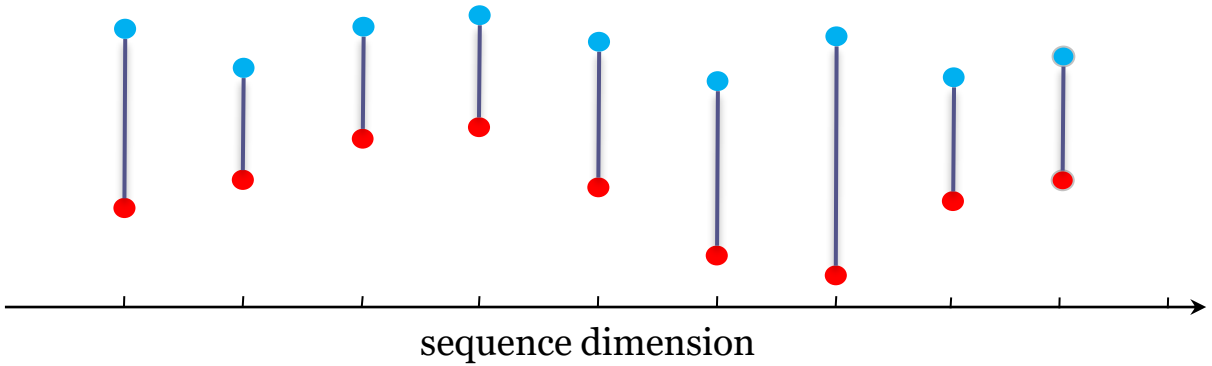


Data series Distance



- Euclidean distance
  - pair-wise point distance
  - $D(red, blue) = \sqrt{\sum_{i=1}^{n}(red_i - blue_i)^2}$

Data series Reconstruction Error



- Euclidean distance
  - pair-wise point distance
  - $D(red, blue) = \sqrt{\sum_{i=1}^{n}(red_i - blue_i)^2}$

Outline

- terminology and definitions
- motivation
- pre-processing tasks
- data series representation techniques

Simple Query Answering

select values in time interval

select values in some range

select some data series

combinations of those

Analysis Tasks

Clustering

Outlier Detection

Classification

Frequent Pattern Mining

Analysis Tasks

- analyze evolution of values across x-dimension
- identify trends

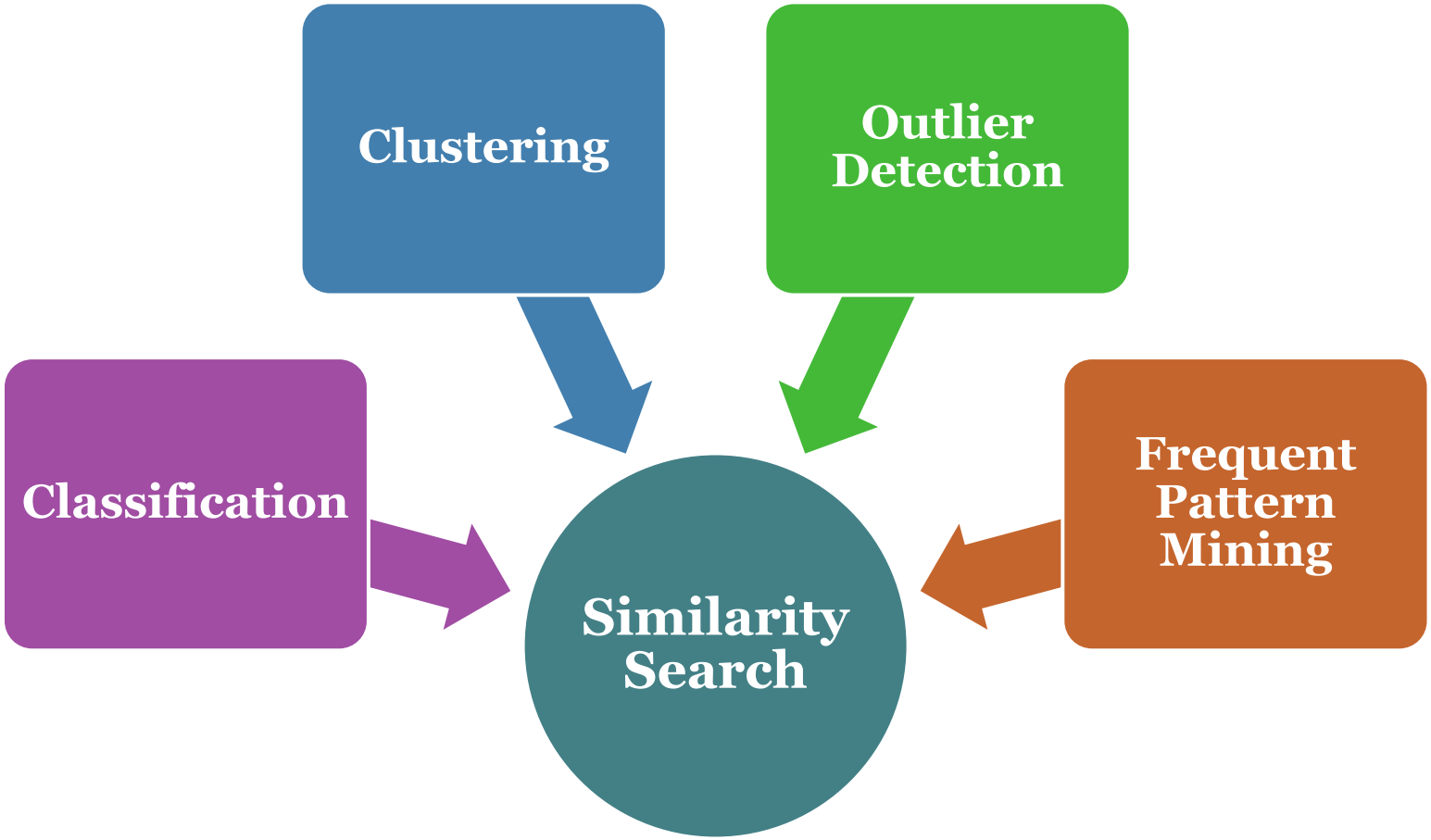
- treat data series as a first class citizen
  - analyze each data series as a single object
  - process all n-dimensions at once

Analysis Tasks
Subsequences

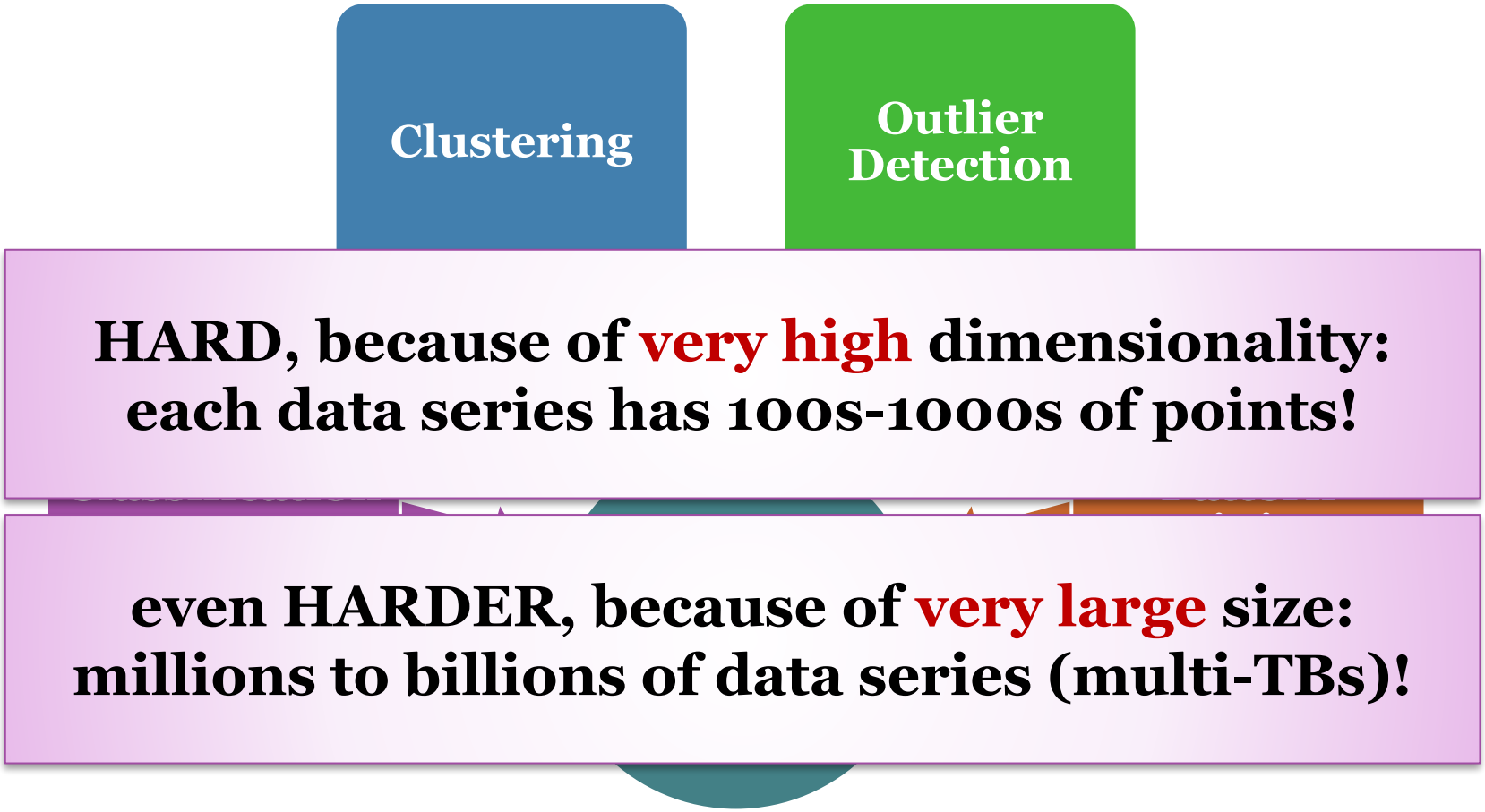
- often times the data series are very long
  - n >> 1
  - streaming data series

- we then chop the long sequence in subsequences
  - e.g., using sliding window, or shifting window
  - pick carefully length of subsequence
    - should contain patterns of interest
- and process each subsequence separately

Complex Analytics



Complex Analytics



Outline

- terminology and definitions
- motivation
- pre-processing tasks
- data series representation techniques

Pre-Processing
z-Normalization

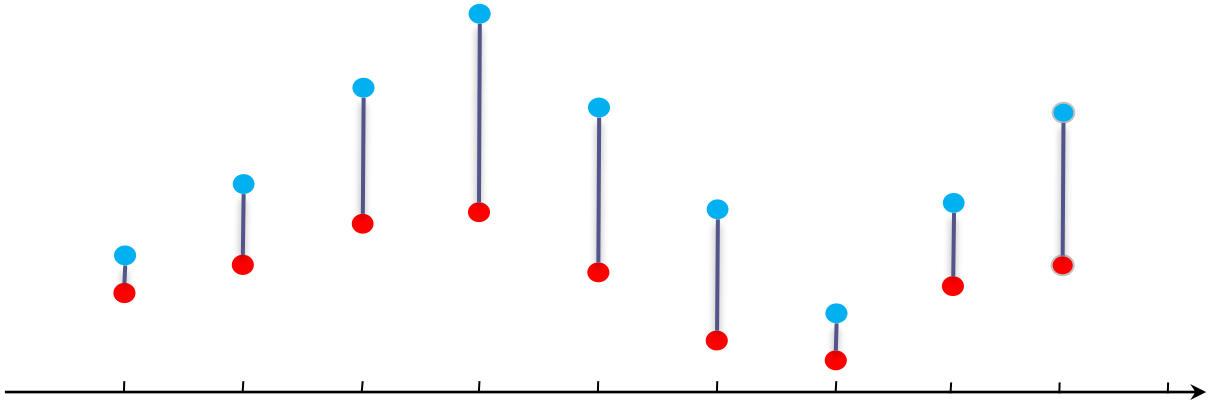
- data series encode trends
- usually interested in identifying similar trends

Pre-Processing
z-Normalization

- data series encode trends
- usually interested in identifying similar trends

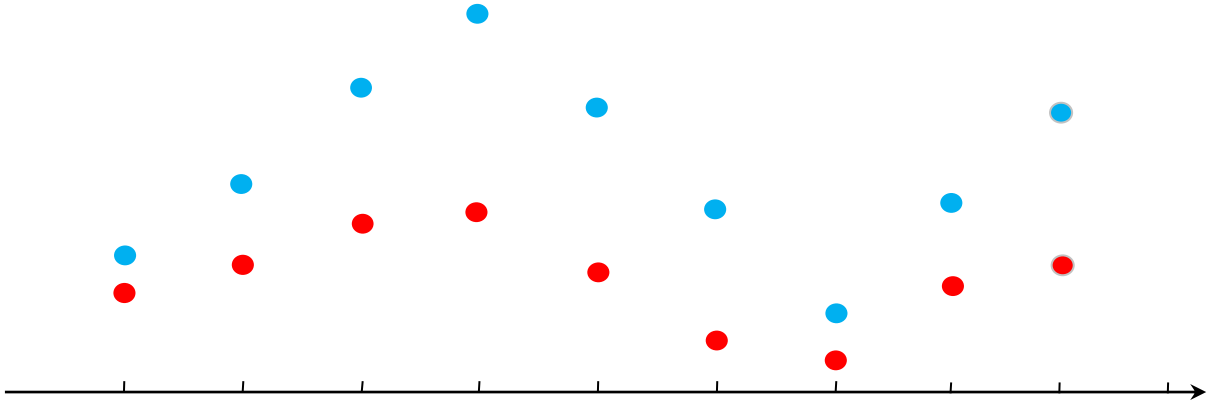
- but absolute values may mask this similarity

Pre-Processing
z-Normalization



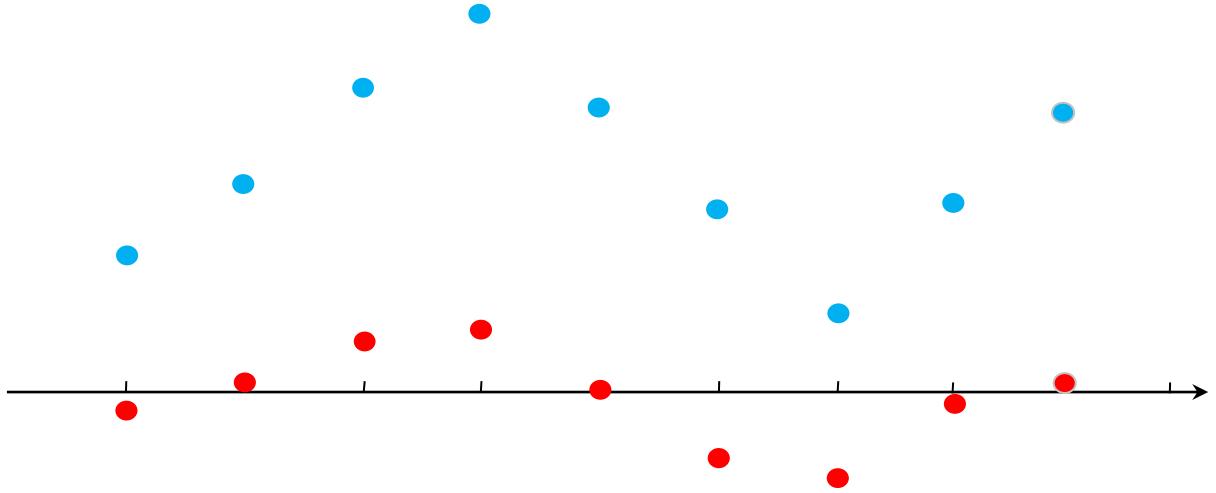
- two data series with similar trends
- but large distance…

Pre-Processing
z-Normalization



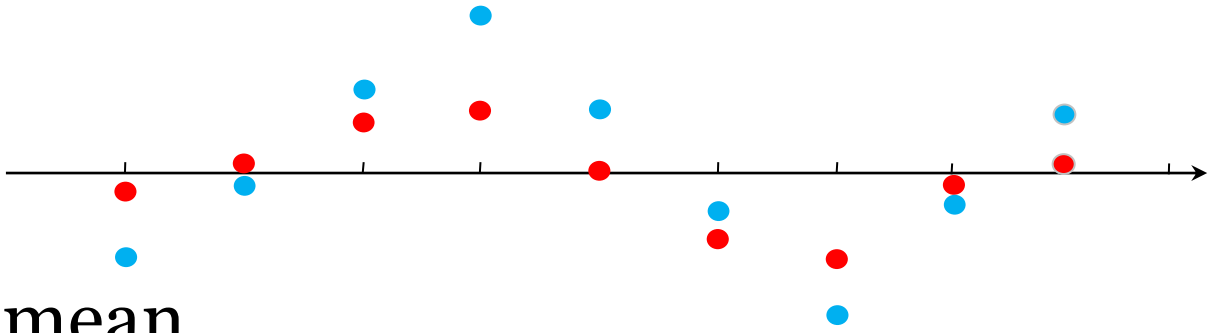
- zero mean
  - compute the mean of the sequence
  - subtract the mean from every value of the sequence

Pre-Processing
z-Normalization



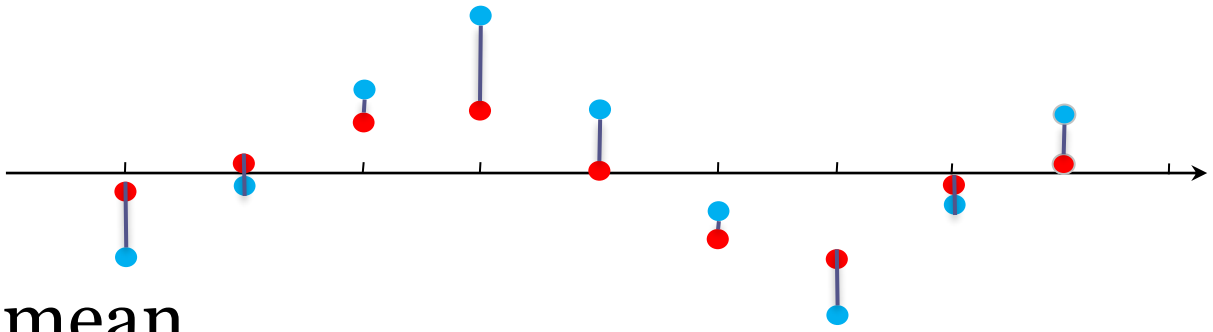
- zero mean
  - compute the mean of the sequence
  - subtract the mean from every value of the sequence

Pre-Processing
z-Normalization



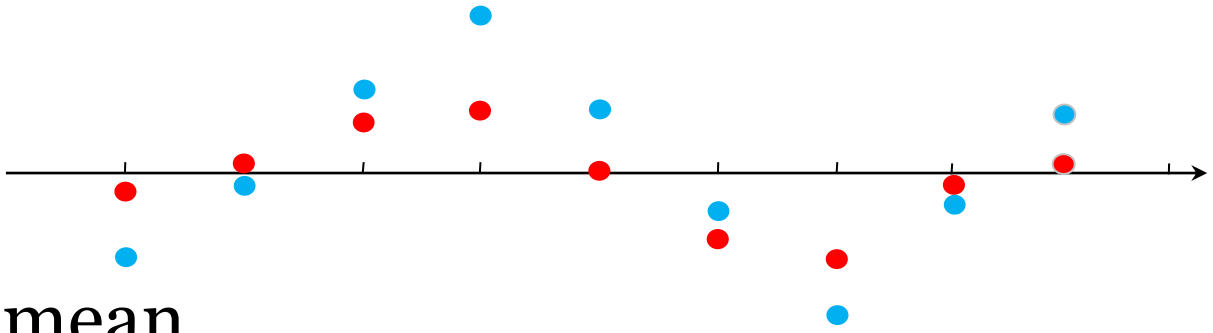
- zero mean
  - compute the mean of the sequence
  - subtract the mean from every value of the sequence

Pre-Processing
z-Normalization



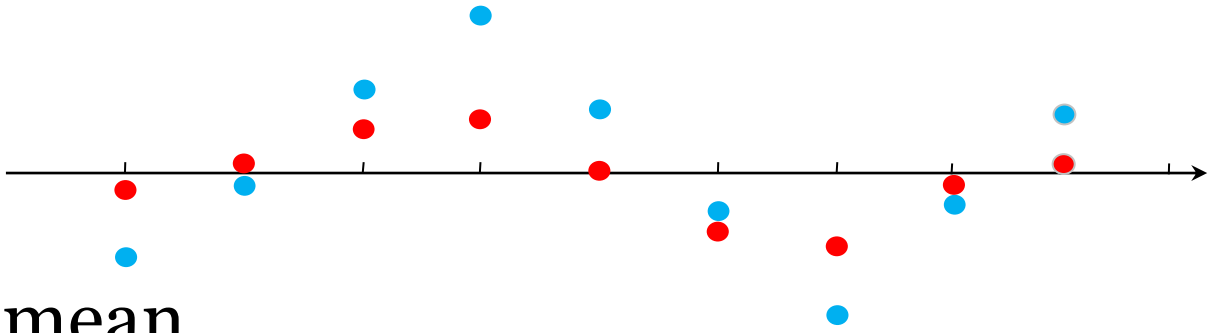
- zero mean
  - compute the mean of the sequence
  - subtract the mean from every value of the sequence

Pre-Processing
z-Normalization



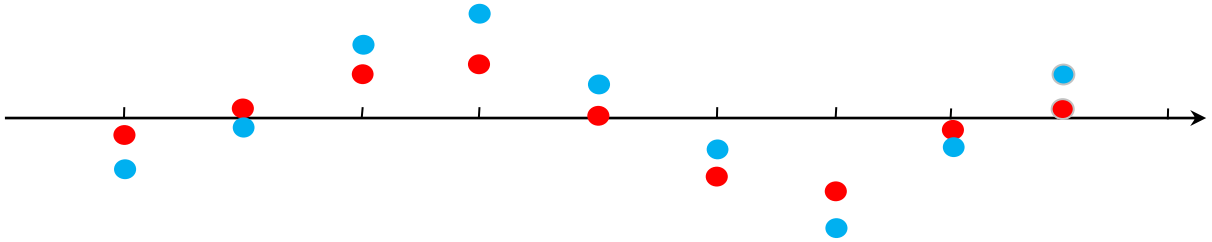
- zero mean
- standard deviation one
  - compute the standard deviation of the sequence
  - divide every value of the sequence by the stddev

Pre-Processing
z-Normalization



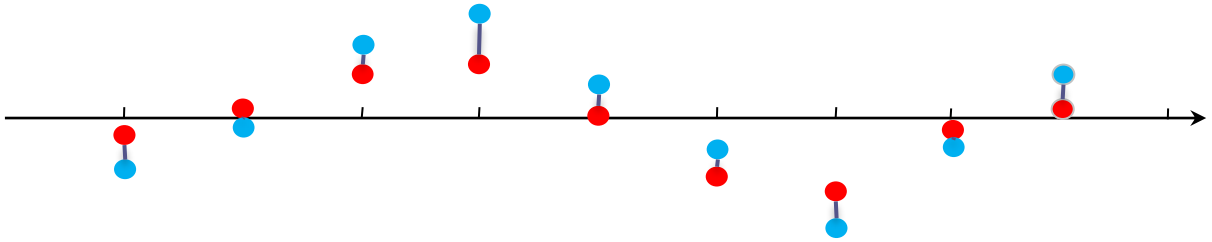
- zero mean
- standard deviation one
  - compute the standard deviation of the sequence
  - divide every value of the sequence by the stddev

Pre-Processing
z-Normalization



- zero mean
- standard deviation one
  - compute the standard deviation of the sequence
  - divide every value of the sequence by the stddev

Pre-Processing
z-Normalization



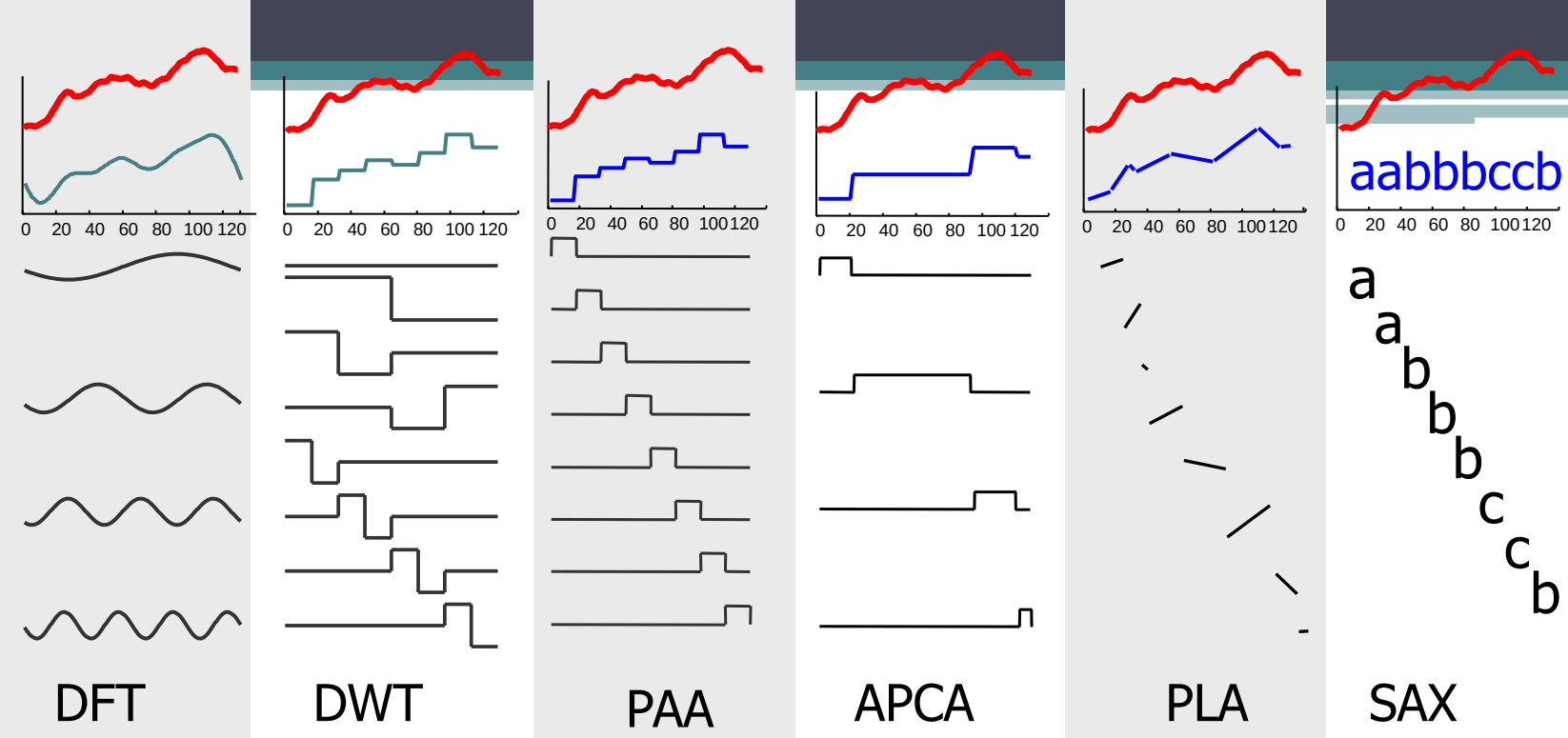
- zero mean
- standard deviation one

Pre-Processing z-Normalization

- when to z-normalize
  - interested in trends
- when not to z-normalize
  - interested in absolute values

Outline

- terminology and definitions
- motivation
- pre-processing tasks
- data series representation techniques



Agrawal, Faloutsos, & Swami. FODO 1993
Faloutsos, Ranganathan, & Manolopoulos. SIGMOD 1994

Chan & Fu. ICDE 1999

Keogh, Chakrabarti, Pazzani & Mehrotra KAIS 2000
Yi & Faloutsos VLDB 2000

Keogh, Chakrabarti, Pazzani & Mehrotra SIGMOD 2001

Morinaka, Yoshikawa, Amagasa, & Uemura, PAKDD 2001

for a complete and detailed presentation, see tutorial:

Publications

Keogh - KDD'04

Publications

- Palpanas et al. ICDE'04
- Palpanas et al. TKDE'08
- Shieh et al. KDD'08

Comparison of Representations

- which representation is the best?
- depends on data characteristics
  - periodic, smooth, spiky, …
- overall (averaged over many diverse datasets, using same memory budget), when measuring reconstruction error (RMSE)
  - no big differences among methods
  - DFT, PAA, DWT (Haar), iSAX slightly better
- should also take into account other factors
  - visualization, indexable, …

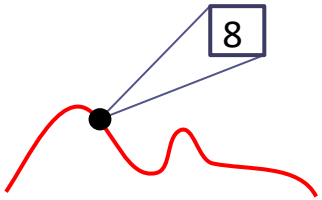
Questions?

Data Series Similarity

Data Series Similarity Problem Variations

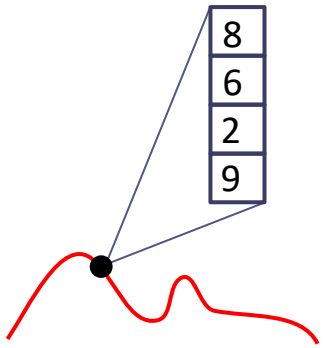
Problem Variations

Series



Univariate

each point represents one value (e.g., temperature)

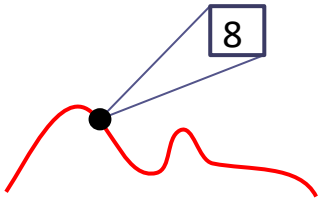


Multivariate

each point represents many values (e.g., temperature, humidity, pressure, wind, etc.)

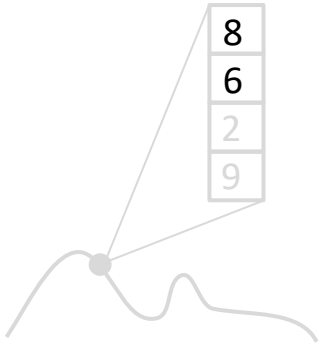
Problem Variations

Series



Univariate

each point represents one value (e.g., temperature)



Multivariate

each point represents many values (e.g., temperature, humidity, pressure, wind, etc.)

Publications

Ding-PVLDB'08

Paparrizos-SIGMOD'20

Problem Variations

Distance Measures

- similarity search is based on measuring distance between sequences
- dozens of distance measures have been proposed
  - lock-step
    - Minkowski, Manhattan, Euclidean, Maximum, DISSIM, …
  - sliding
    - Normalized Cross-Correlation, SBD, …
  - elastic
    - DTW, LCSS, MSM, EDR, ERP, Swale, …
  - kernel-based
    - KDTW, GAK, SINK, …
  - embedding
    - GRAIL, RWS, SPIRAL, …

Publications

Ding-PVLDB'08

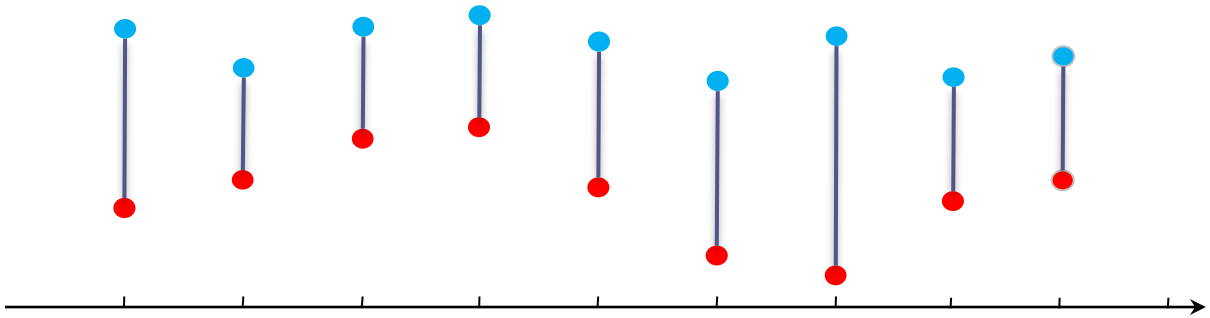
Paparrizos-SIGMOD'20

Problem Variations

Distance Measures

- similarity search is based on measuring distance between sequences
- dozens of distance measures have been proposed
  - lock-step
    - Minkowski, Manhattan, Euclidean, Maximum, DISSIM, …
  - sliding
    - Normalized Cross-Correlation, SBD, …
  - elastic
    - DTW, LCSS, MSM, EDR, ERP, Swale, …
  - kernel-based
    - KDTW, GAK, SINK, …
  - embedding
    - GRAIL, RWS, SPIRAL, …

Euclidean Distance



- Euclidean distance
  - pair-wise point distance

$$ED(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Correlation

- measures the degree of relationship between data series
  - indicates the degree and direction of relationship
- direction of change
  - positive correlation
    - values of two data series change in same direction
  - negative correlation
    - values of two data series change in opposite directions
- linear correlation
  - amount of change in one data series bears constant ratio of change in the other data series

- useful in several applications

Pearson's Correlation Coefficient

- used to see linear dependency between values of data series of equal length, n

$$PC = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s_x}\right)\left(\frac{y_i - \bar{y}}{s_y}\right)$$

- where $\bar{x}$ is the mean: $\bar{x} = \frac{1}{n-1}\sum_{i=1}^{n} x_i$
- and s_x is the standard deviation: $s_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$

Pearson's Correlation Coefficient

- used to see linear dependency between values of data series of equal length, n

$$PC = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s_x}\right)\left(\frac{y_i - \bar{y}}{s_y}\right)$$

- takes values in [-1,1]
  - 0 – no correlation
  - -1, 1 – inverse/direct correlation

- there is a statistical test connected to PC, where null hypothesis is the no correlation case (correlation coefficient = 0)
  - test is used to ensure that the correlation similarity is not caused by a random process

PC and ED

- Euclidean distance: $ED = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$,
- In case of Z-normalized data series (mean = 0, stddev = 1):

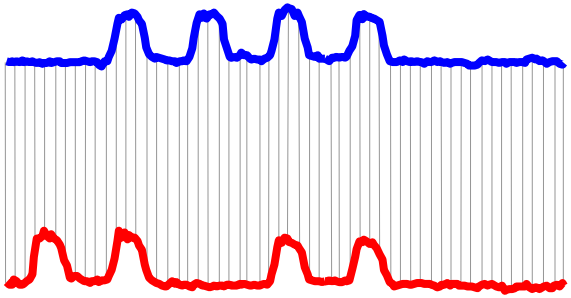
$$PC = \frac{1}{n-1}\sum_{i=1}^{n} x_i \cdot y_i \quad \text{and} \quad ED^2 = 2n(n-1) - 2\sum_{i=1}^{n} x_i y_i$$

so the following formula is true: $ED^2 = 2(n-1)(n-PC)$

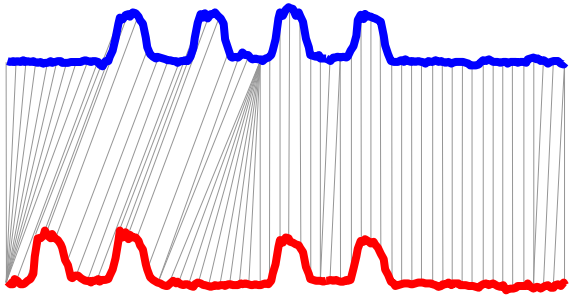
- direct connection between ED and PC for Z-normalized data series
  - if ED is calculated for normalized data series, it can be directly used to calculate the p-value for statistical test of Pearson's correlation instead of actual PC value.

Distance Measures: LCSS against Euclidean, DTW

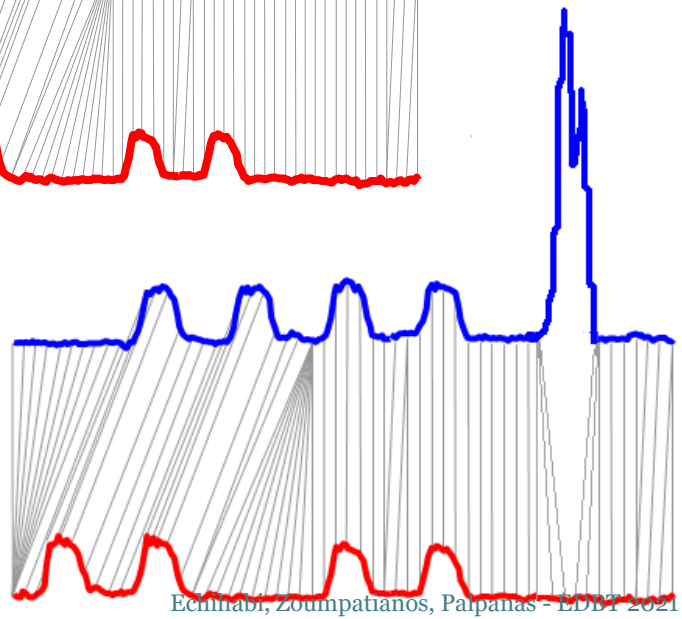
- Euclidean
  - rigid



- Dynamic Time Warping (DTW)
  - allows local scaling

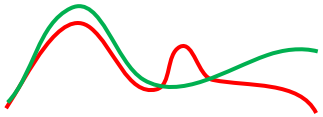


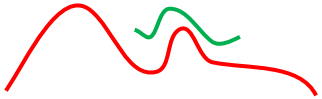
- Longest Common SubSequence (LCSS)
  - allows local scaling
  - ignores outliers



Problem Variations

Queries





Whole matching

Entire query

Entire candidate

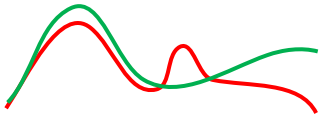
Subsequence matching

Entire query

A subsequence of a candidate

Problem Variations

Queries



Whole matching

Entire query

Entire candidate

Subsequence matching

Entire query

A subsequence of a candidate

Problem Variations

Queries

Nearest Neighbor (1NN)
k-Nearest Neighbor (kNN)
Farthest Neighbor
epsilon-Range

and more…

Similarity Matching

- given a data series collection D and a query data series q, return the data series from D that are the most similar to q
  - there exist different flavors of this basic operation
- basis for most data series analysis tasks

Similarity Matching
Nearest Neighbor (NN) Search

- given a data series collection D and a query data series q, return the data series from D that has the smallest distance to q

- result set contains one data series

Similarity Matching
Nearest Neighbor (NN) Search

- serial scan
  - compute the distance between q and every $d_i \in D$
  - return d_i with the smallest distance to q

Similarity Matching
Nearest Neighbor (NN) Search

- serial scan
  - bsf = Inf // best so far distance
  - for every $d_i \in D$
    - compute distance, dist, between d_i and q
    - if this dist less than bsf then bsf=dist
  - return d_i corresponding to bsf

Similarity Matching
k-Nearest Neighbors (kNN) Search

- given a data series collection D and a query data series q, return the k data series from D that have the k smallest distances to q

- result set contains k data series

Similarity Matching
k-Nearest Neighbors (kNN) Search

- serial scan
  - compute the distance between q and every $d_i \in D$
  - return the k d_i with the k smallest distances to q

Similarity Matching
k-Nearest Neighbors (kNN) Search

- serial scan
  - kbsf = Null // best so far max-heap of k elements
  - for every $d_i \in D$
    - compute distance, dist, between d_i and q
    - if this dist less than max of kbsf then insert dist in kbsf
  - return k d_i corresponding to k elements in kbsf

Similarity Matching
ε-Range Search

- given a data series collection D and a query data series q, return all data series from D that are within distance ε from q

- result set contains [?] data series

Similarity Matching
ε-Range Search

- serial scan
  - compute the distance between q and every $d_i \in D$
  - return all d_i with distance less than ε to q

Similarity Matching
ε-Range Search

- serial scan
  - res = {} // empty result set
  - for every $d_i \in D$
    - compute distance, dist, between d_i and q
    - if this dist less than ε then insert dist in res
  - return all d_i corresponding to elements in res

Problem Variations

Queries

- Nearest Neighbor (1NN)
- k-Nearest Neighbor (kNN)
- Farthest Neighbor
- epsilon-Range
- And more…

Nearest Neighbor (NN) Queries…

Publications

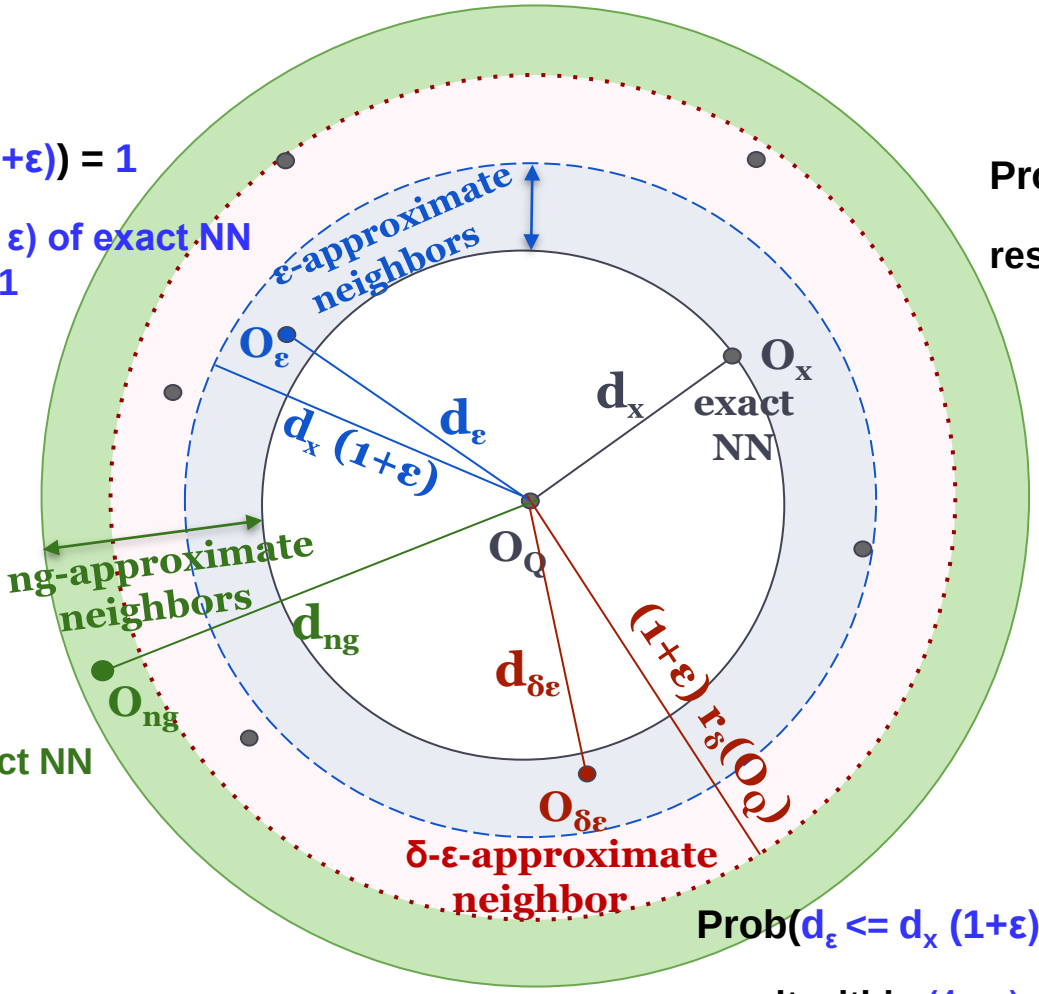
Echihabi et al. PVLDB'19

Prob($d_\varepsilon <= d_x\ (1+\varepsilon)$) = 1

result within (1+ ε) of exact NN with probability 1

Prob($d_x = \min\{d_i\}$) = 1

result is exact NN



Prob(d_{ng} <>= ?) = ?

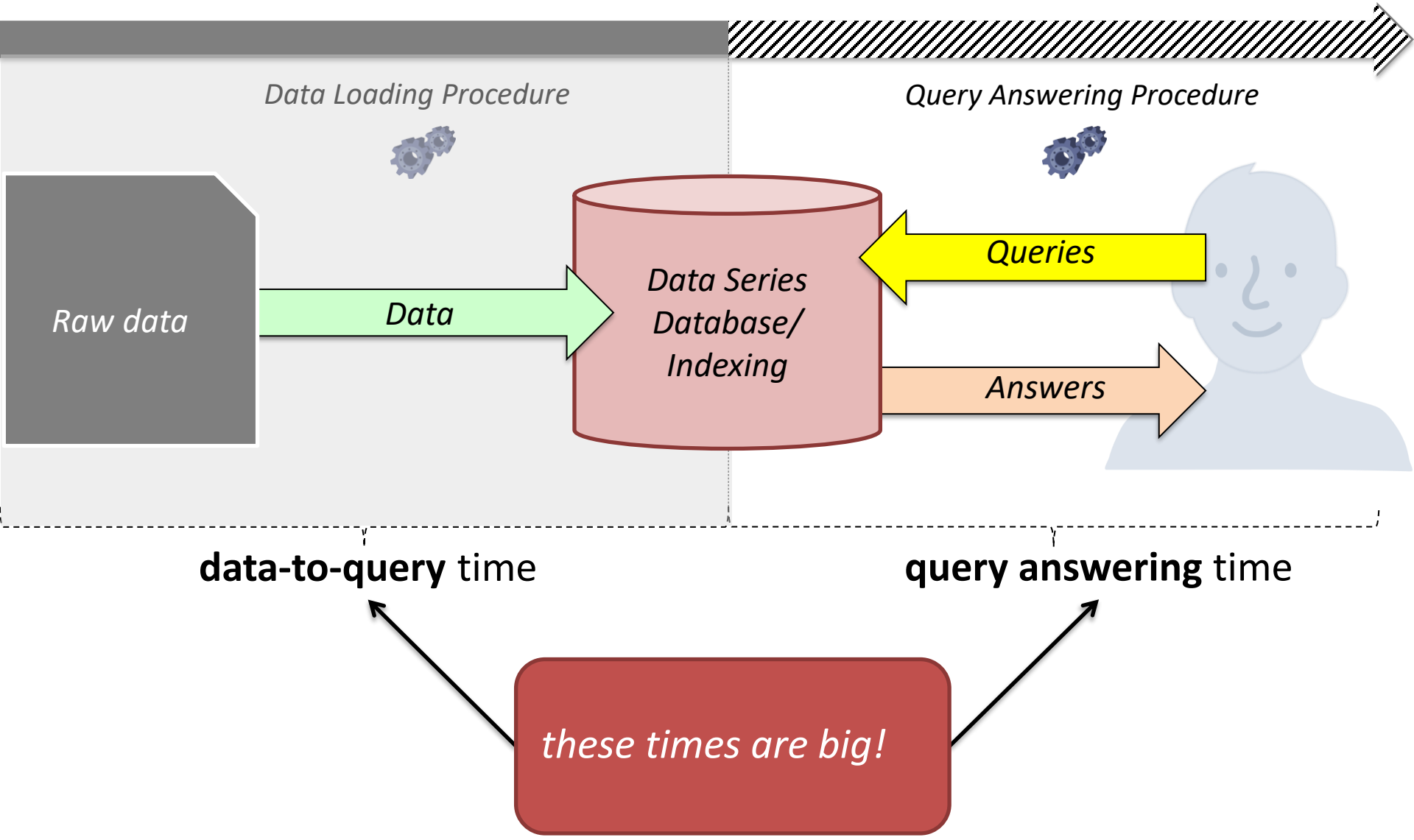
result within ? of exact NN

Prob($d_\varepsilon <= d_x\ (1+\varepsilon)$) >= δ

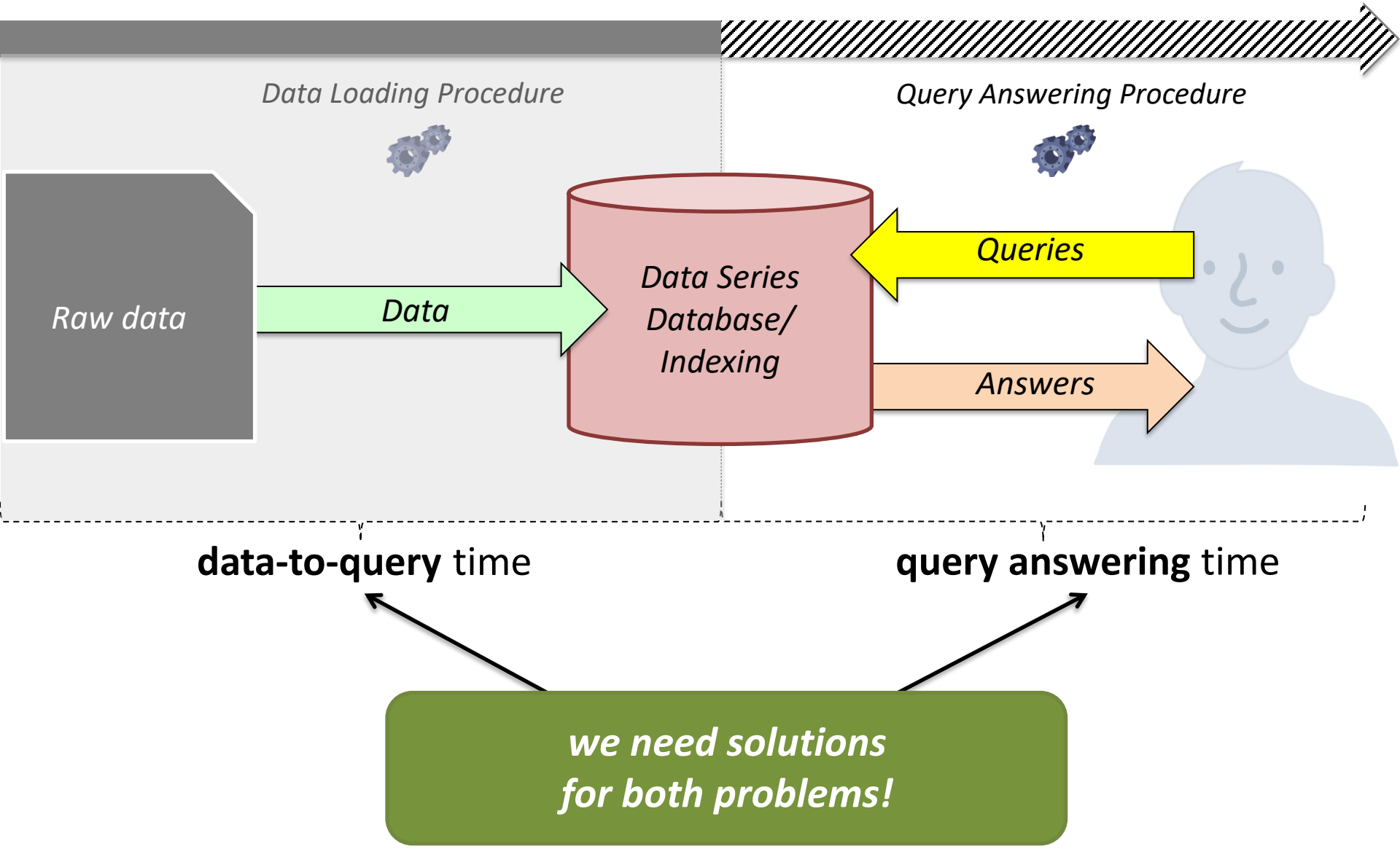
result within (1+ ε) of exact NN with probability at least δ

Data Series Similarity Query Answering

Query answering process



Query answering process



Similarity Matching
Fast Euclidean Distance

- similarity matching requires many distance computations
  - can significantly slow down processing
    - because of large number of data series in the collection
    - because of high dimensionality of each data series
- in case of Euclidean Distance, we can speedup processing by
  - smart implementation of distance function
  - early abandoning
- result in considerable performance improvement

Publications

Keogh-DMKD'03

Similarity Matching
Fast Euclidean Distance

- smart implementation of distance function

$$ED(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Publications

Keogh-DMKD'03

Similarity Matching
Fast Euclidean Distance

- smart implementation of distance function
  - do not compute the square root (of the Euclidean Distance)

$$ED(X,Y) = \sum_{i=1}^{n} (x_i - y_i)^2$$

- does not alter the results
- saves precious CPU cycles

Publications

Keogh-DMKD'03

Similarity Matching
Fast Euclidean Distance

- early abandoning
  - stop the distance computation as soon as it exceeds the value of bsf

$$ED(X,Y) = \sum_{i=1}^{m} (x_i - y_i)^2 , \qquad m \leq n$$

- does not alter the results
- avoids useless computations

GEMINI Framework

- Raw data: original full-dimensional space
- Summarization: reduced dimensionality space
- Searching in original space *costly*
- Searching in reduced space *faster*:
  - Less data, indexing techniques available, lower bounding
- Lower bounding enables us to
  - *prune search space:* throw away data series based on reduced dimensionality representation
  - *guarantee correctness* of answer
    - no false negatives
    - false positives filtered out based on raw data

GEMINI Framework

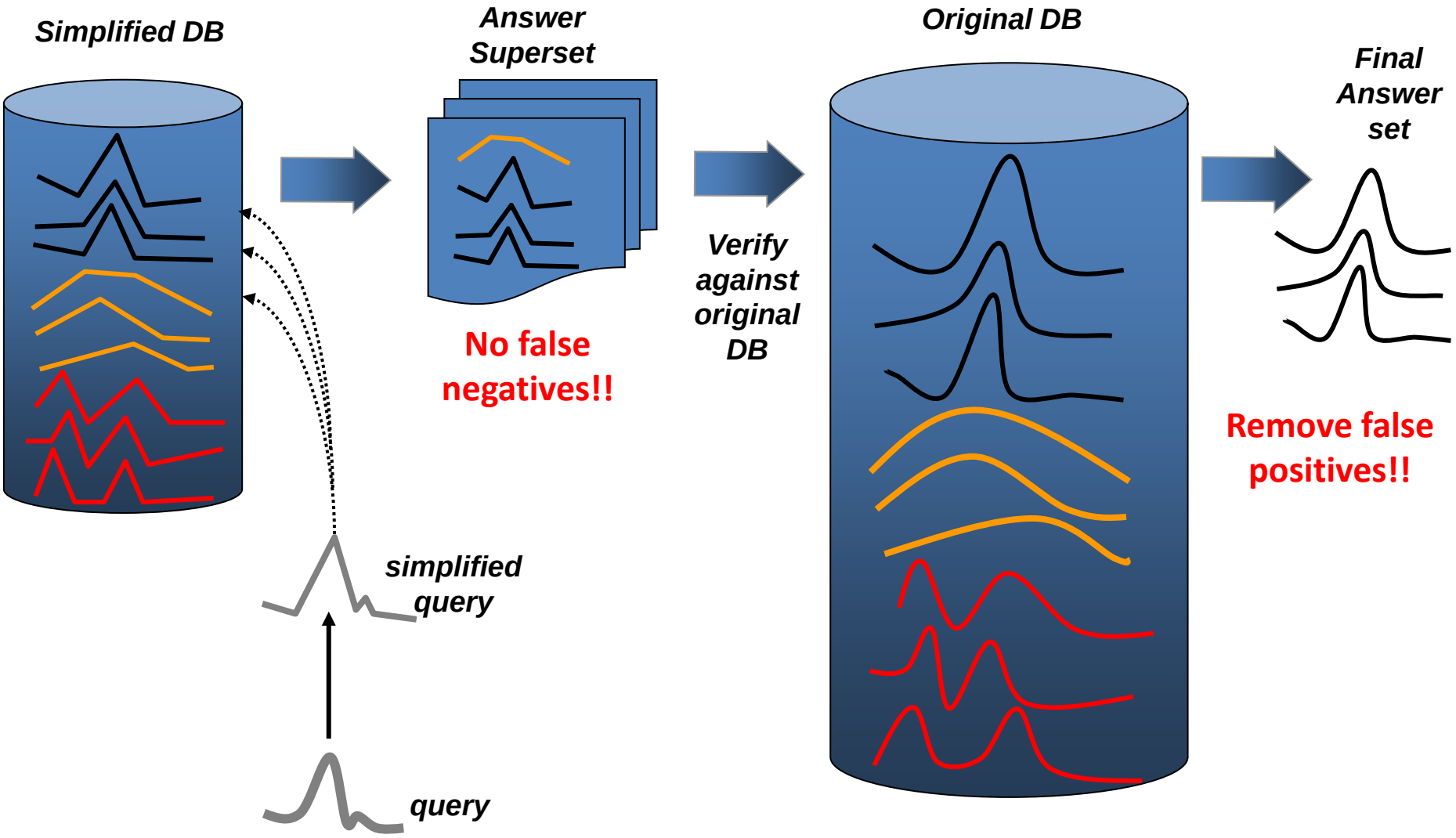
Publications

Faloutsos-SIGMOD'94

GEMINI Solution: Quick filter-and-refine:

- extract *m* features (numbers, e.g., average)
- map to point in *m*-dimensional feature space
- organize points
- retrieve the answer using a NN query
- discard false positives

Generic Search using Lower Bounding



GEMINI: contractiveness

Publications

Faloutsos-SIGMOD'94

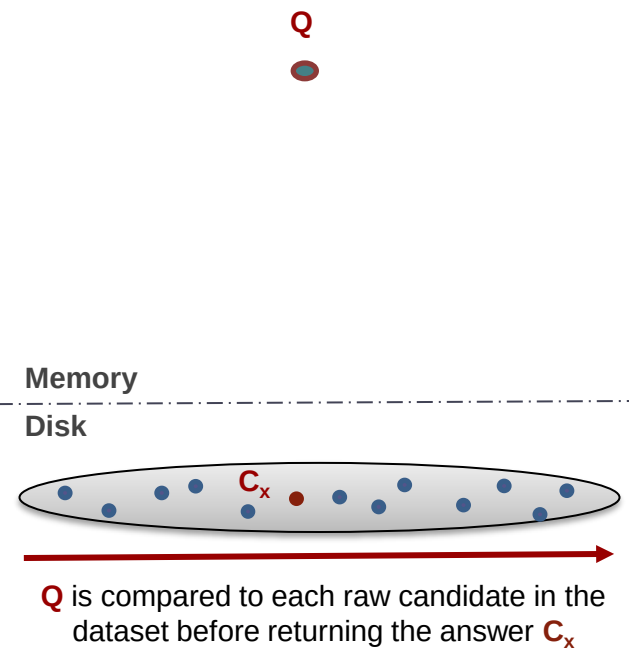
- GEMINI works when:

$$D_{feature}(F(x), F(y)) <= D(x, y)$$

- *Note that, the closer the feature distance to the actual one, the better*

Similarity Search
Classes of Methods

Similarity Matching Serial Scan



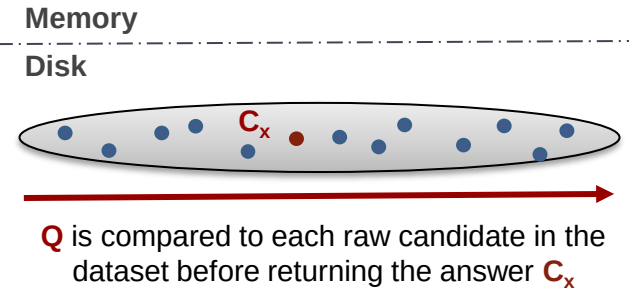
(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching Serial Scan

bsf = +∞

Q

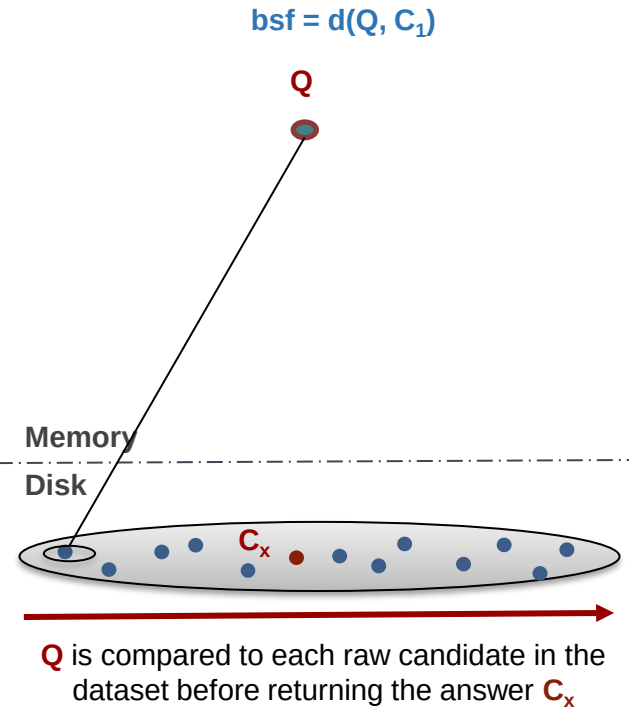


Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching Serial Scan

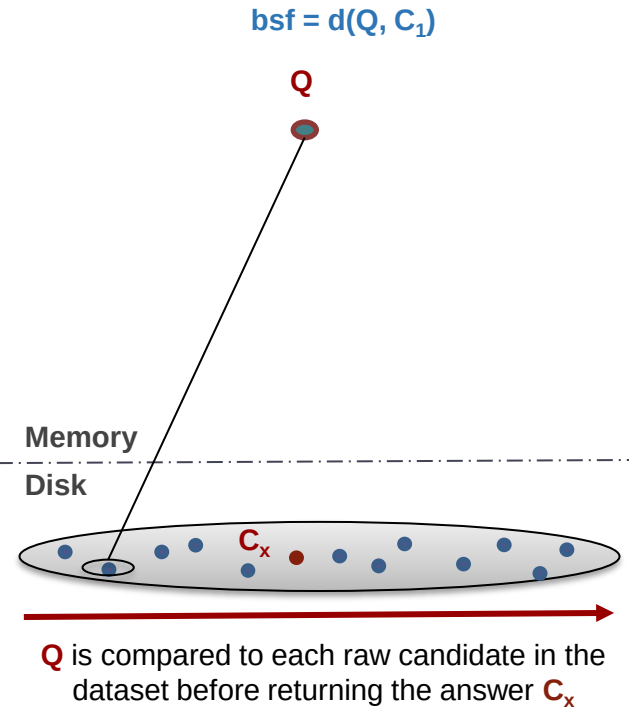


Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching Serial Scan

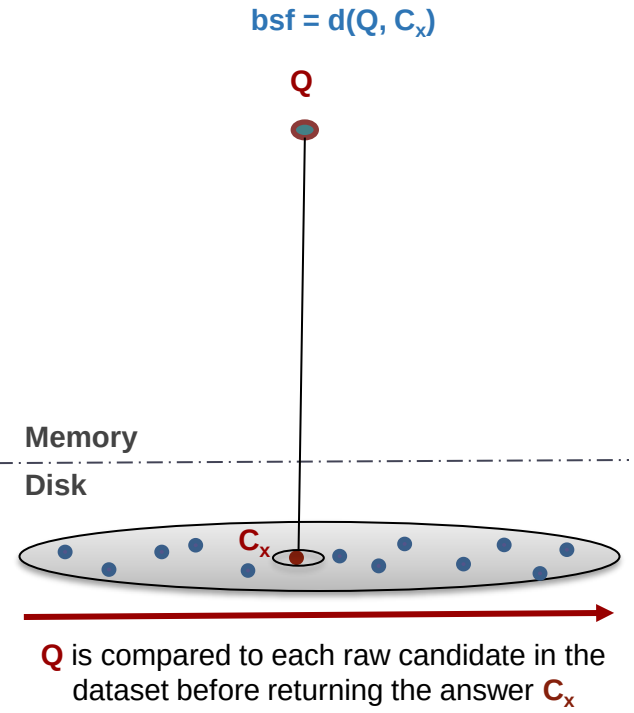


Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching Serial Scan

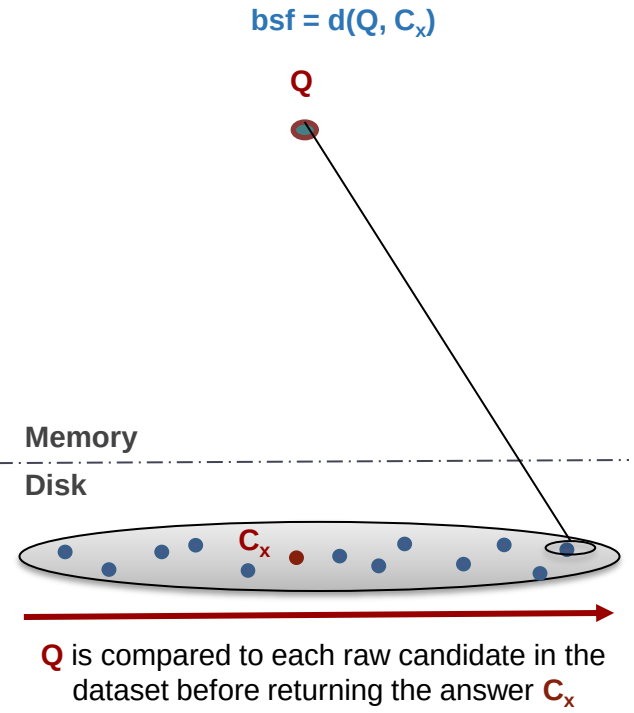


Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching Serial Scan

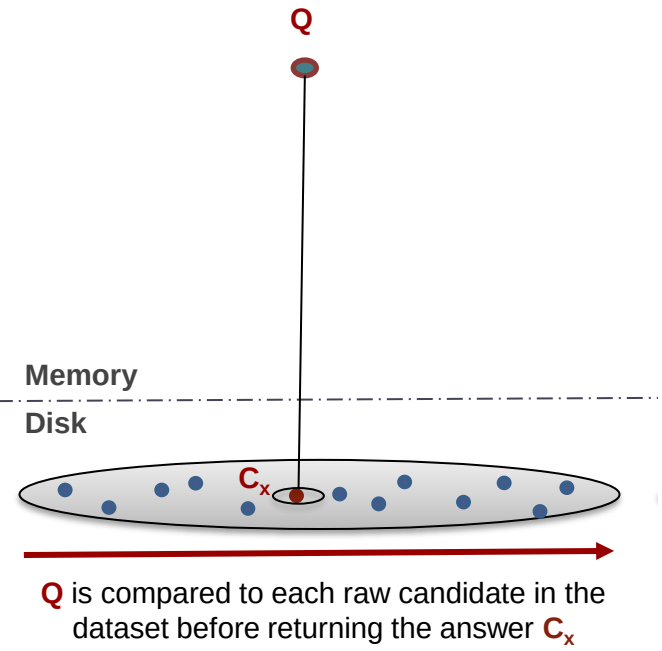


Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching Serial Scan

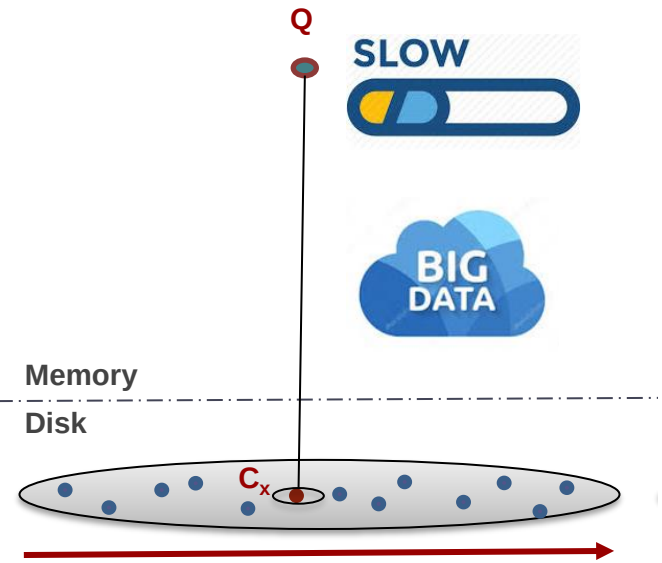


Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching Serial Scan

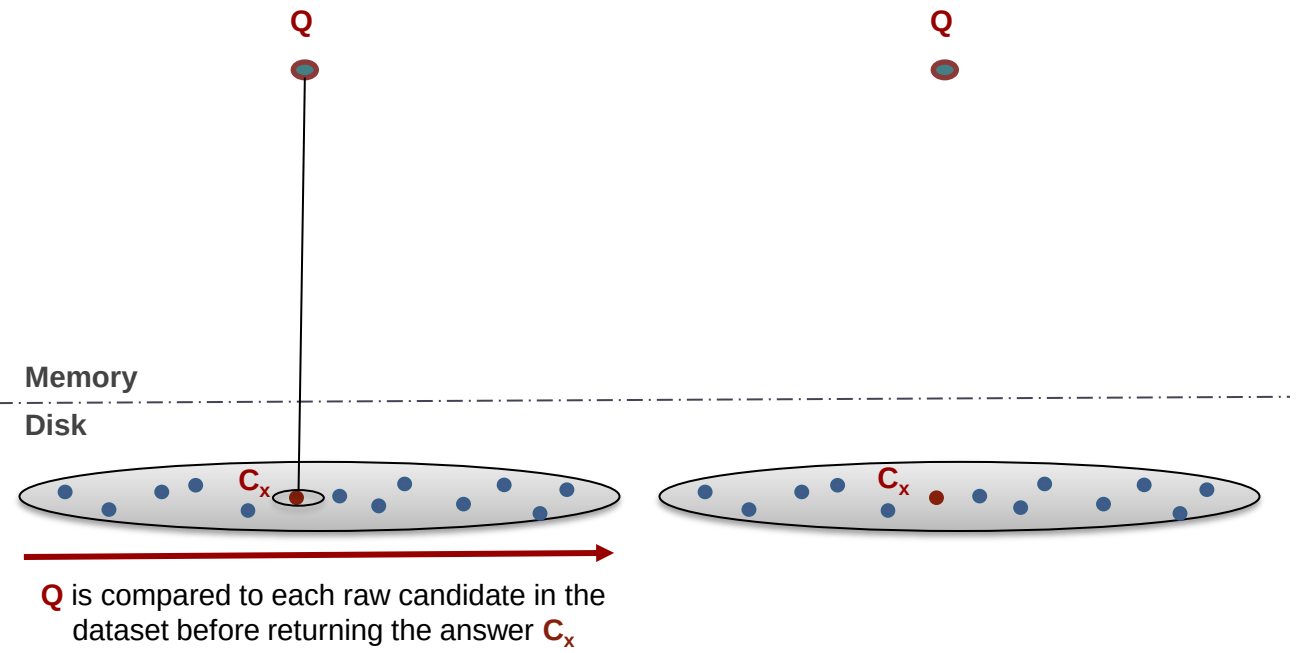


Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

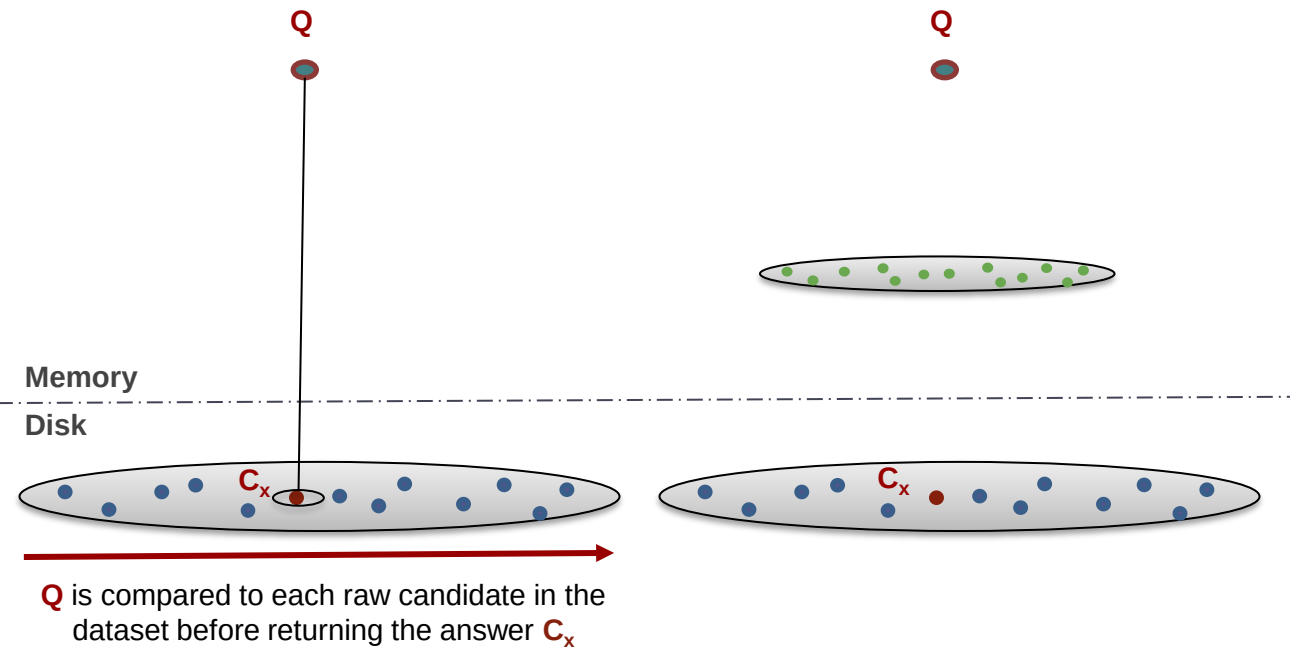
Answering a similarity search query using different access paths

Indexes vs. Scans



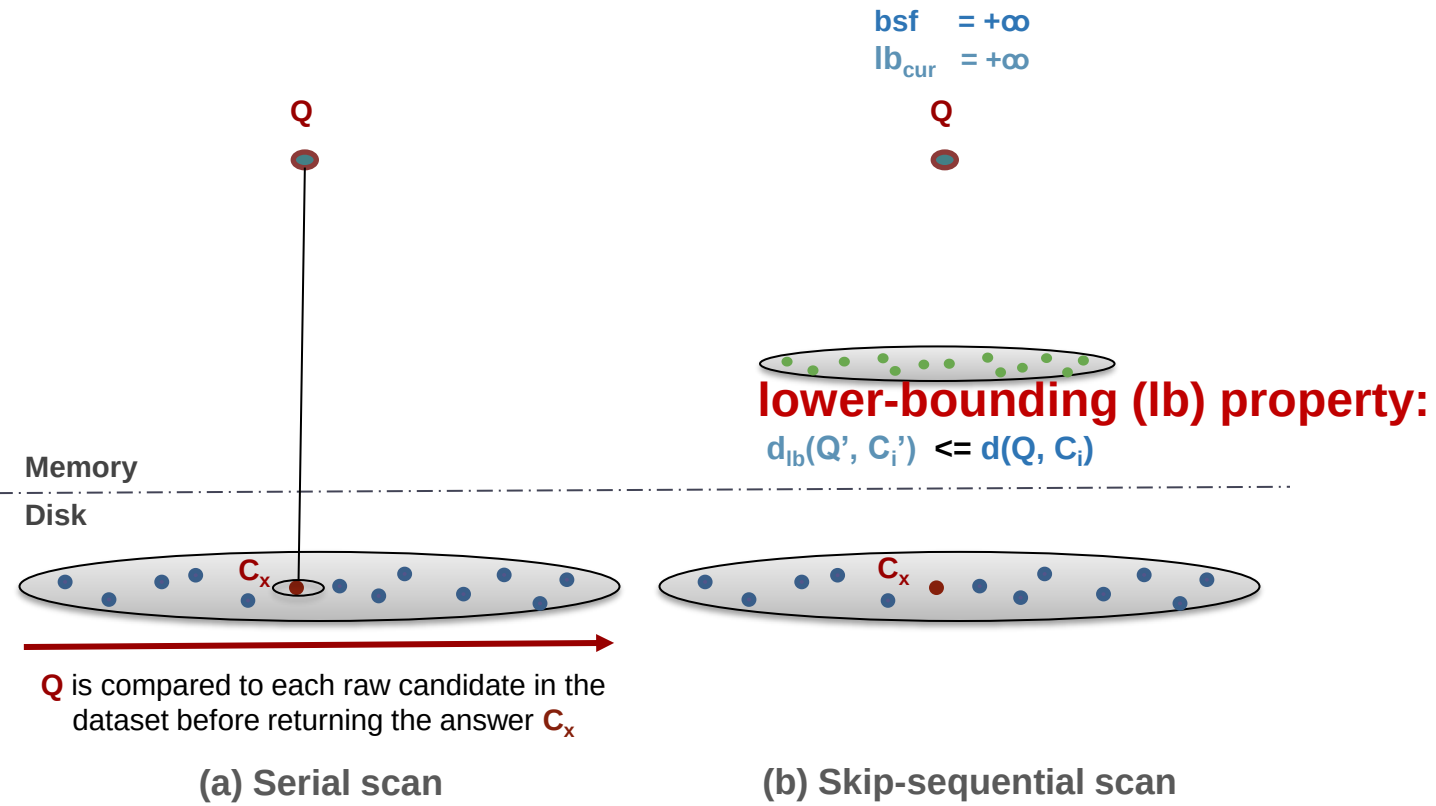
Answering a similarity search query using different access paths

Indexes vs. Scans



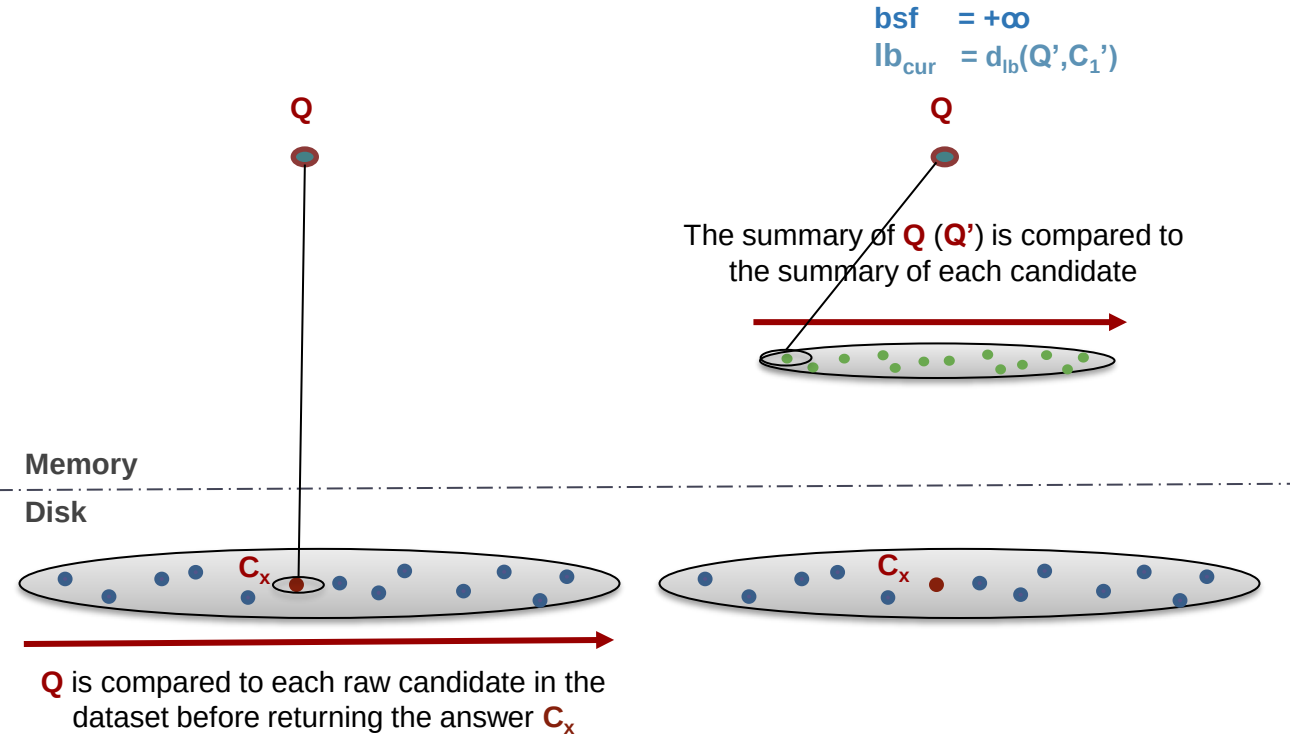
Answering a similarity search query using different access paths

Indexes vs. Scans



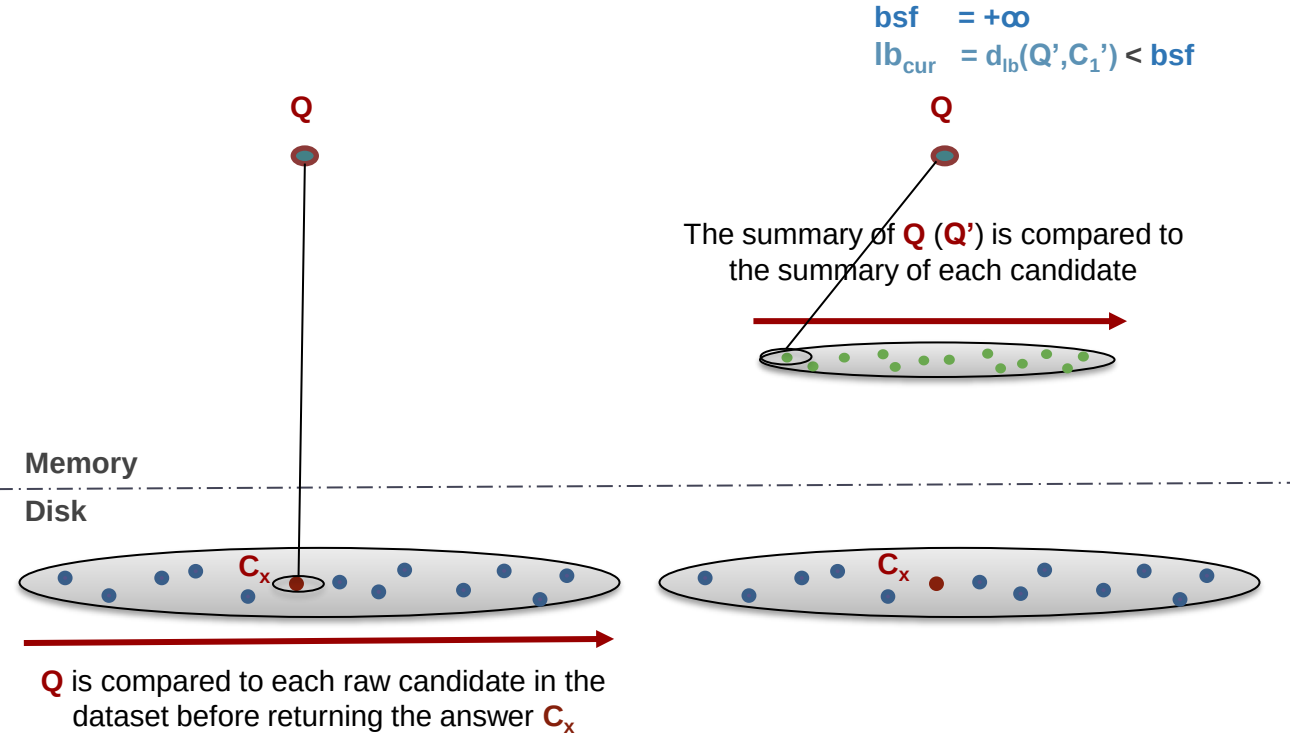
Answering a similarity search query using different access paths

Indexes vs. Scans



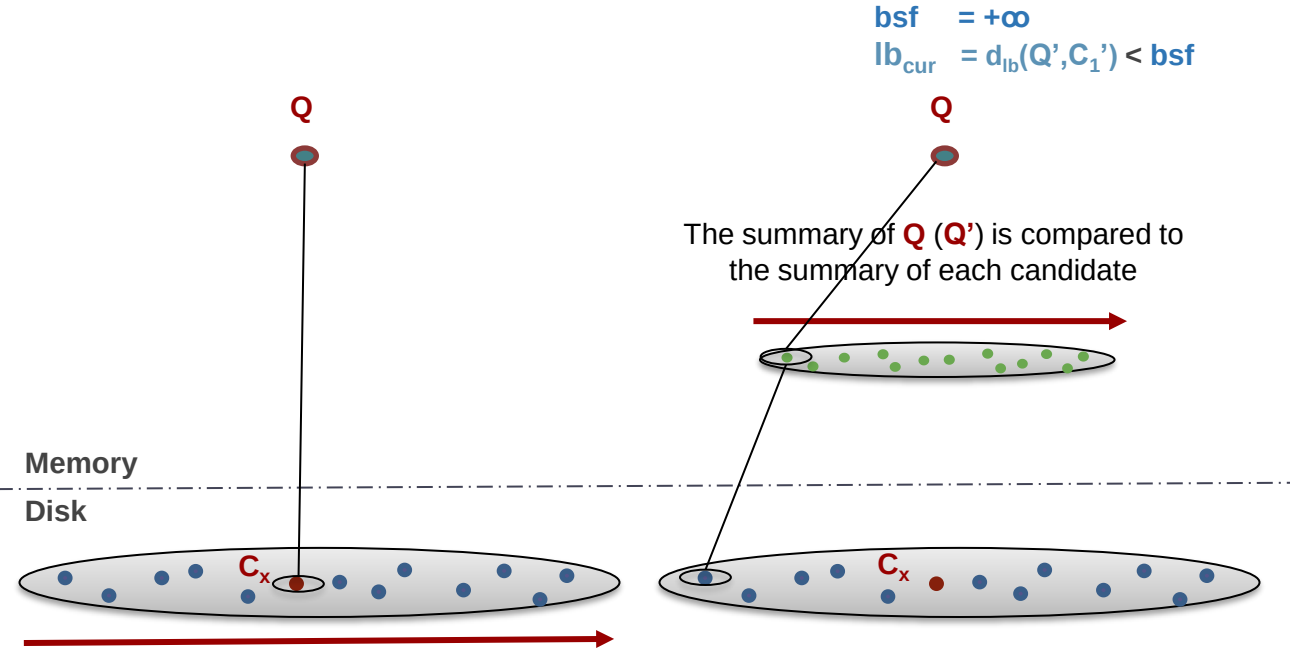
Answering a similarity search query using different access paths

Indexes vs. Scans



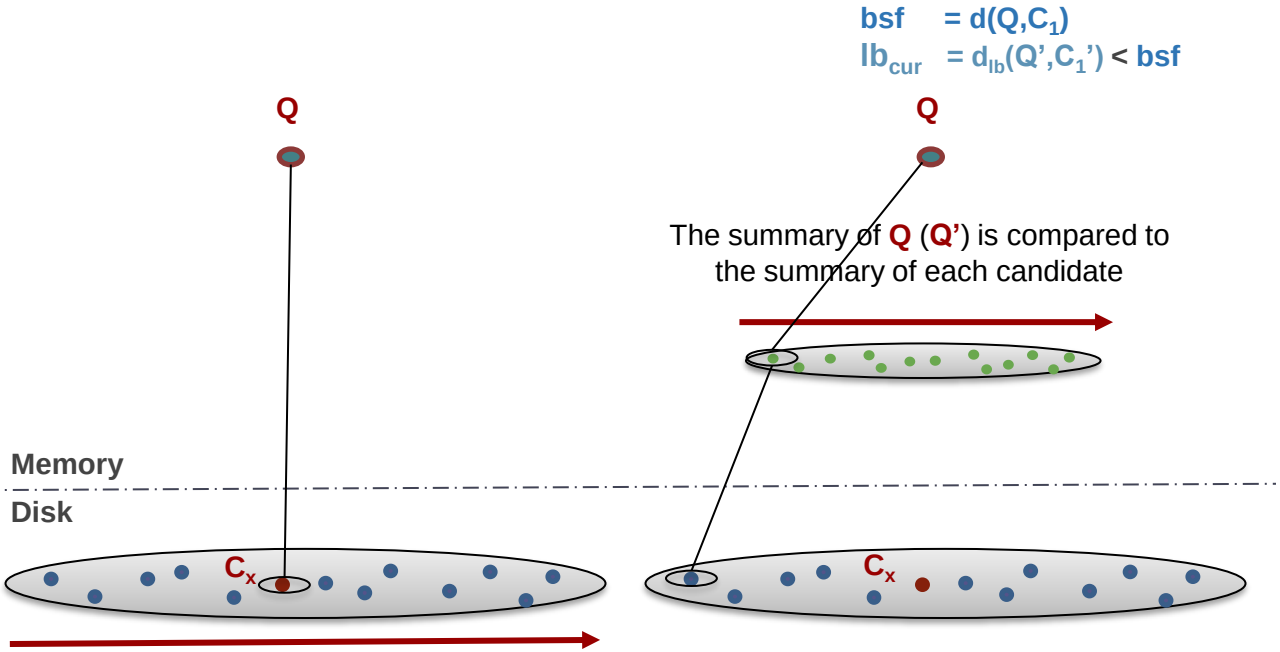
Answering a similarity search query using different access paths

Indexes vs. Scans



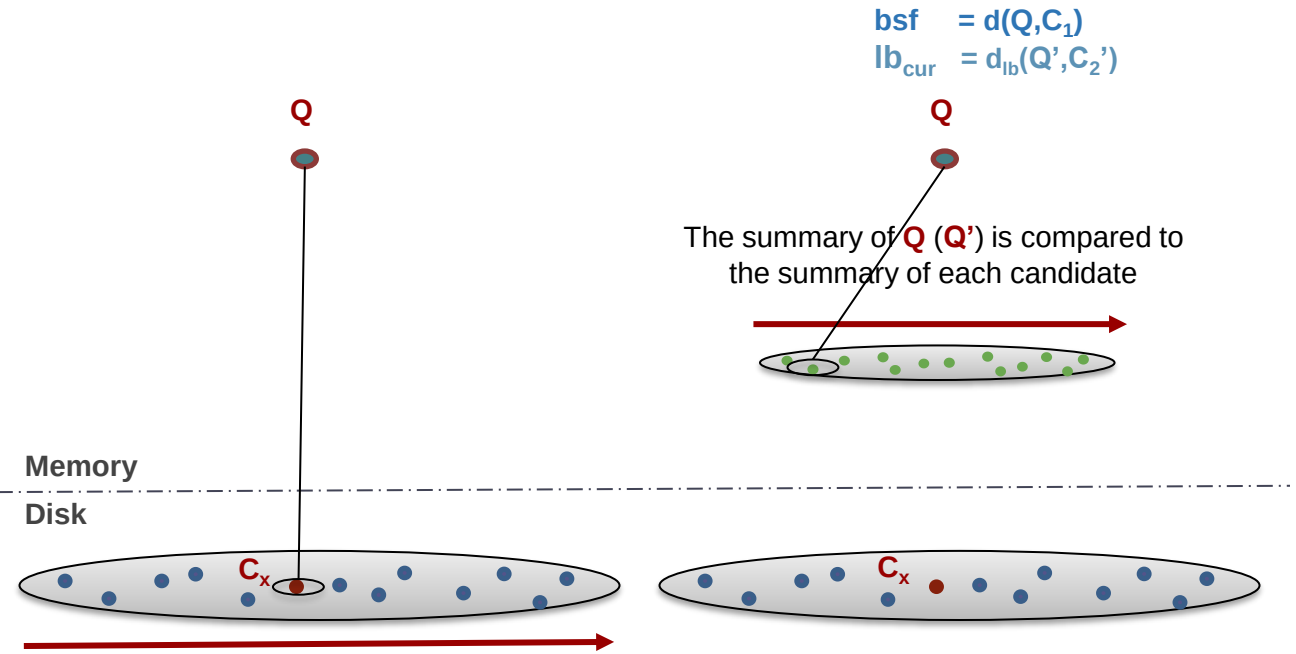
Answering a similarity search query using different access paths

Indexes vs. Scans



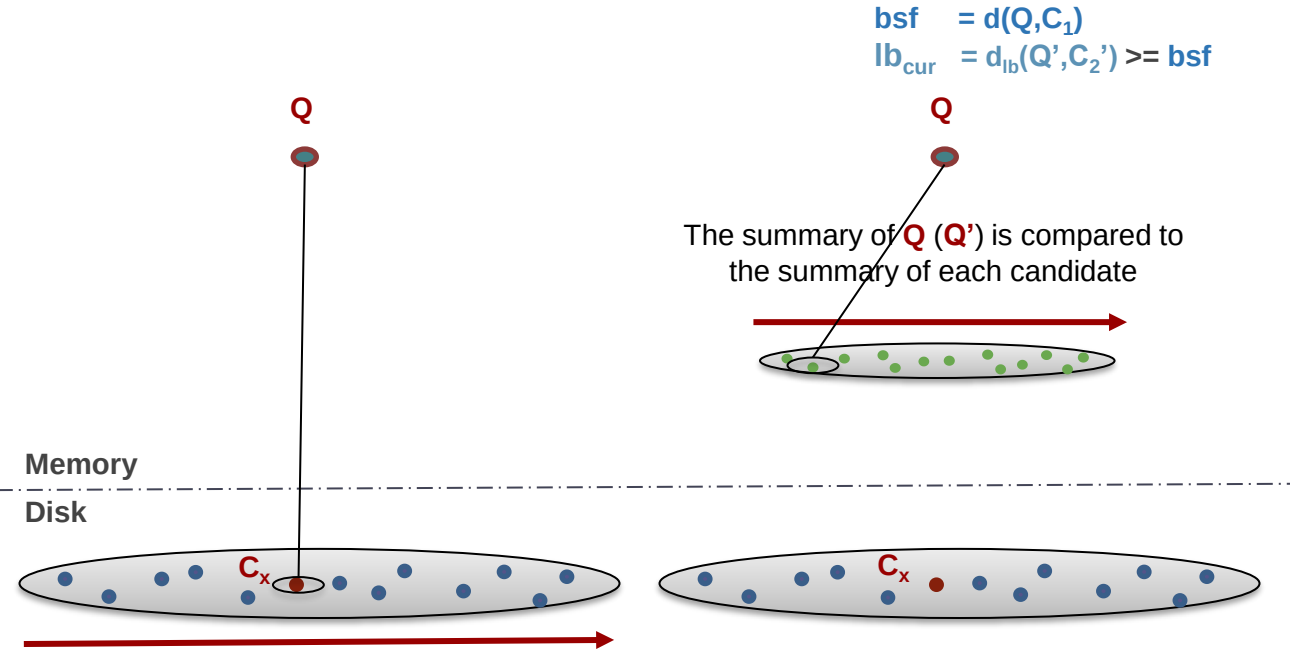
Answering a similarity search query using different access paths

Indexes vs. Scans



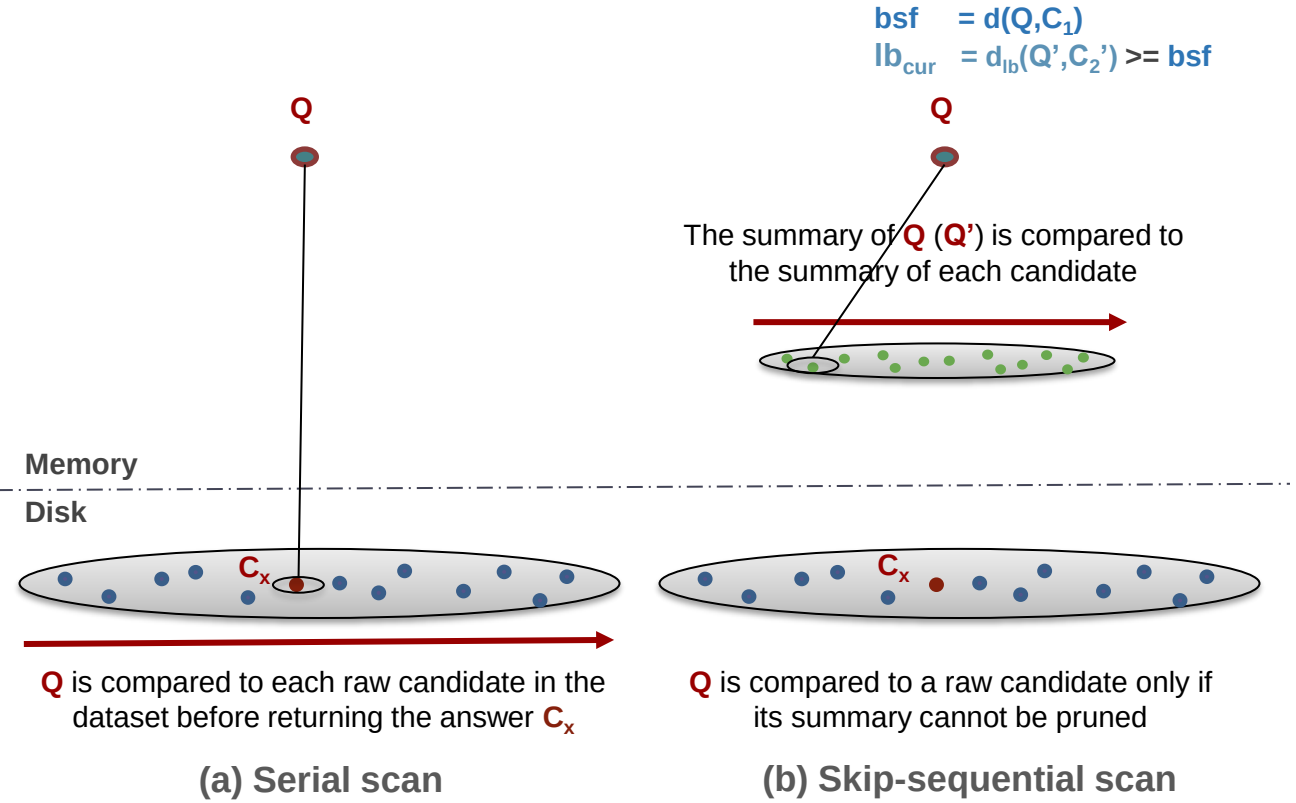
Answering a similarity search query using different access paths

Indexes vs. Scans



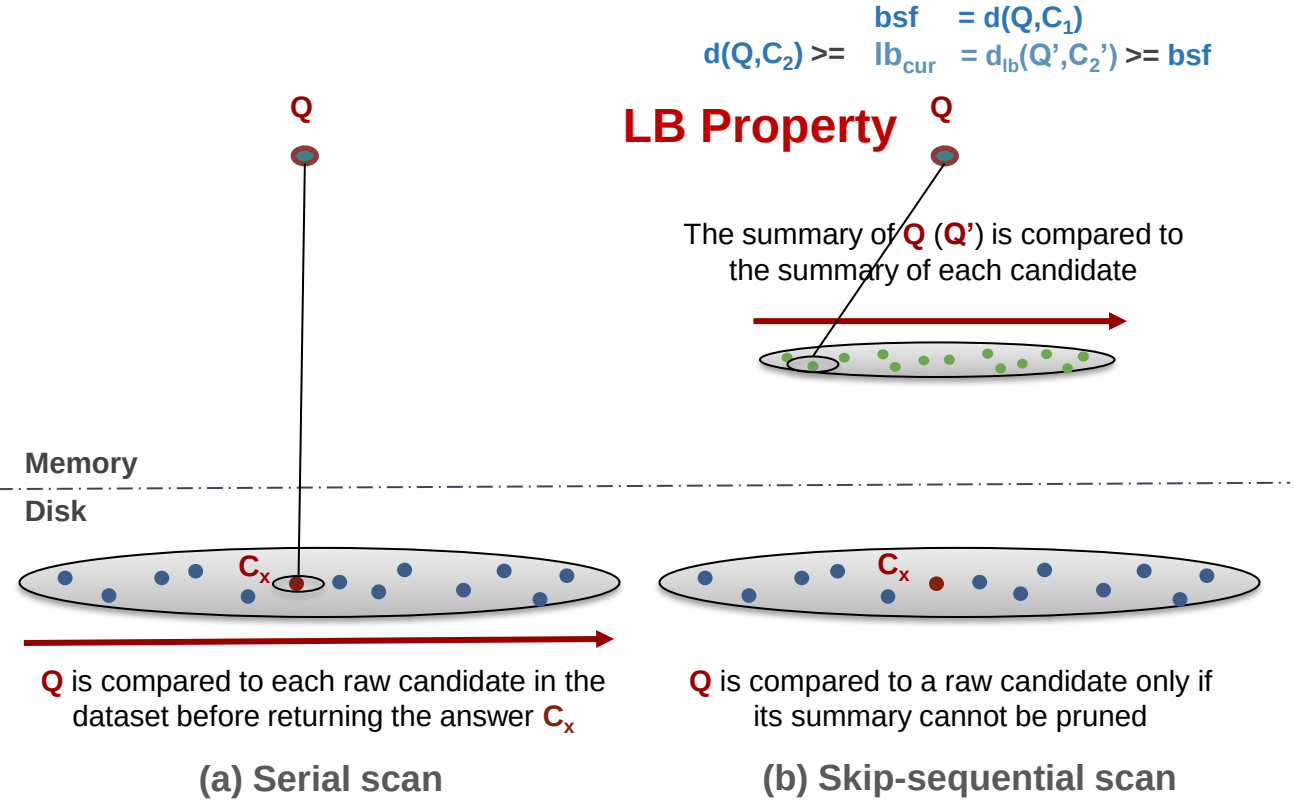
Answering a similarity search query using different access paths

Indexes vs. Scans



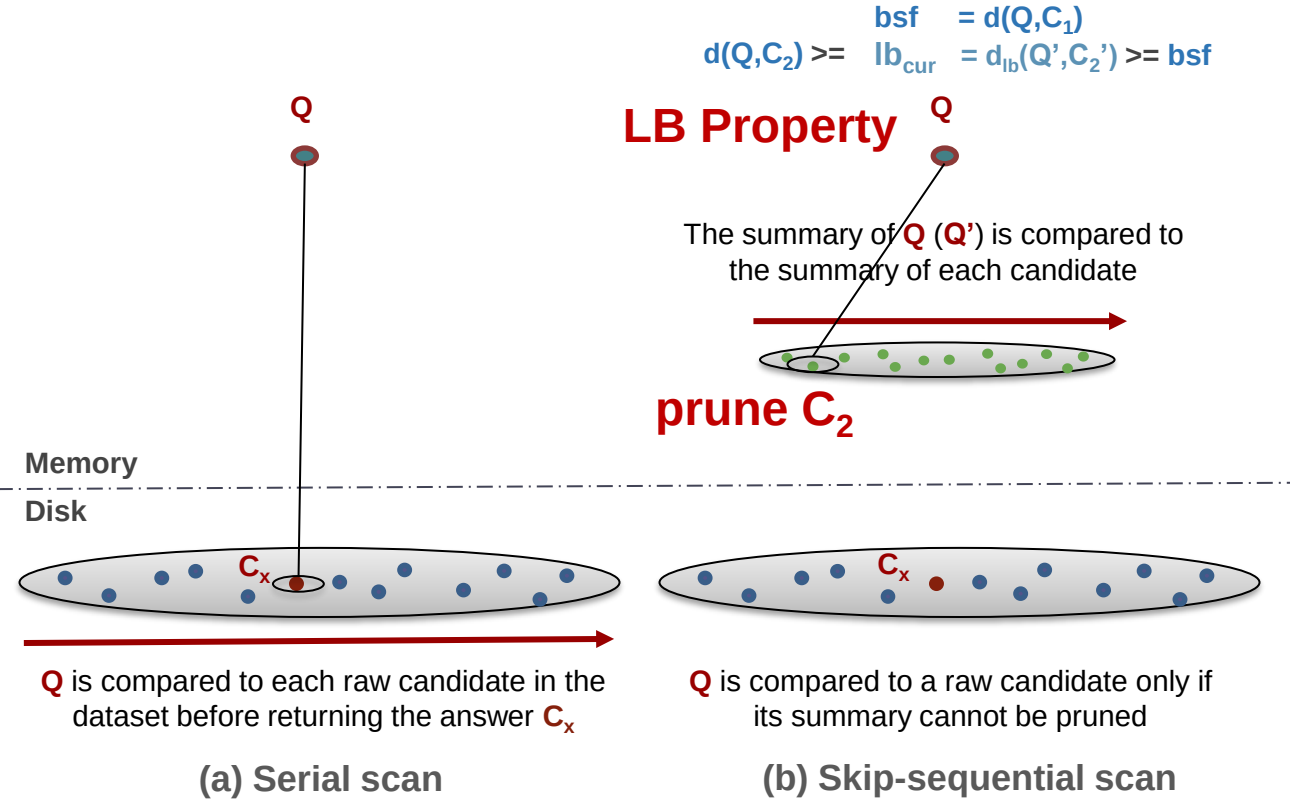
Answering a similarity search query using different access paths

Indexes vs. Scans



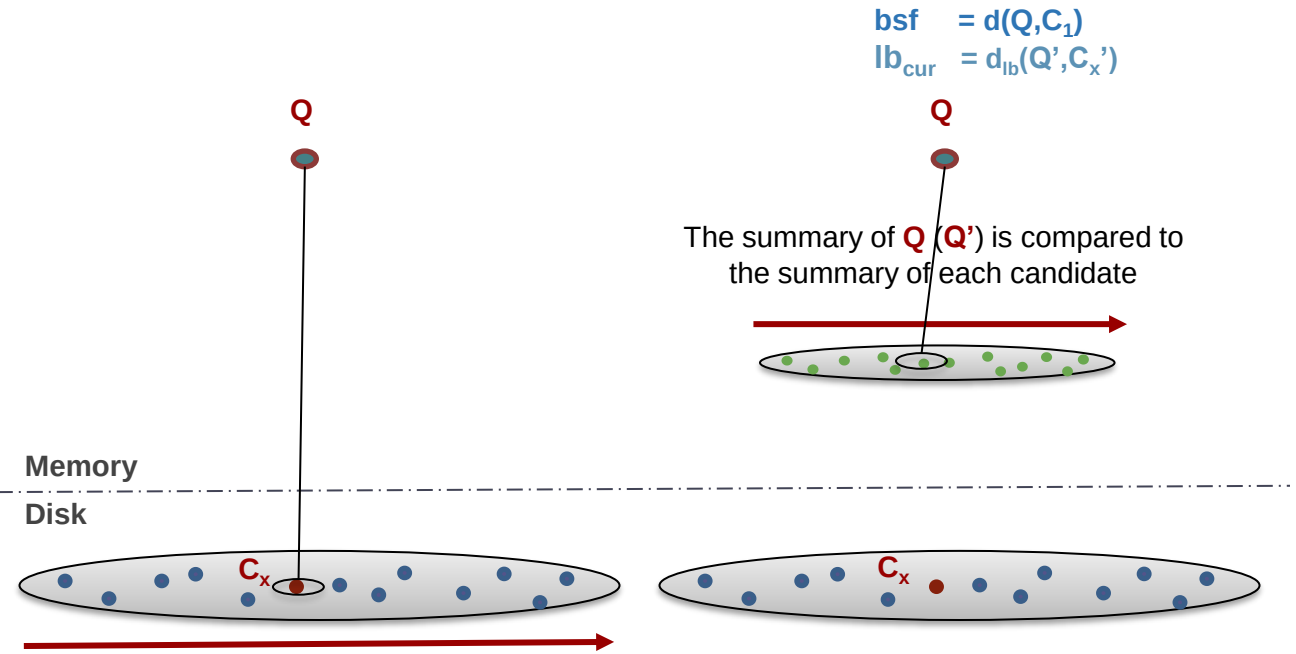
Answering a similarity search query using different access paths

Indexes vs. Scans



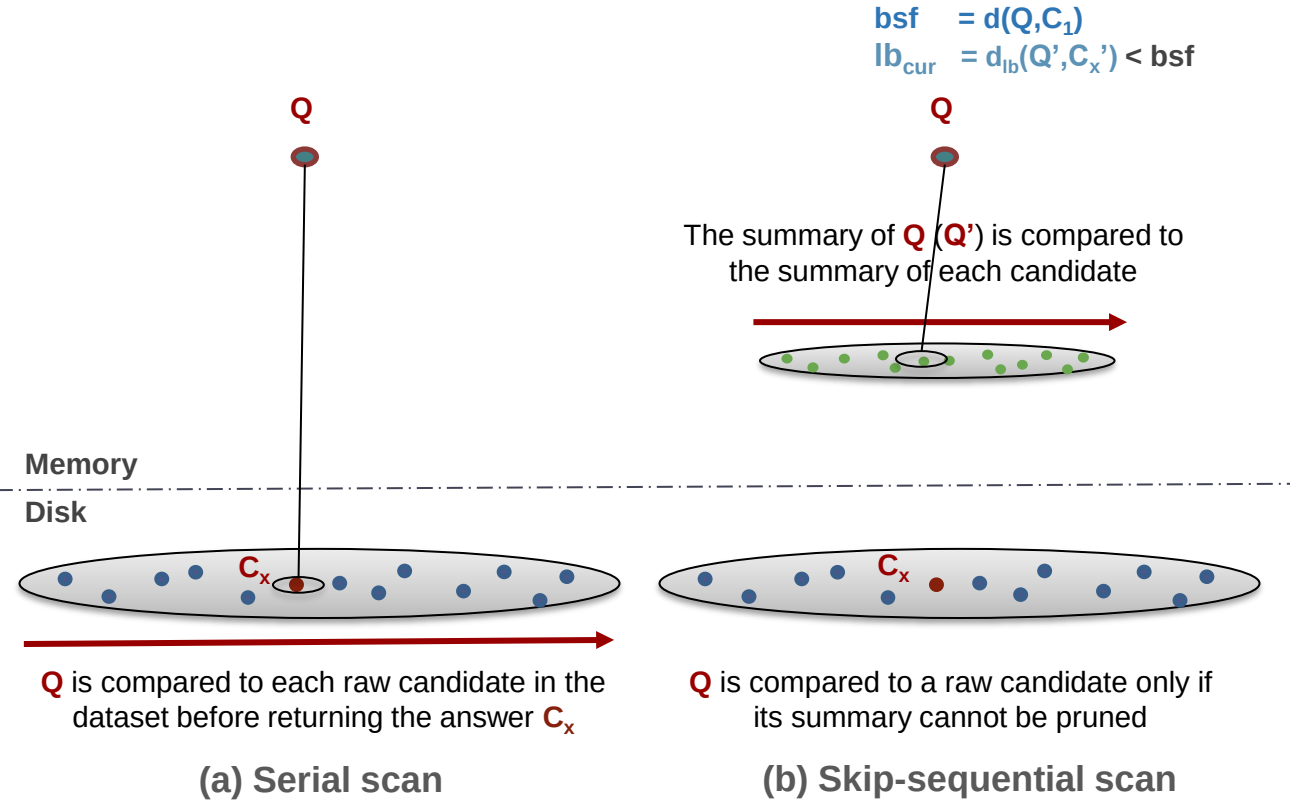
Answering a similarity search query using different access paths

Indexes vs. Scans



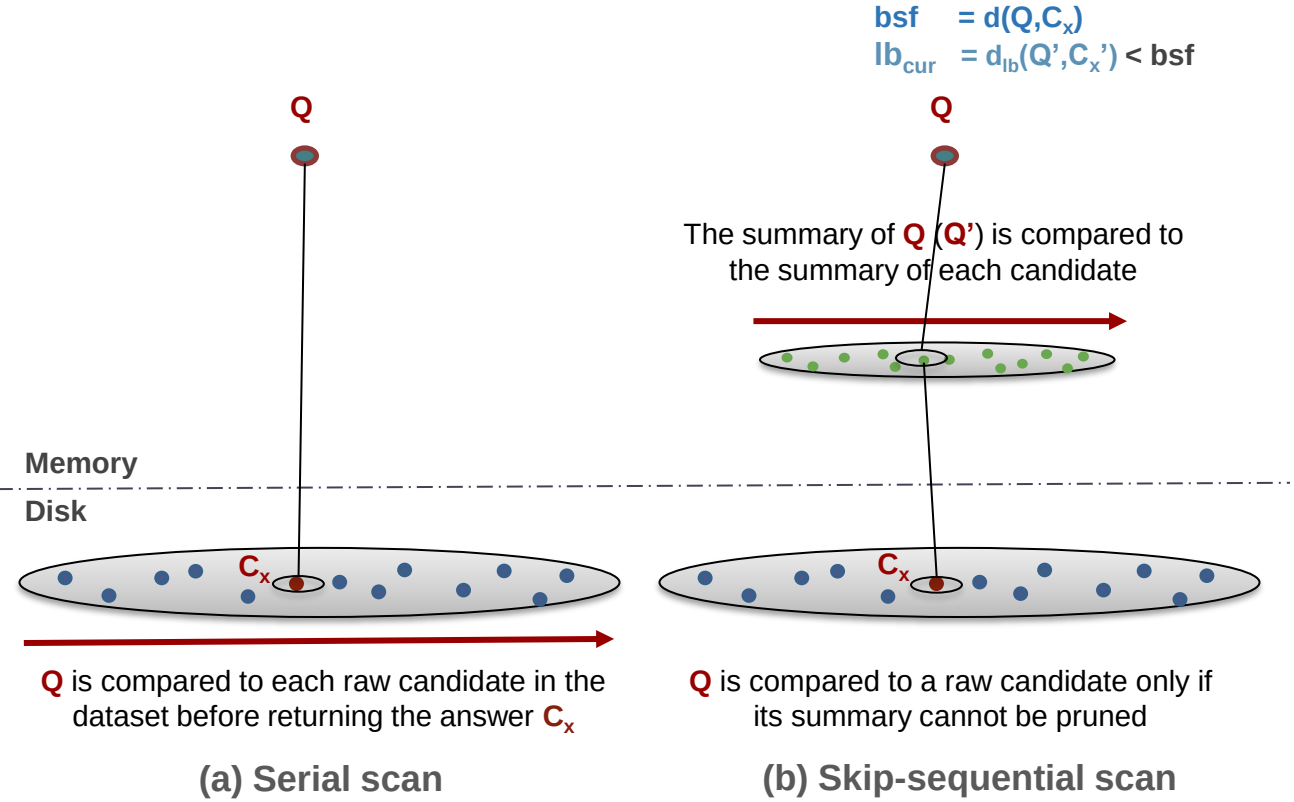
Answering a similarity search query using different access paths

Indexes vs. Scans



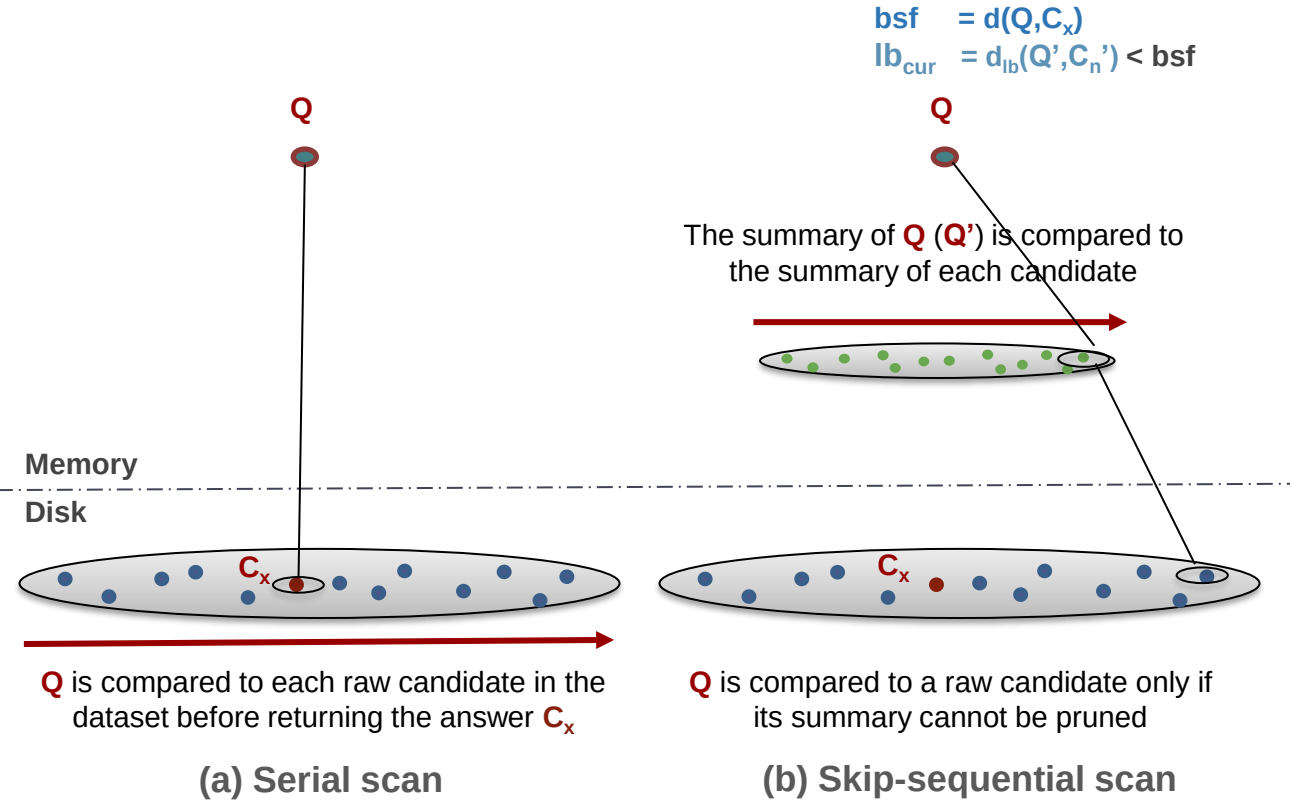
Answering a similarity search query using different access paths

Indexes vs. Scans



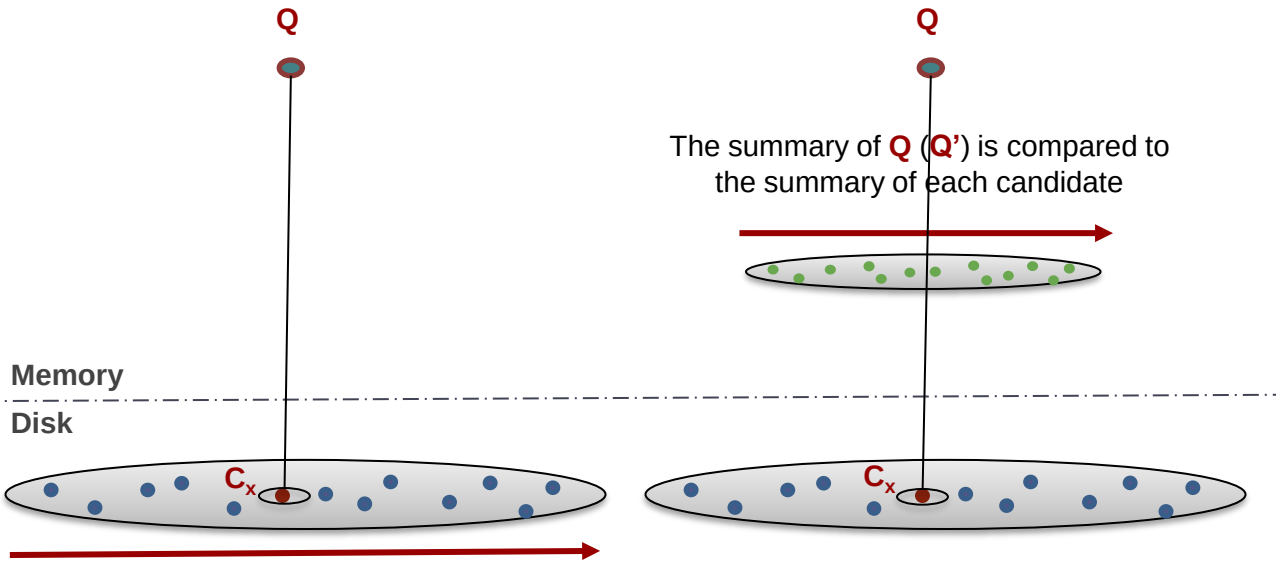
Answering a similarity search query using different access paths

Indexes vs. Scans



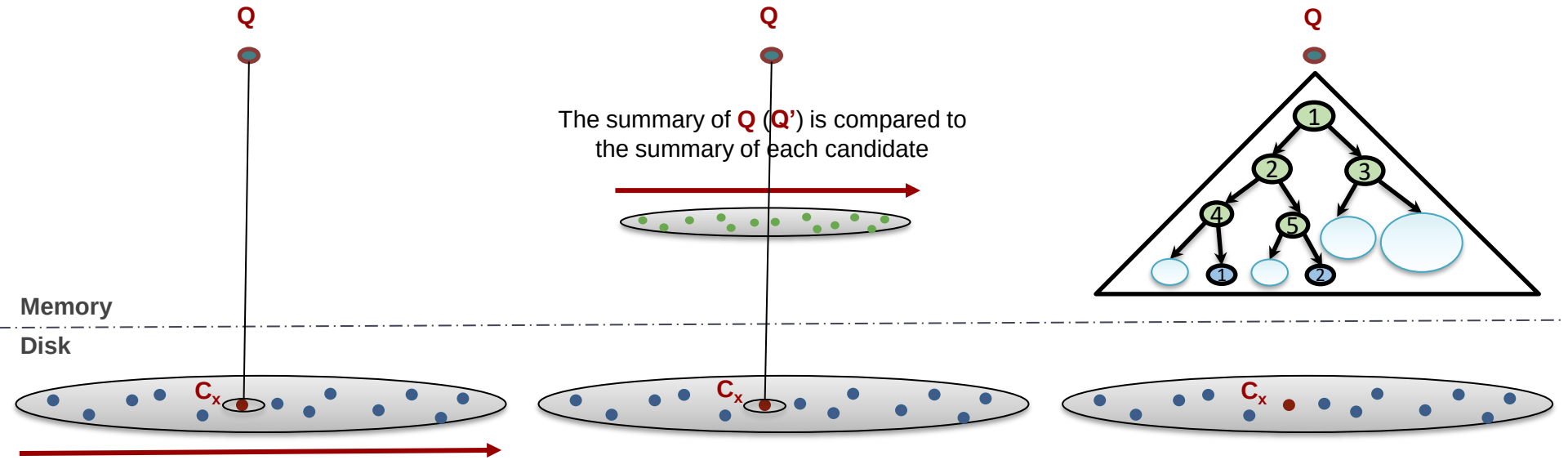
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

Indexes vs. Scans



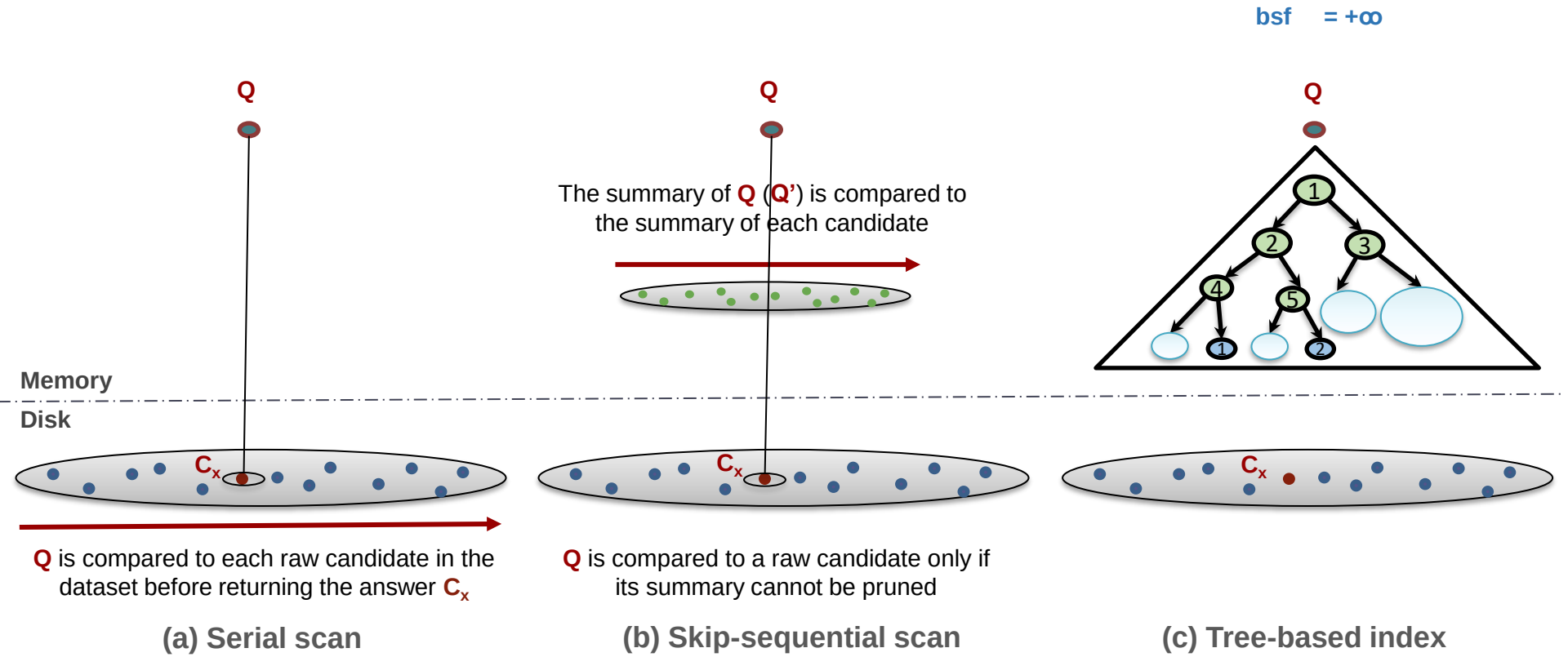
Q is compared to each raw candidate in the dataset before returning the answer C_x

Q is compared to a raw candidate only if its summary cannot be pruned

(a) Serial scan **(b) Skip-sequential scan** **(c) Tree-based index**

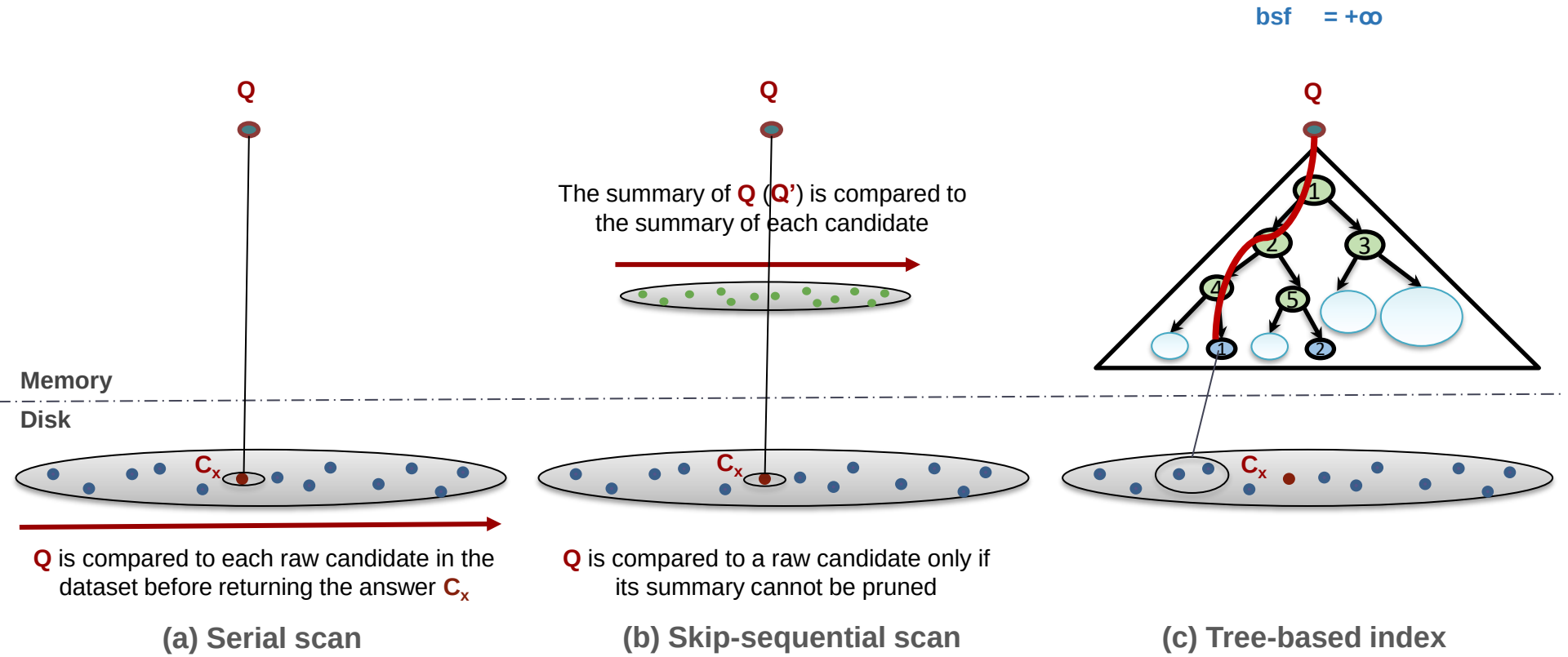
Answering a similarity search query using different access paths

Indexes vs. Scans



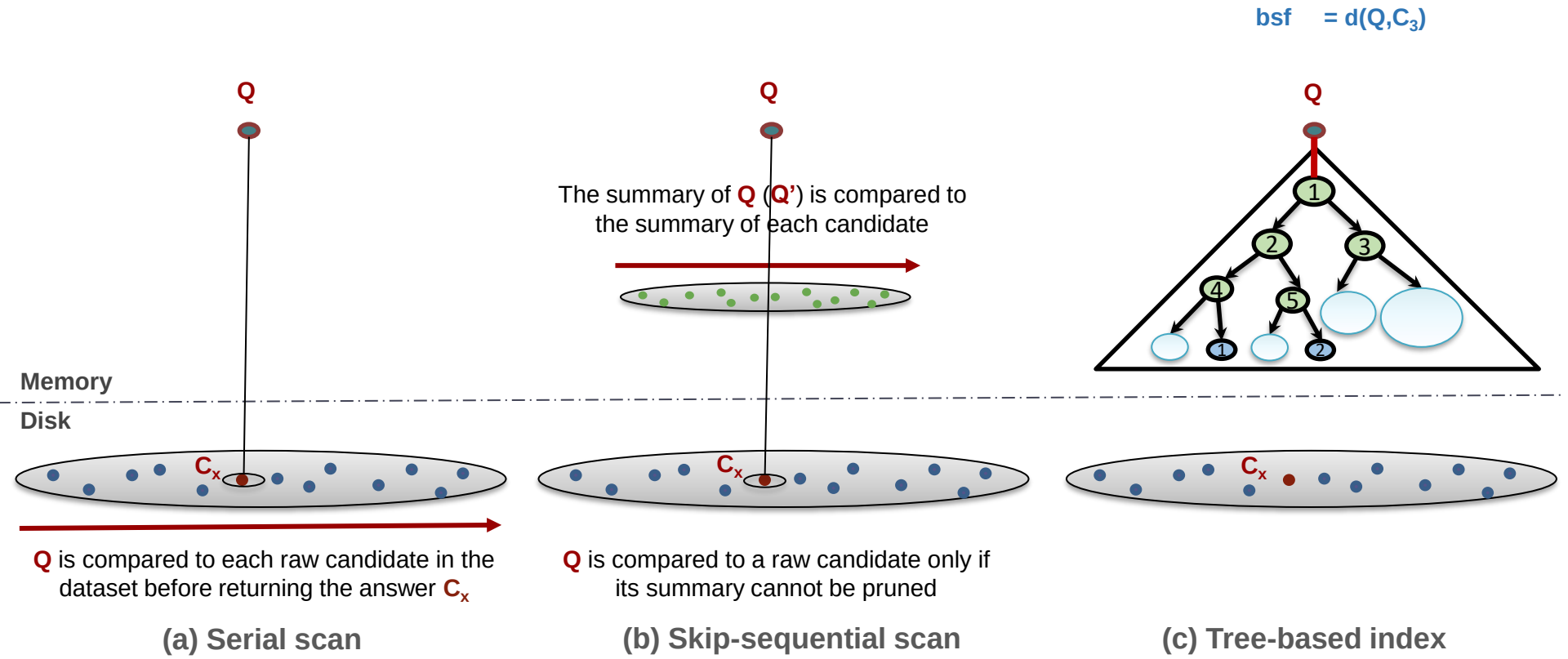
Answering a similarity search query using different access paths

Indexes vs. Scans



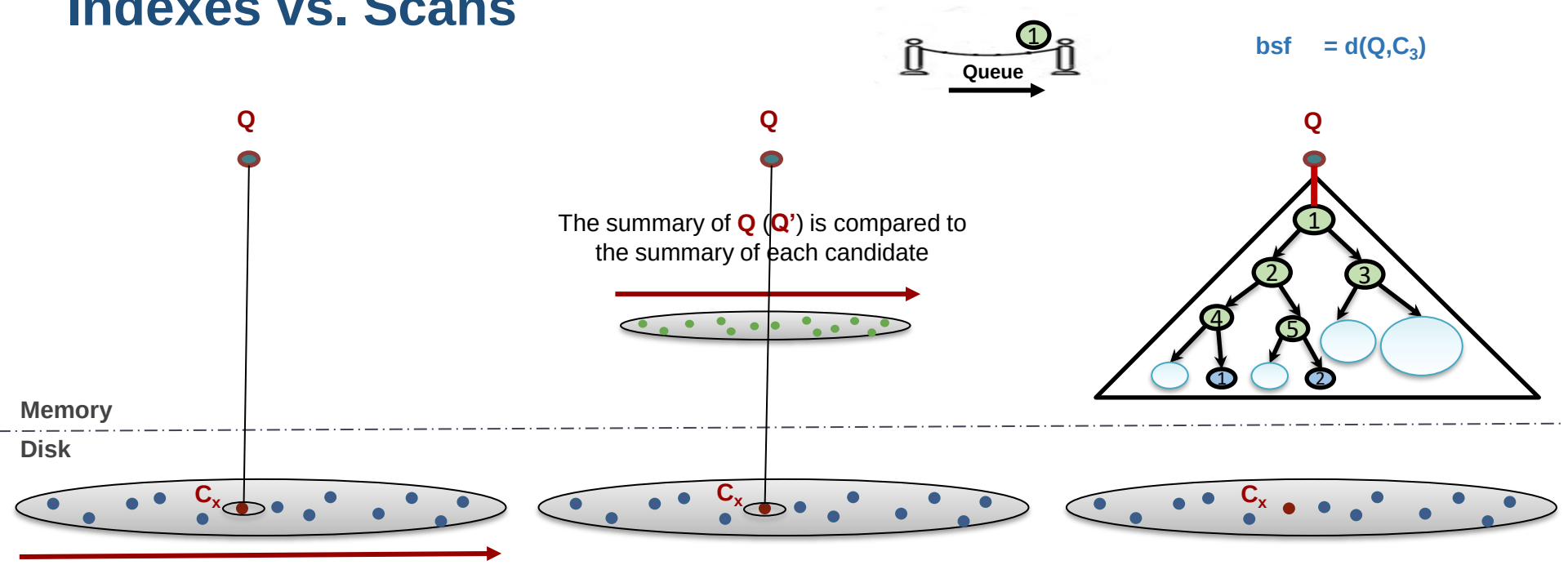
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

Indexes vs. Scans



Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

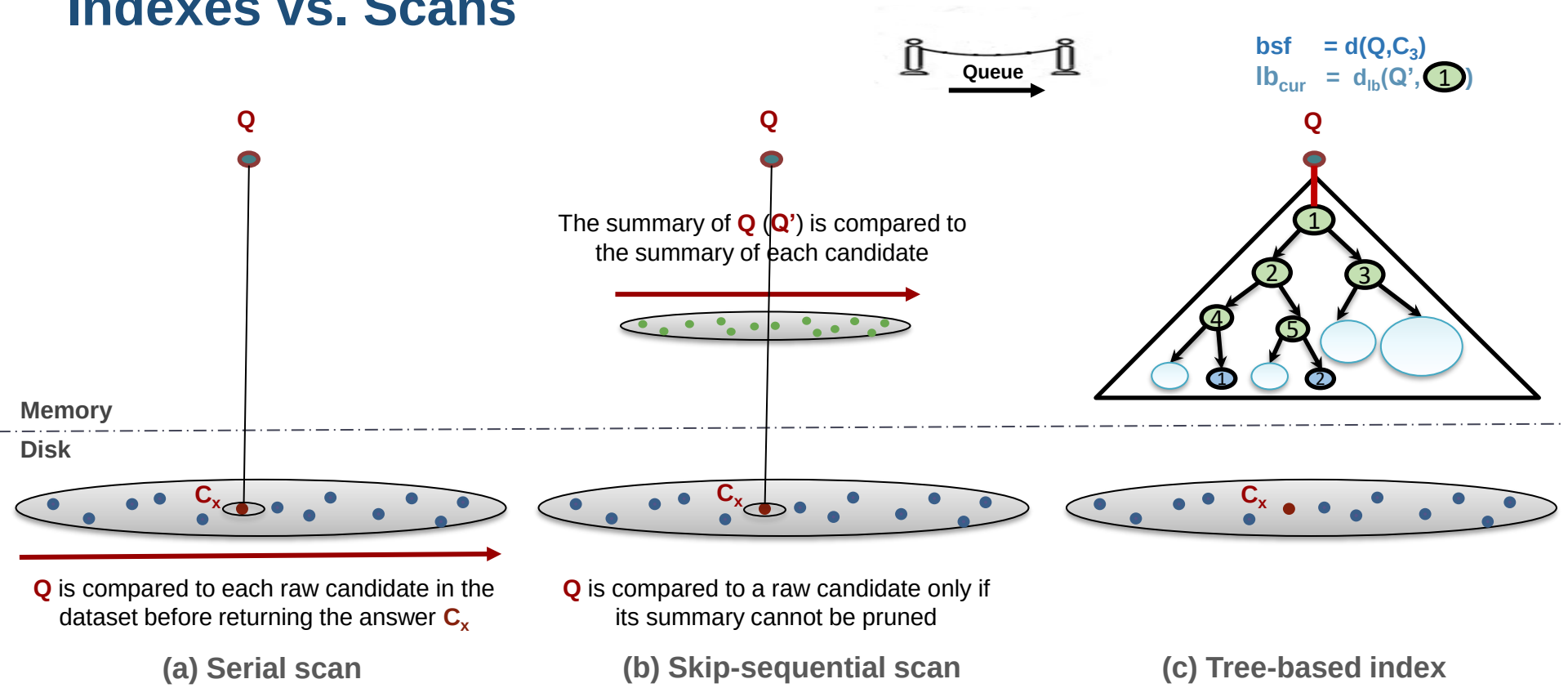
Q is compared to a raw candidate only if its summary cannot be pruned

(b) Skip-sequential scan

(c) Tree-based index

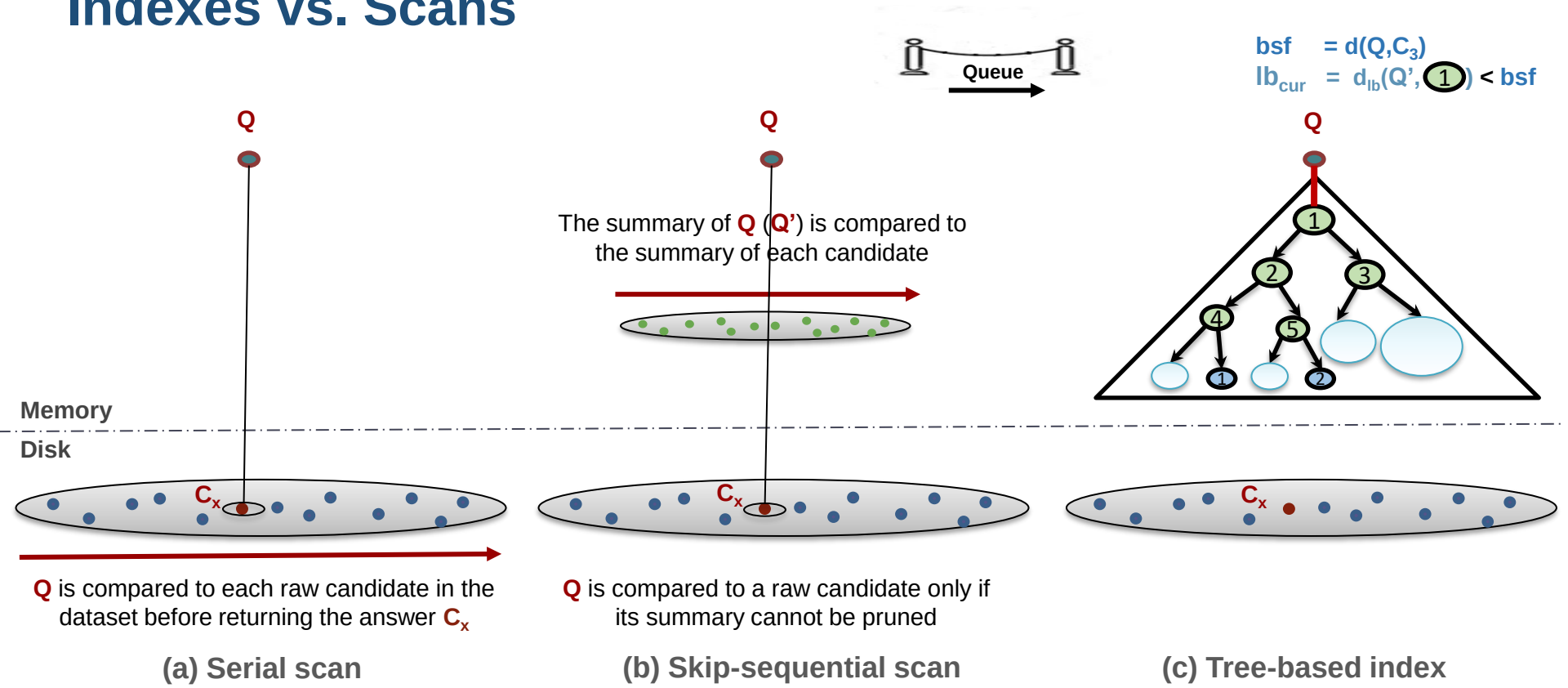
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

Indexes vs. Scans



Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

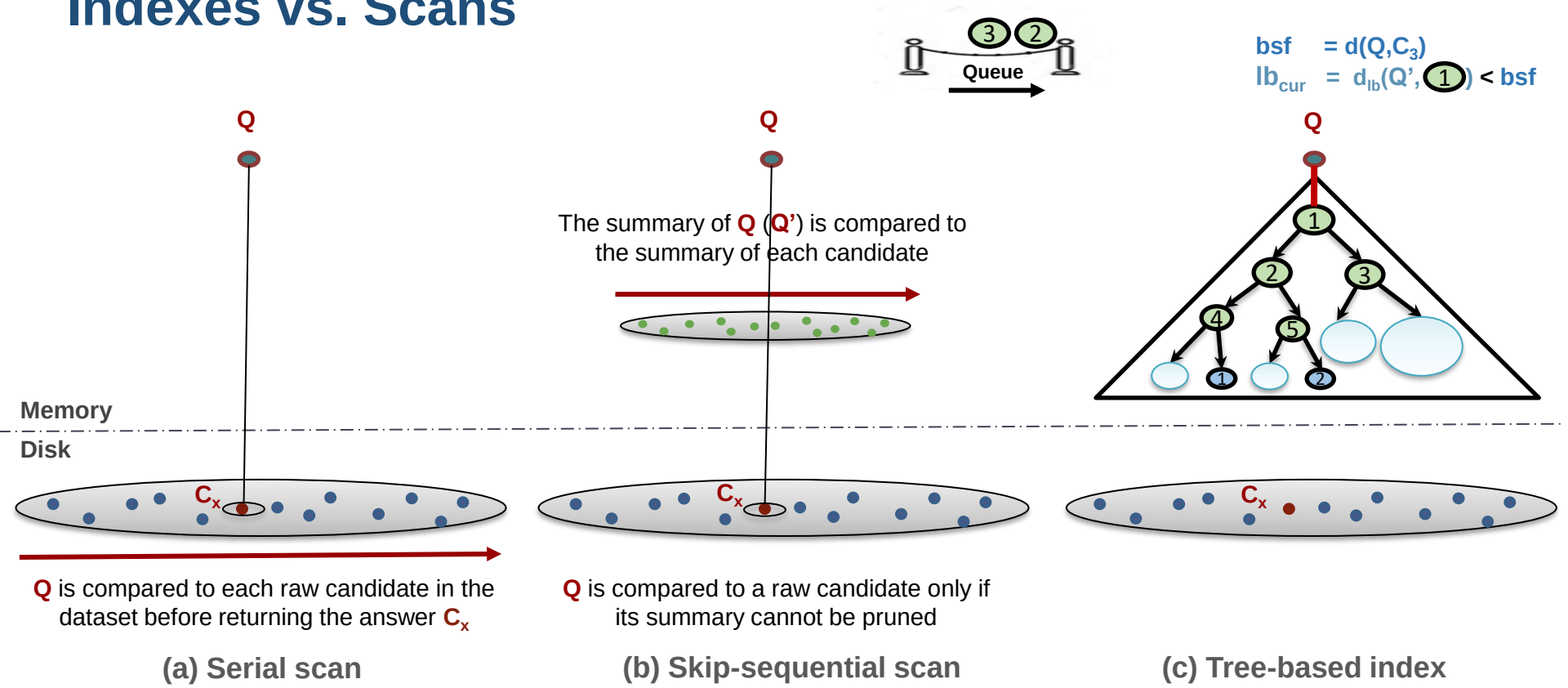
Q is compared to a raw candidate only if its summary cannot be pruned

(b) Skip-sequential scan

(c) Tree-based index

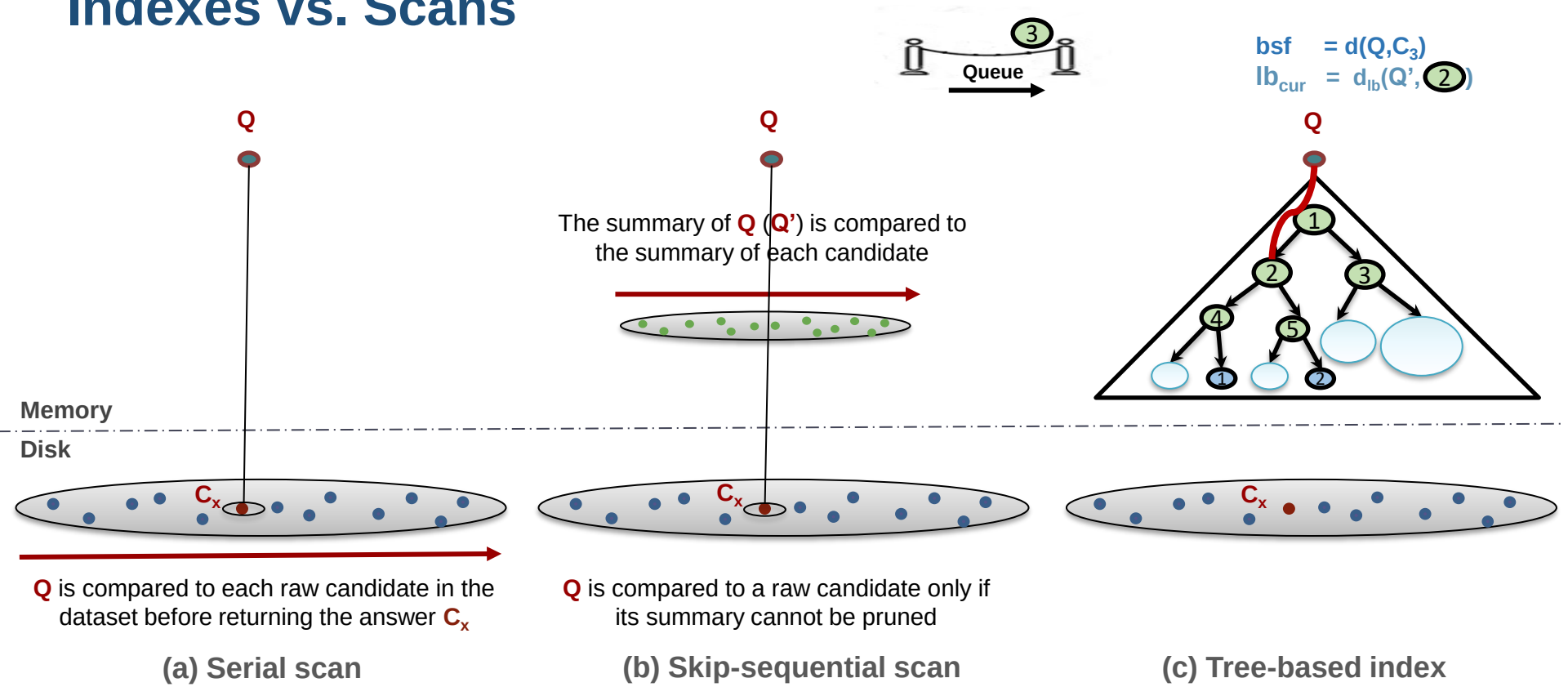
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

Indexes vs. Scans



Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

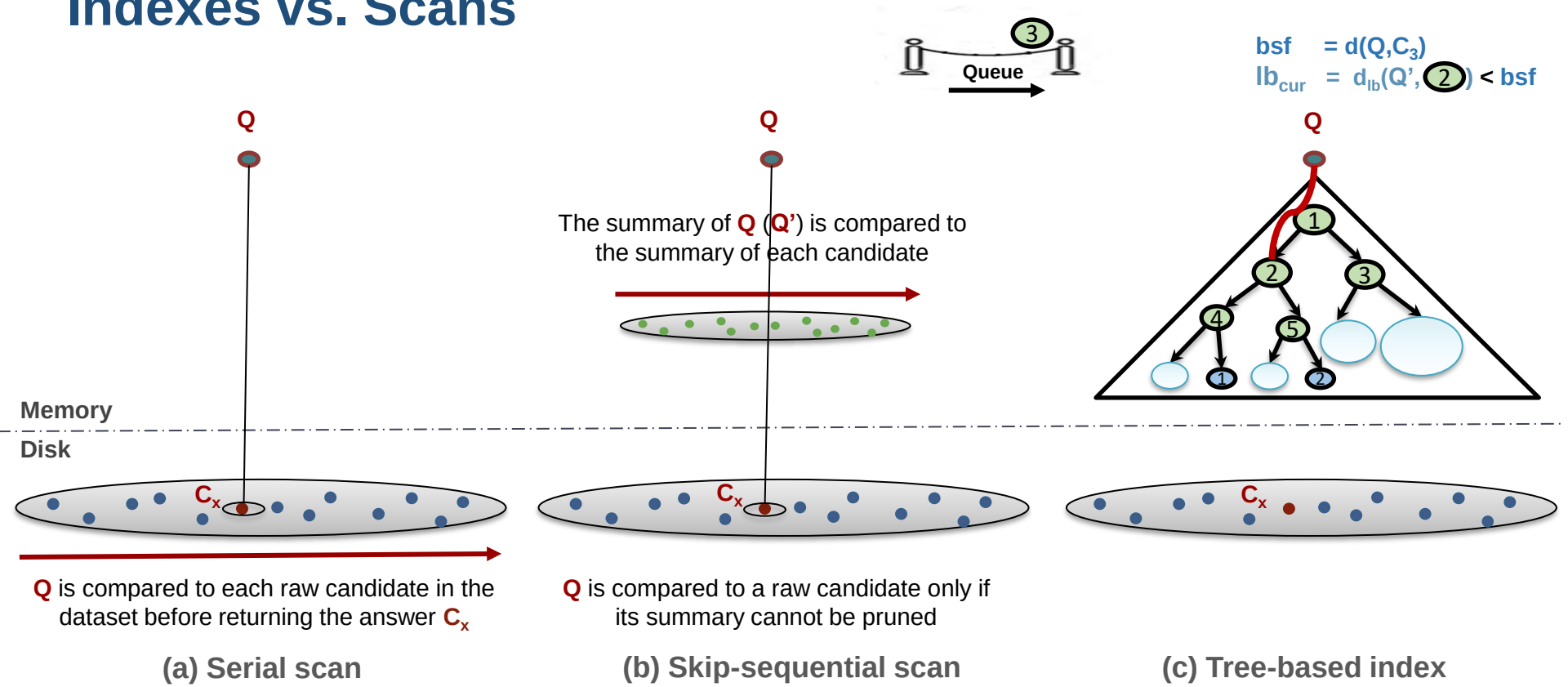
Q is compared to a raw candidate only if its summary cannot be pruned

(b) Skip-sequential scan

(c) Tree-based index

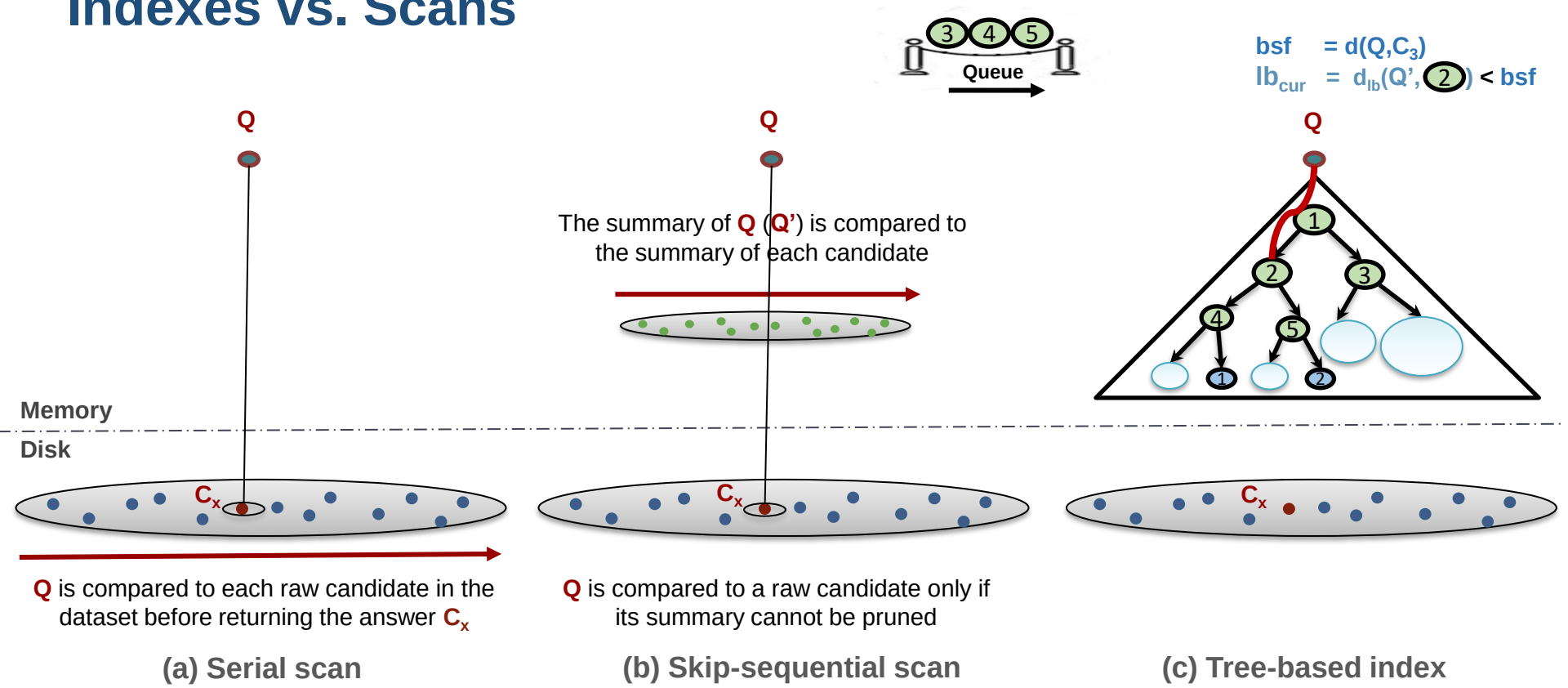
Answering a similarity search query using different access paths

Indexes vs. Scans



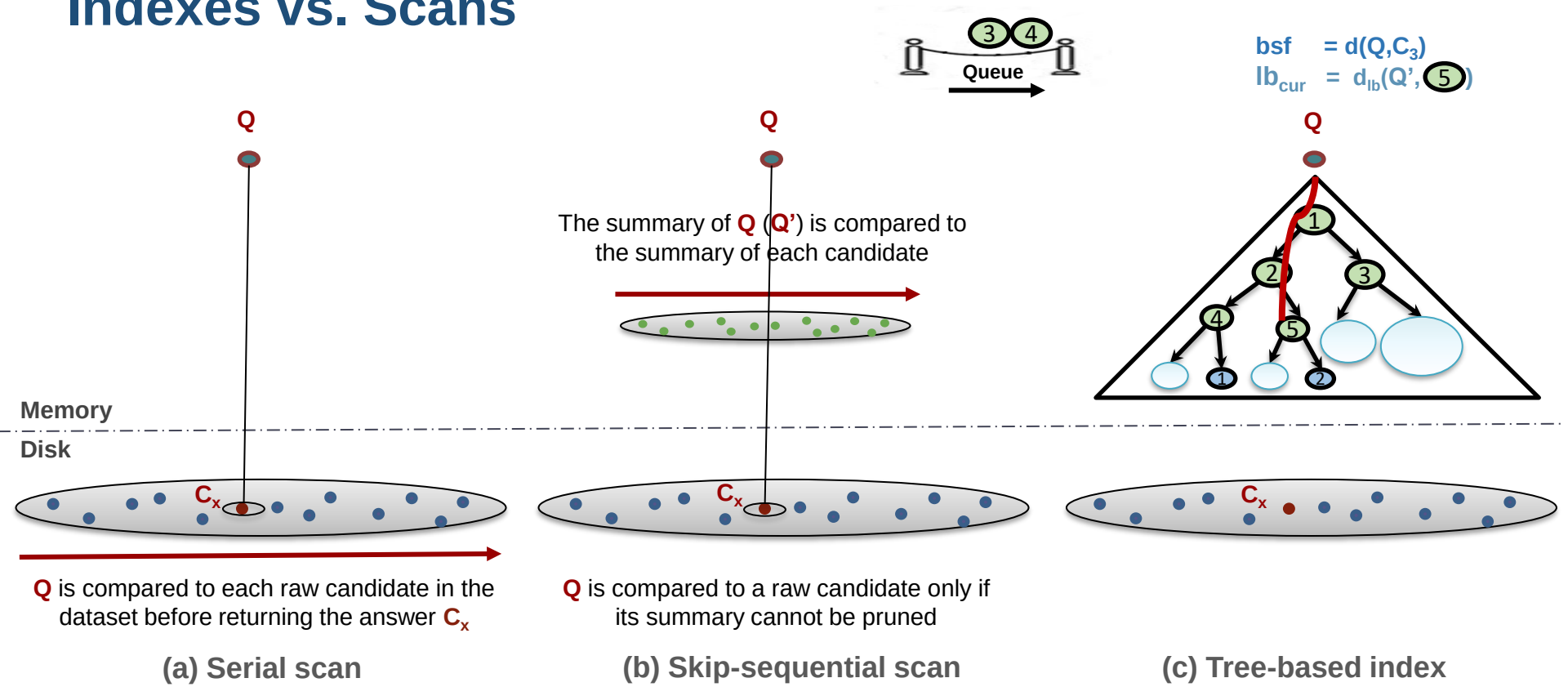
Answering a similarity search query using different access paths

Indexes vs. Scans



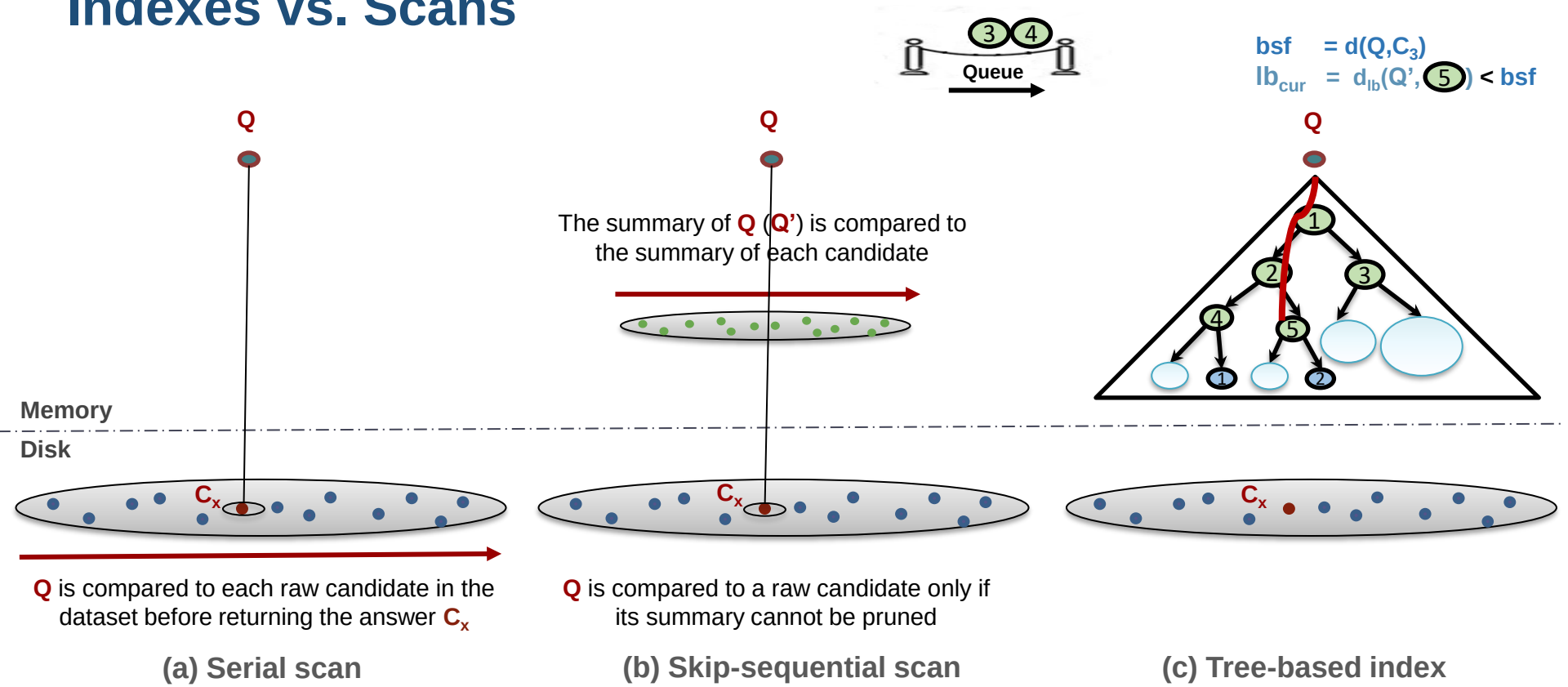
Answering a similarity search query using different access paths

Indexes vs. Scans



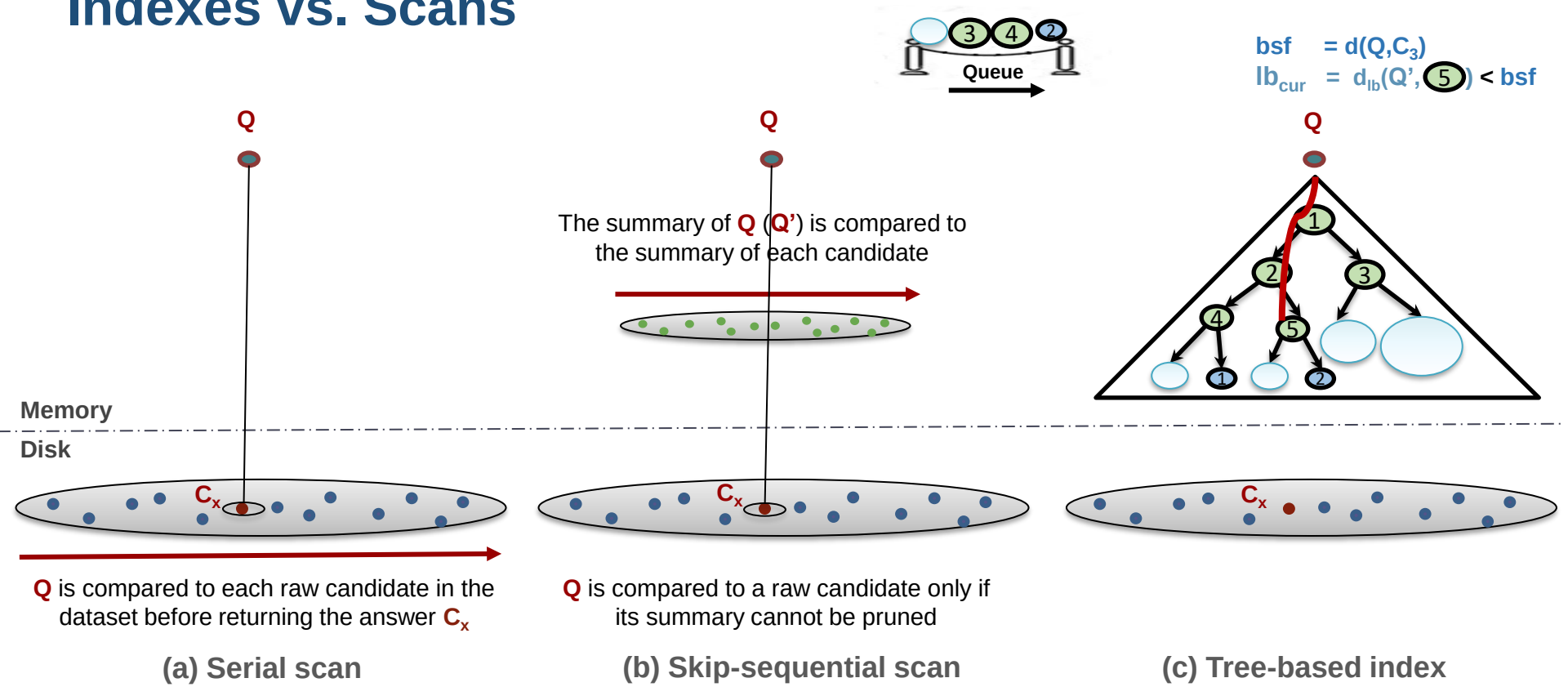
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

Indexes vs. Scans



Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

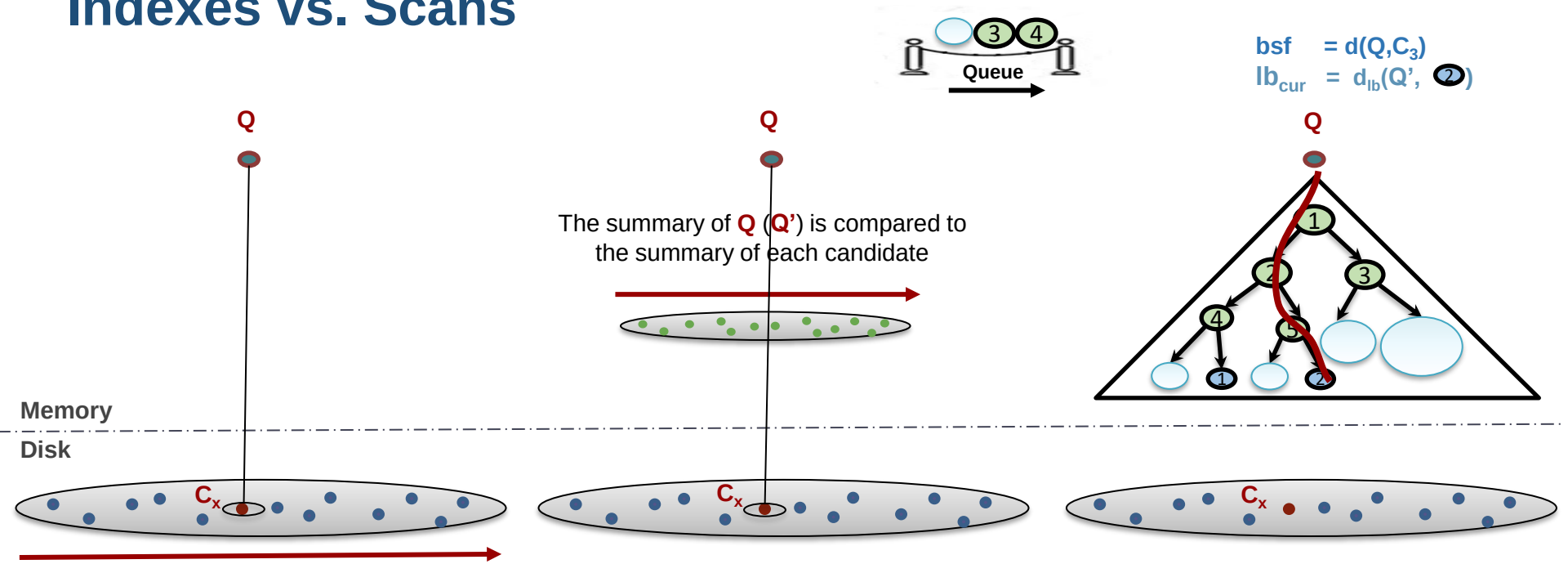
Q is compared to a raw candidate only if its summary cannot be pruned

(b) Skip-sequential scan

(c) Tree-based index

Answering a similarity search query using different access paths

Indexes vs. Scans



Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

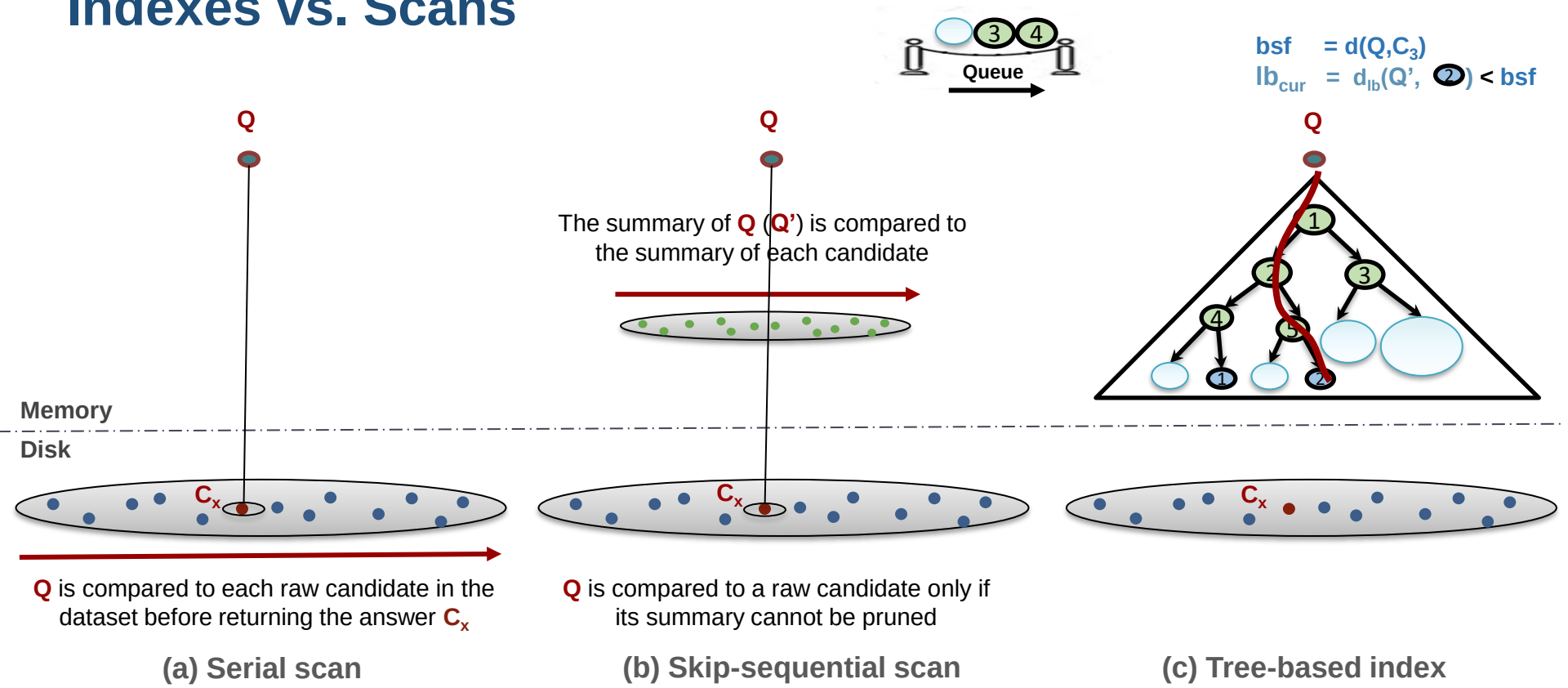
Q is compared to a raw candidate only if its summary cannot be pruned

(b) Skip-sequential scan

(c) Tree-based index

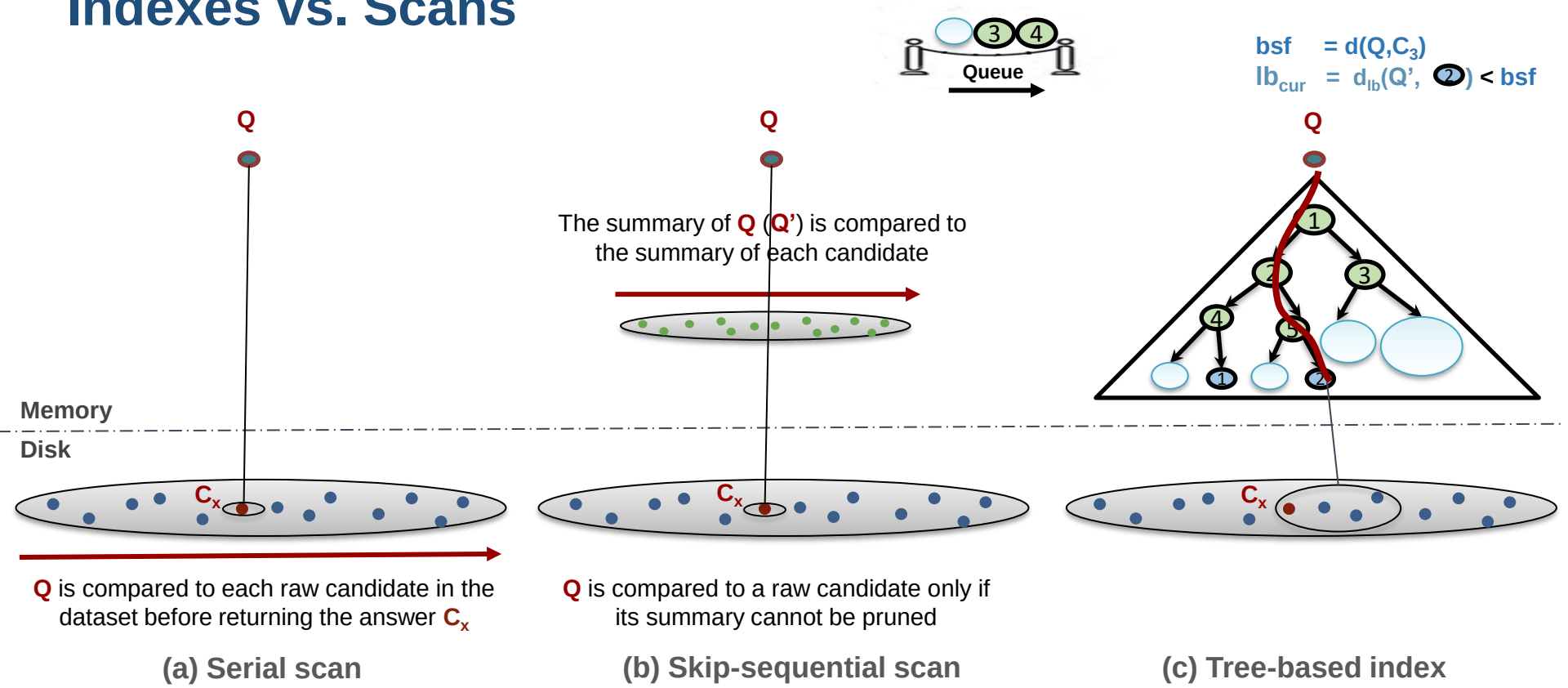
Answering a similarity search query using different access paths

Indexes vs. Scans



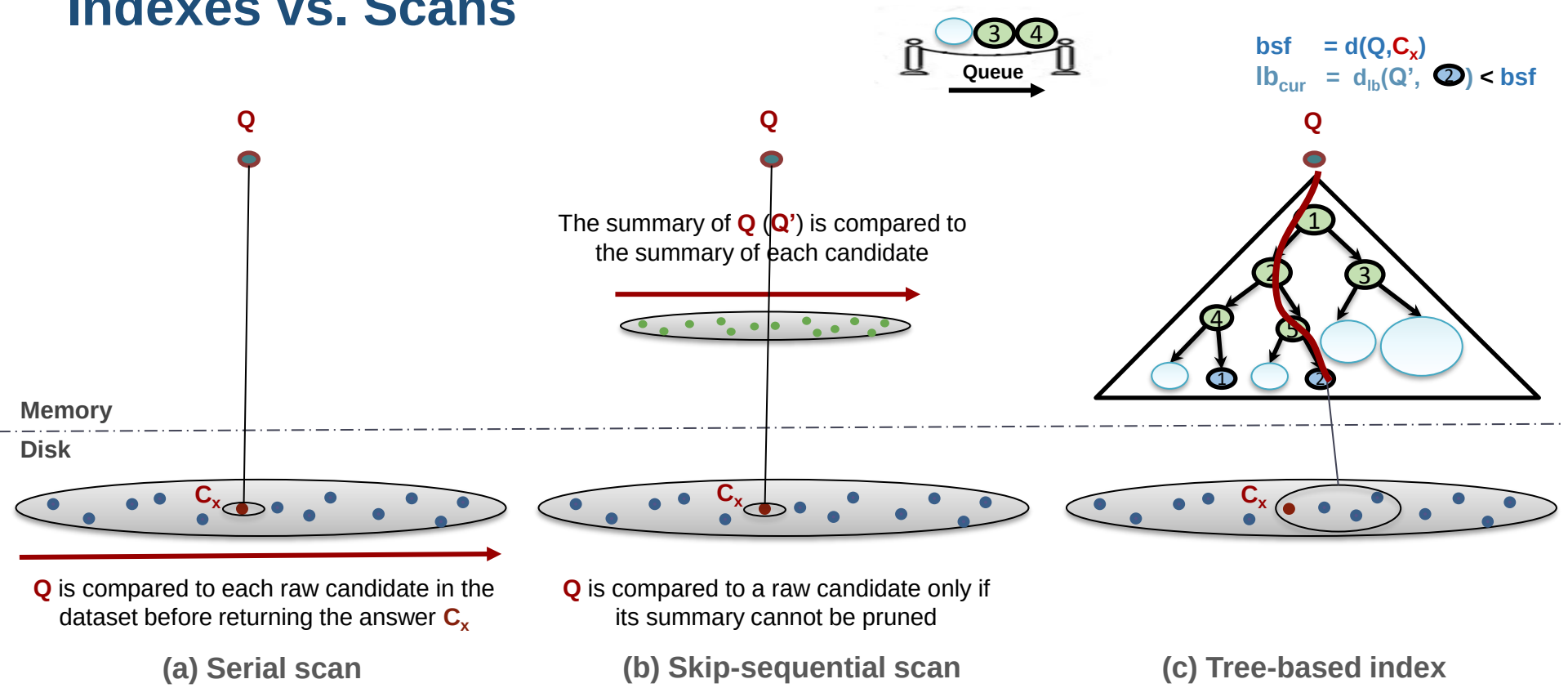
Answering a similarity search query using different access paths

Indexes vs. Scans



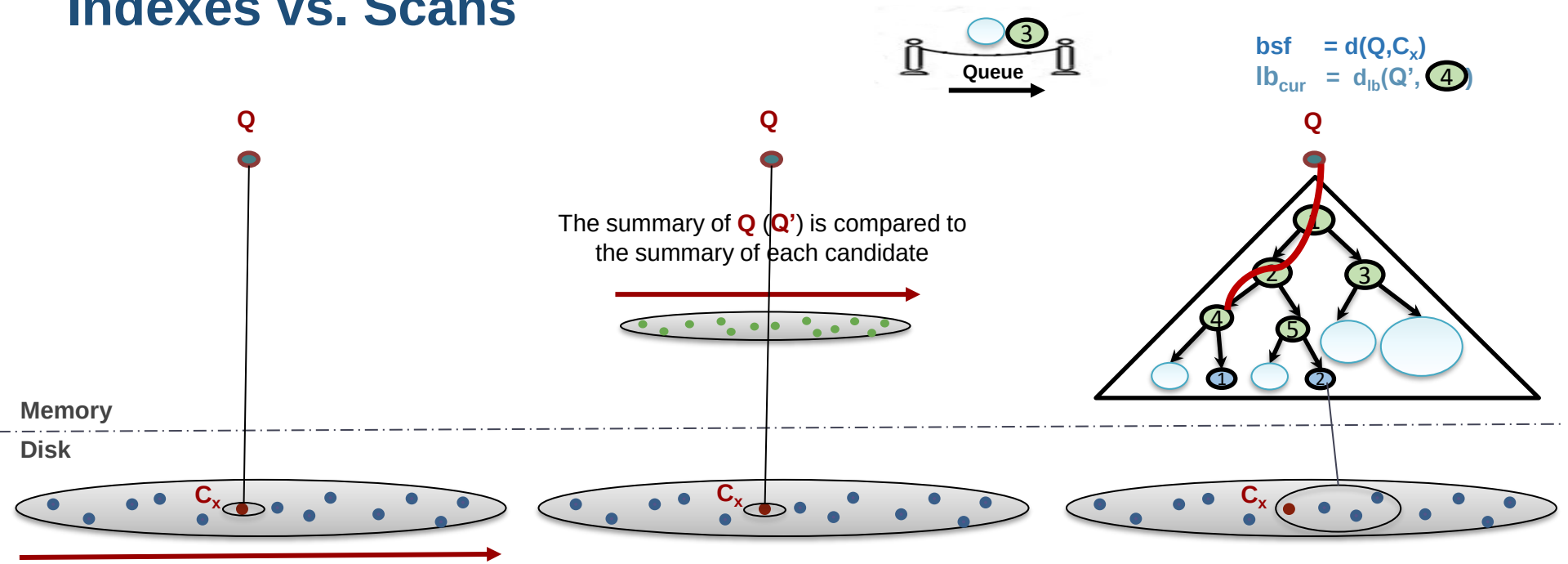
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

Indexes vs. Scans



Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

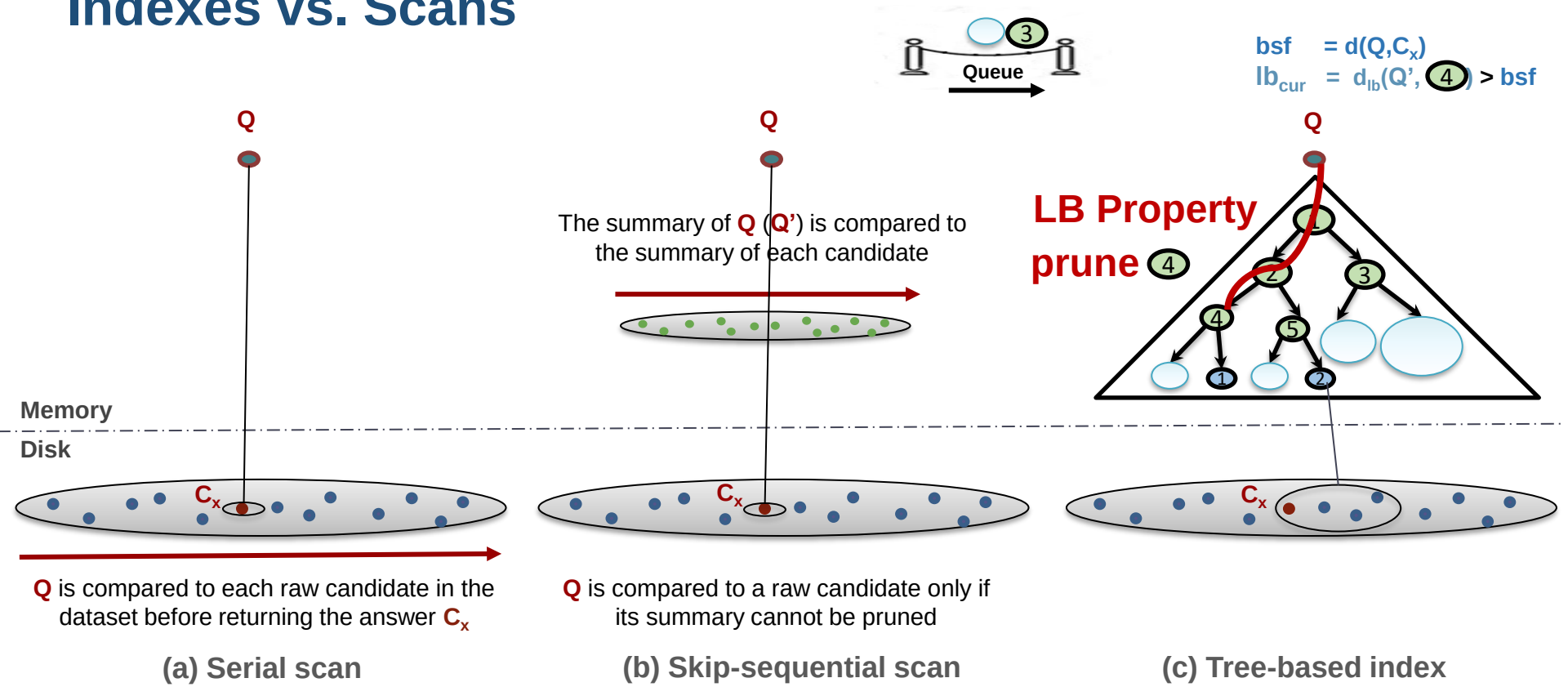
Q is compared to a raw candidate only if its summary cannot be pruned

(b) Skip-sequential scan

(c) Tree-based index

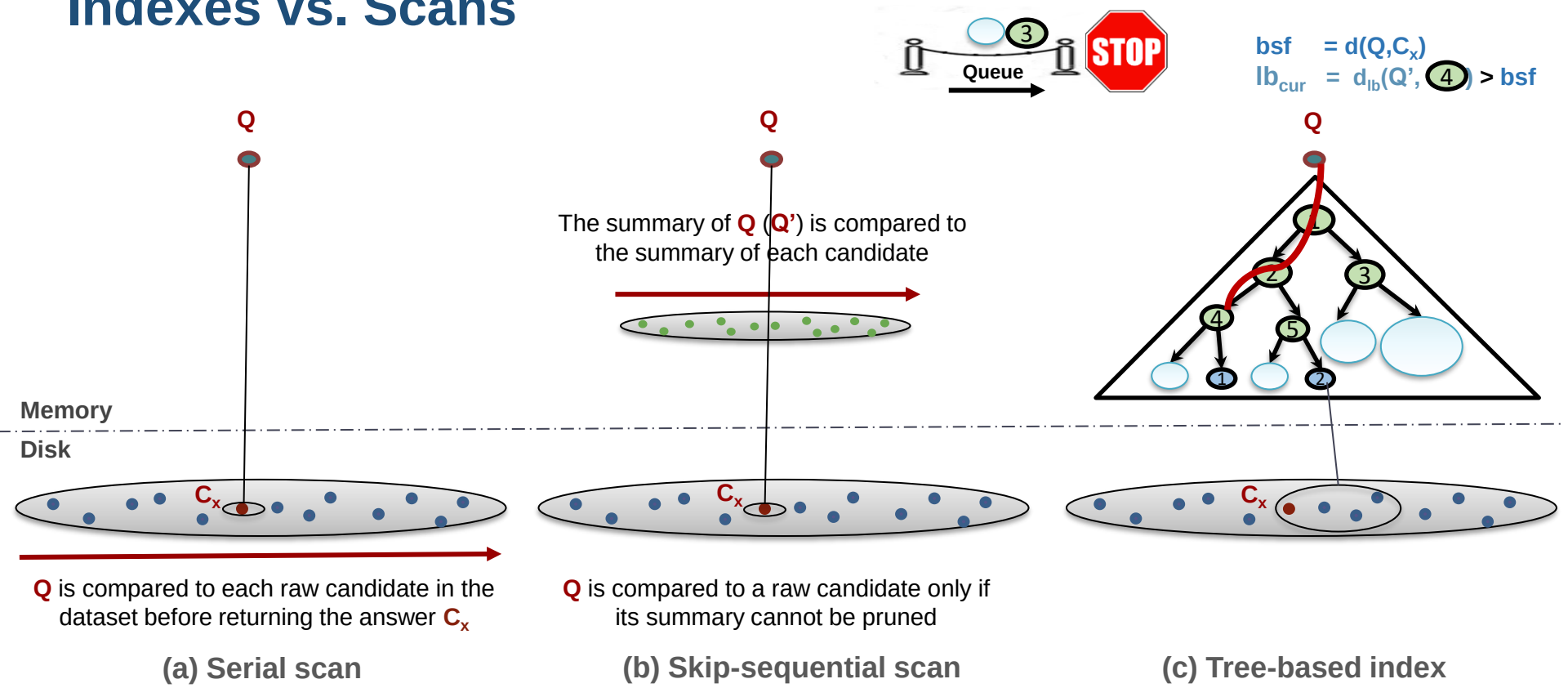
Answering a similarity search query using different access paths

Indexes vs. Scans



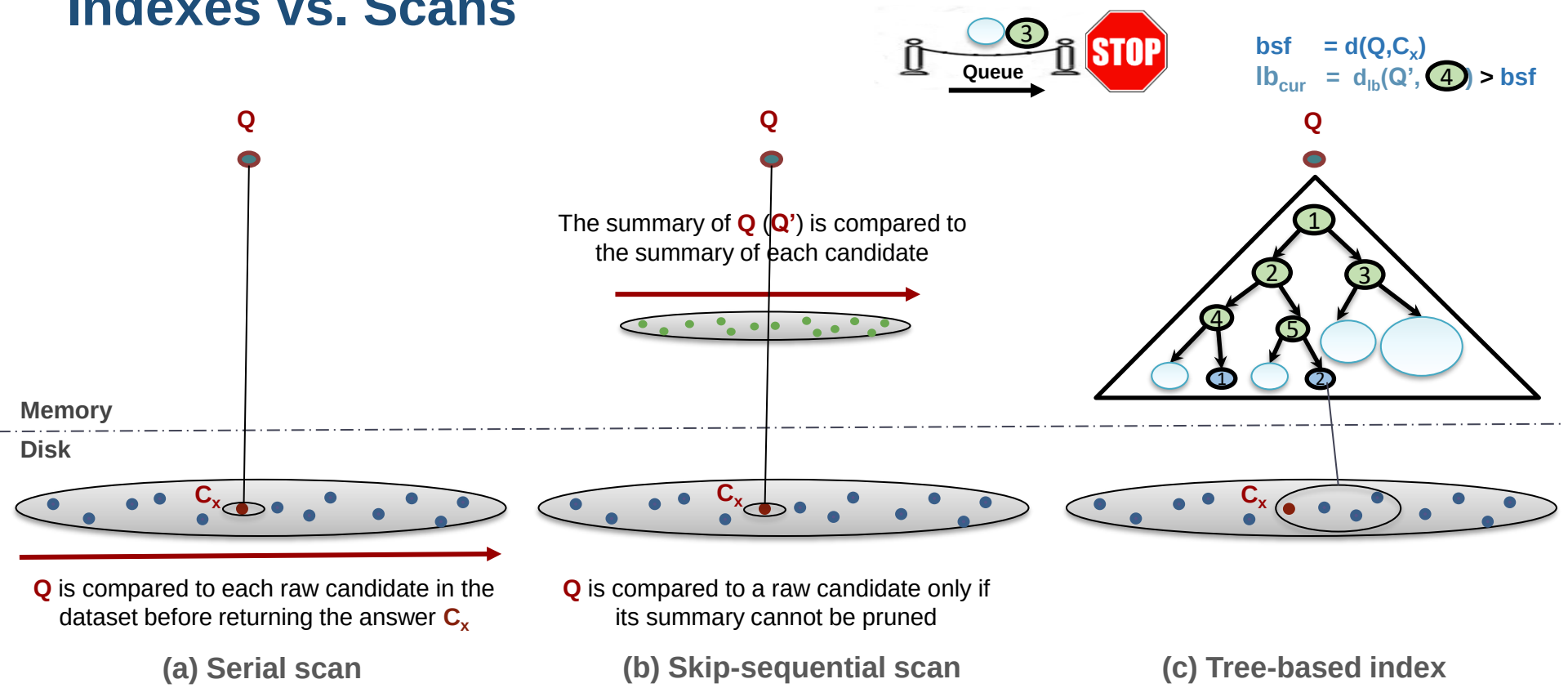
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

Indexes vs. Scans



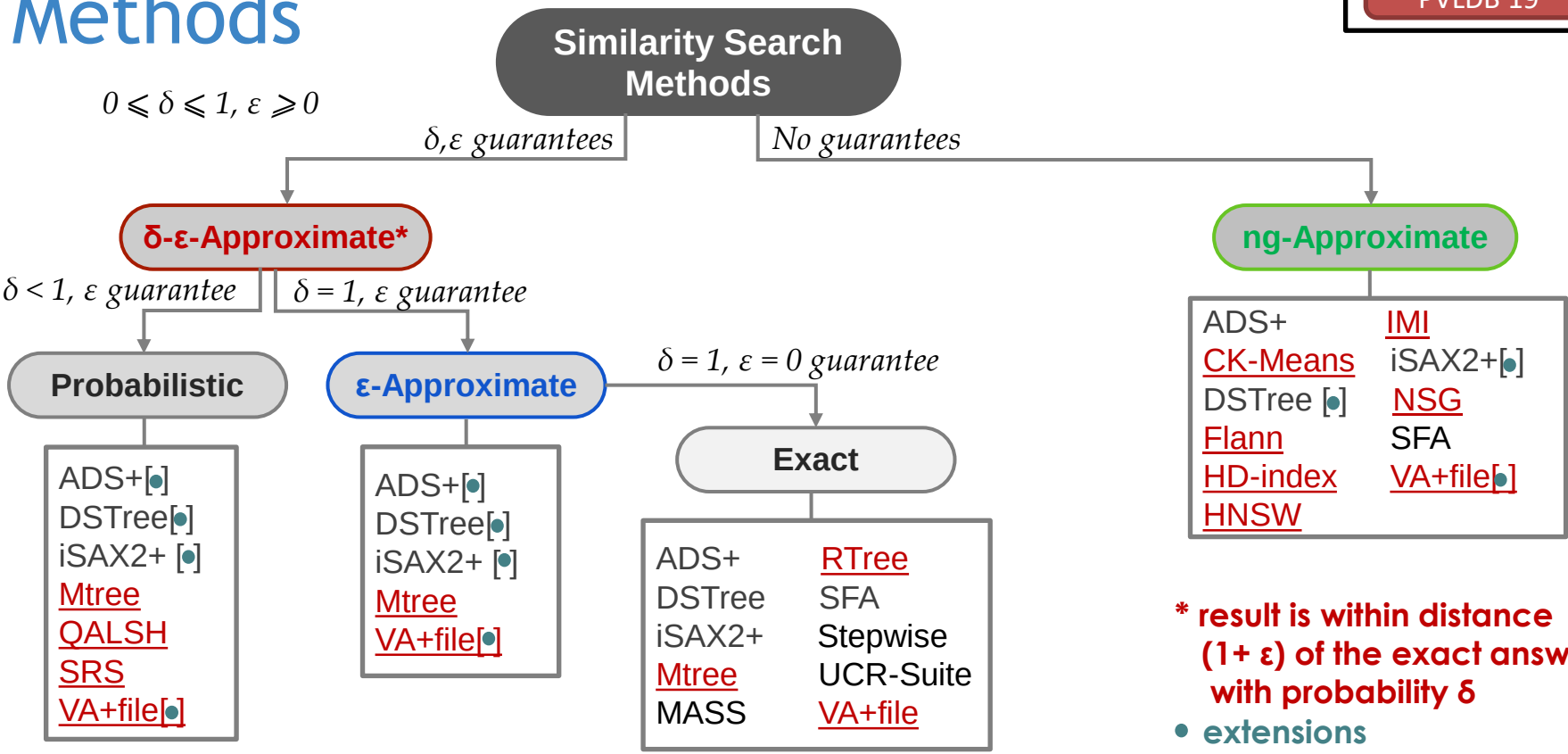
Answering a similarity search query using different access paths

Publications

- Echihabi-PVLDB'18
- Echihabi-PVLDB'19

Techniques for data Series
Techniques for High-D vectors

Methods



*** result is within distance (1+ ε) of the exact answer with probability δ**

- **extensions**

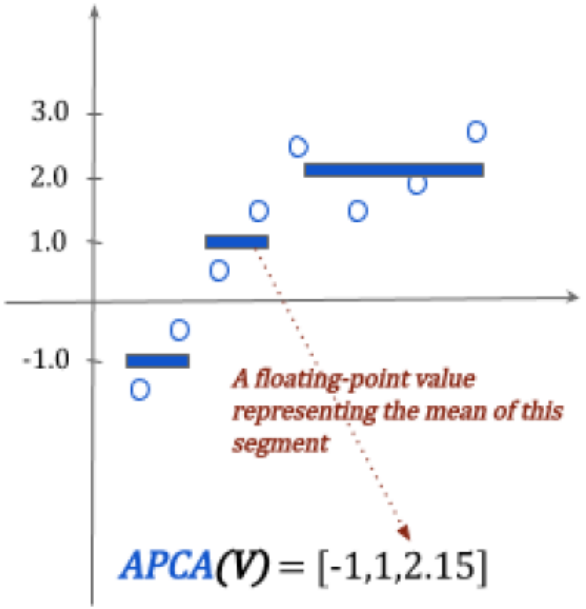
Data Series Indexing

Publications

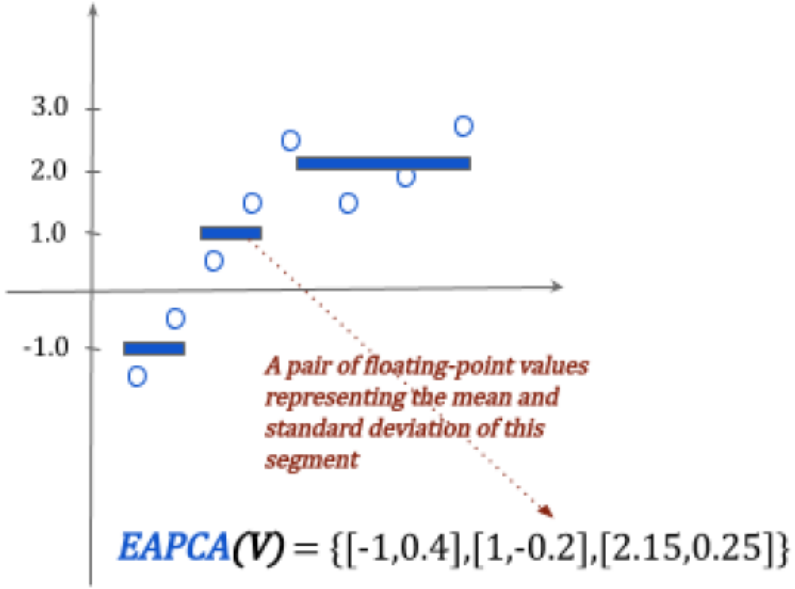
Wang-PVLDB'13

DSTree
Summarization

$\mathbf{V} = [-1.5,-0.5, 0.5, 1.5, 2.5, 1.5,2, 2.6]$



(a) APCA



(b) EAPCA

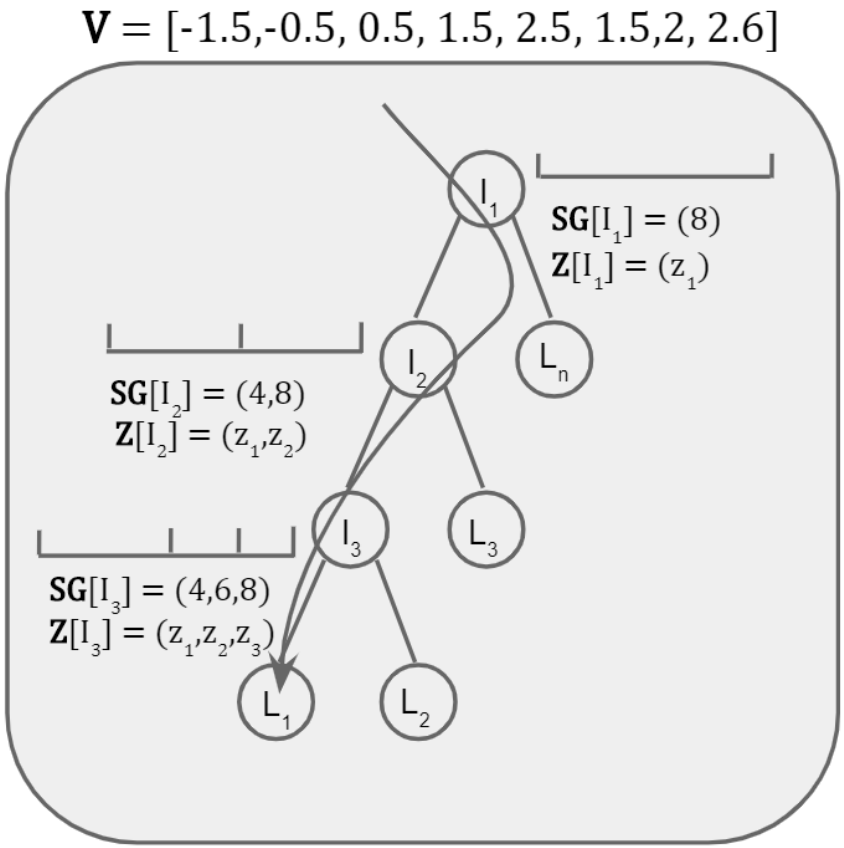
Intertwined with indexing

The APCA and EAPCA representations

Publications

Wang-PVLDB'13

DSTree
Indexing



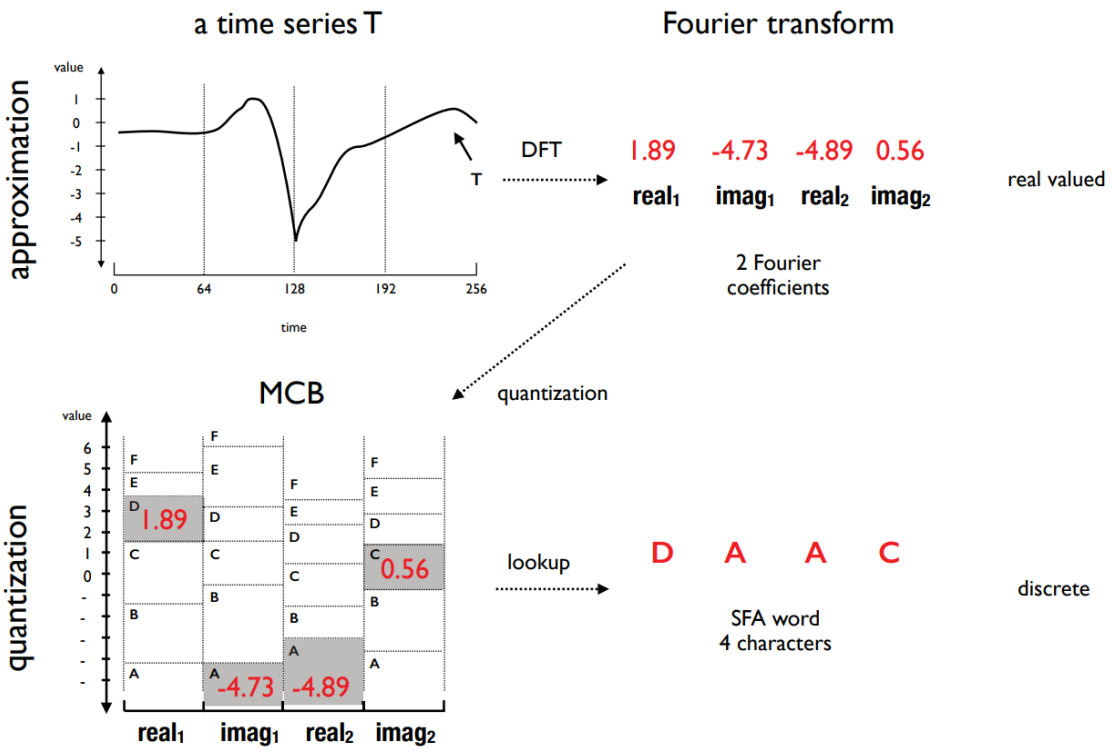
Each node contains
- # vectors
- segmentation **SG**
- synopsis **Z**

Each Leaf node also :
- stores its raw vectors in a separate disk file

Symbolic Fourier Approximation (SFA)
Summarization

Publications

Schafer-EDBT'12



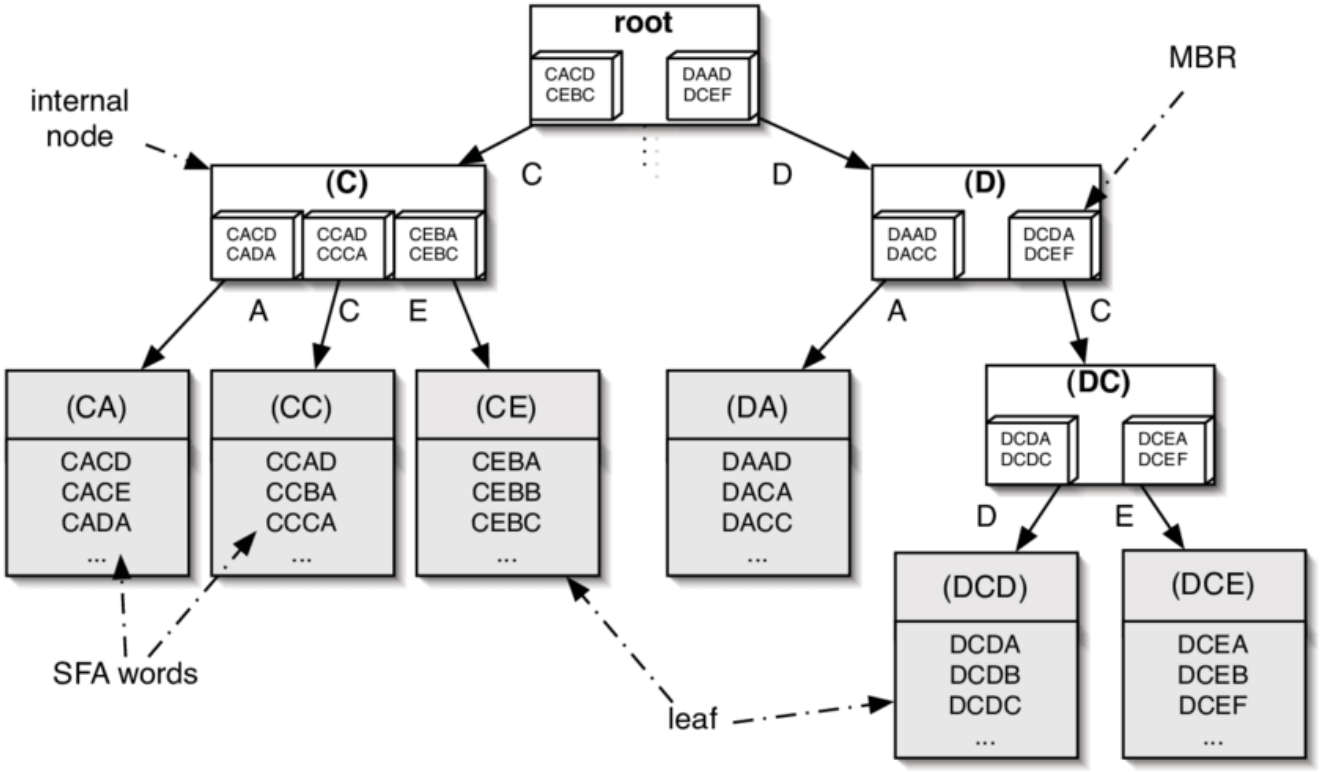
The SFA representation*

*https://www2.informatik.hu-berlin.de/~schaefpa/talks/scalable_classification.pptx

SFA
Indexing

Publications

Schafer-EDBT'12



The SFA Trie*

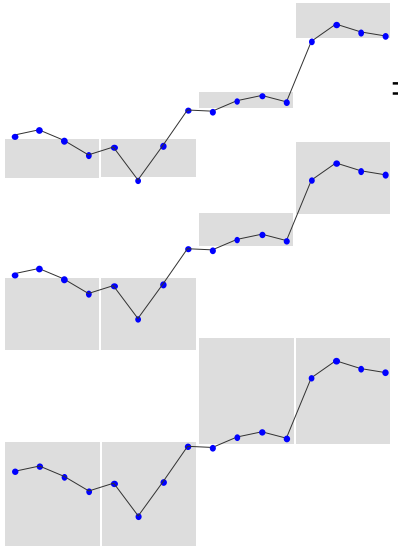
*https://www2.informatik.hu-berlin.de/~schaefpa/talks/scalable_classification.pptx

Publications

Shieh-KDD'08

iSAX Family
iSAX Summarization

- based on *i*SAX representation, which offers a bit-aware, quantized, multi-resolution representation with variable granularity



= { 6, 6, 3, 0} = {110 ,110 ,0111 ,000}

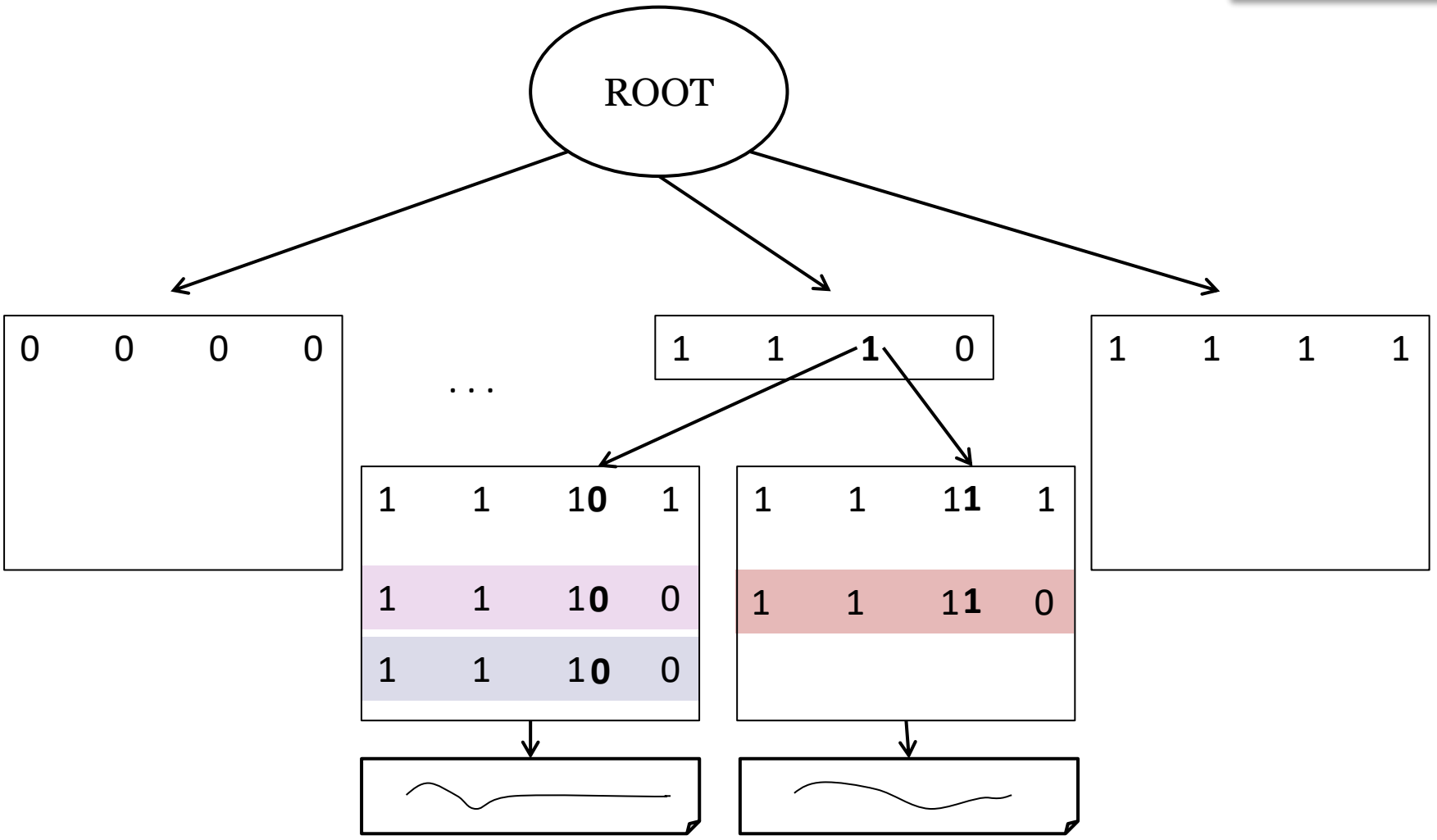
= { 3, 3, 1, 0} = {11 ,11 ,011 ,00 }

= { 1, 1, 0, 0} = {1 ,1 ,0 ,0 }

iSAX-Indexing

Publications

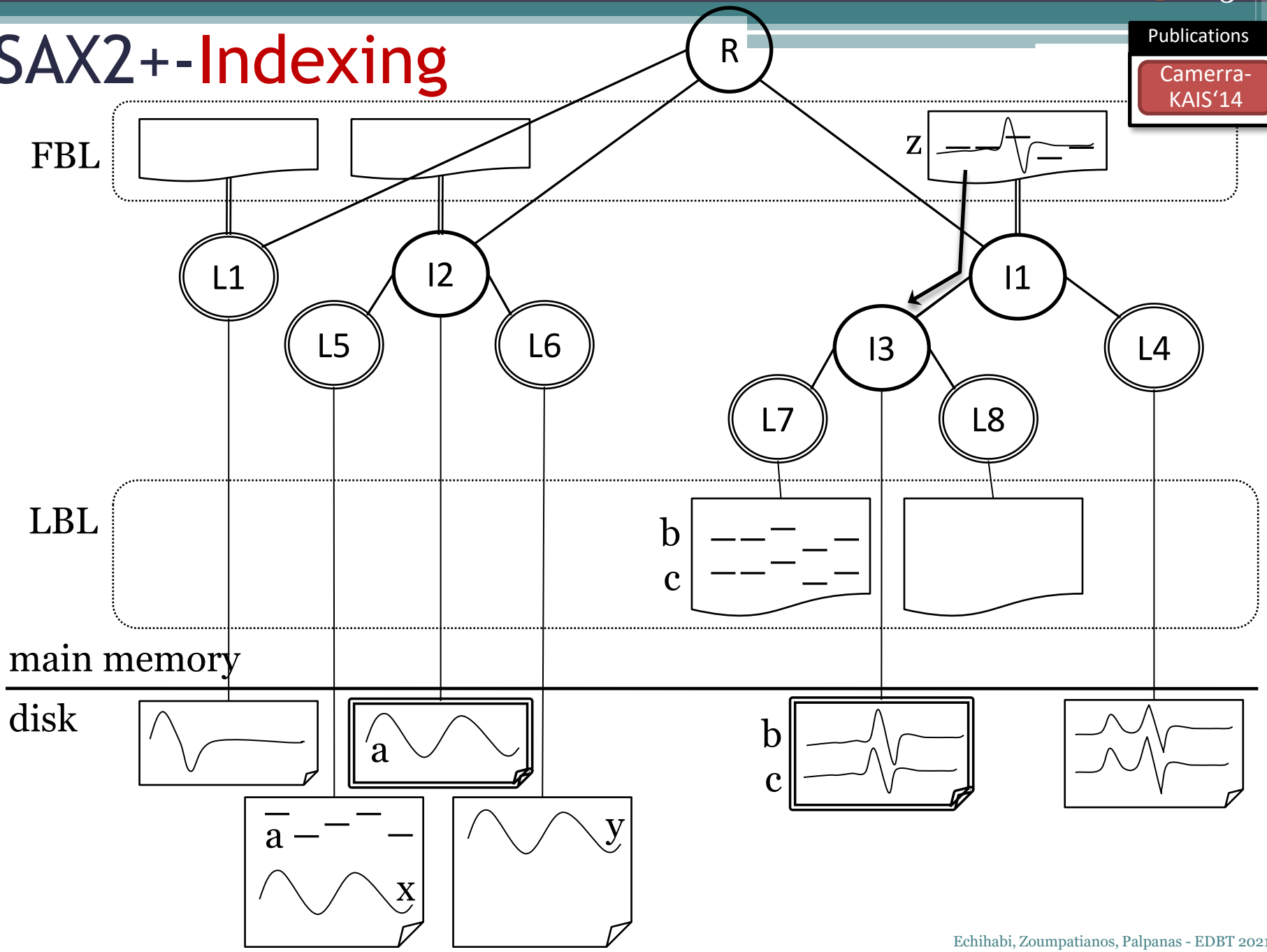
Shieh-KDD'08



iSAX2+-Indexing

Publications

Camerra-KAIS'14

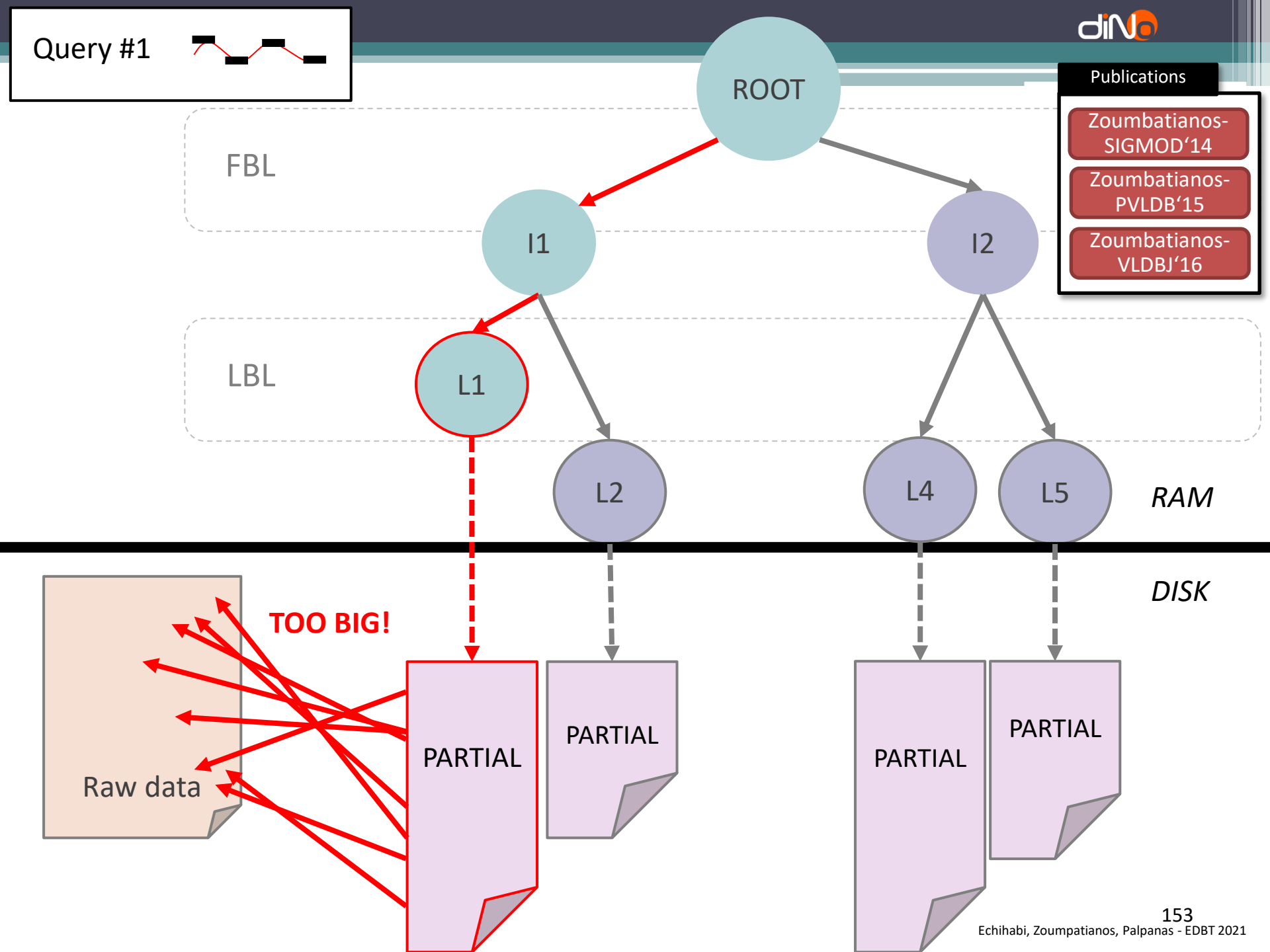


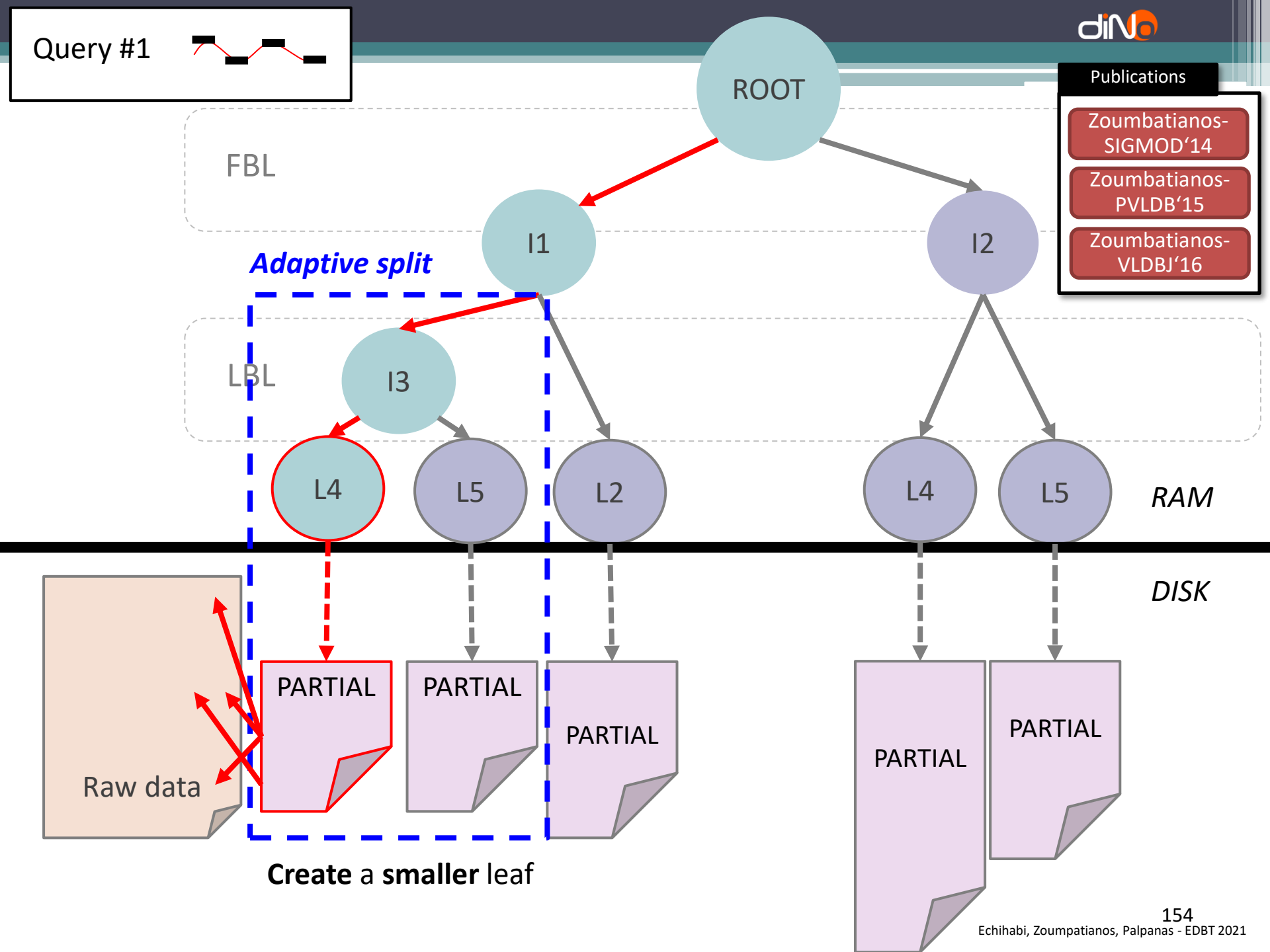
Publications

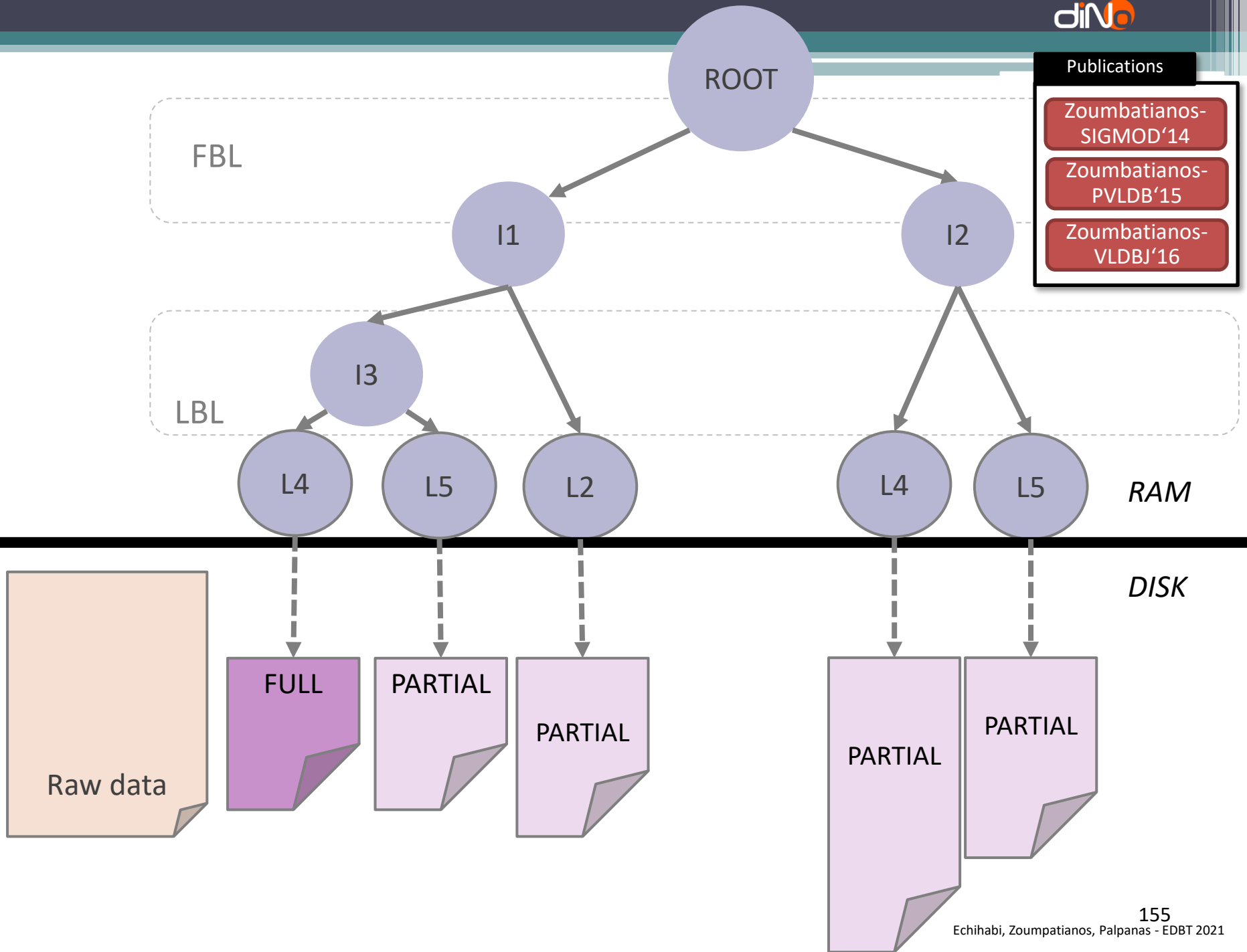
- Zoumbatianos-SIGMOD'14
- Zoumbatianos-PVLDB'15
- Zoumbatianos-VLDBJ'16

ADS+

- novel paradigm for building a data series index
  - does not build entire index and then answer queries
  - starts answering queries by building the part of the index needed by those queries
- still guarantees correct answers
- intuition for proposed solution
  - builds index using only *i*SAX summaries; uses large leaf size
  - postpones leaf materialization to query time
    - only materialize (at query time) leaves needed by queries
  - parts that are queried more are refined more
    - use smaller leaf sizes (reduced leaf materialization and query answering costs)


Query #1
ROOT
FBL
I1
I2
LBL
L1
L2
L4
L5
RAM
DISK
Publications
Zoumbatianos-SIGMOD'14
Zoumbatianos-PVLDB'15
Zoumbatianos-VLDBJ'16
TOO BIG!
Raw data
PARTIAL
PARTIAL
PARTIAL
PARTIAL


Query #1
ROOT
FBL
I1
I2
Adaptive split
LBL
I3
L4
L5
L2
L4
L5
RAM
DISK
PARTIAL
PARTIAL
PARTIAL
PARTIAL
PARTIAL
Raw data
Create a smaller leaf
Publications
Zoumbatianos-SIGMOD'14
Zoumbatianos-PVLDB'15
Zoumbatianos-VLDBJ'16


ROOT
FBL
I1
I2
LBL
I3
L4
L5
L2
L4
L5
RAM
DISK
Raw data
FULL
PARTIAL
PARTIAL
PARTIAL
PARTIAL
Publications
Zoumbatianos-SIGMOD'14
Zoumbatianos-PVLDB'15
Zoumbatianos-VLDBJ'16

Publications

Kondylakis-PVLDB'18

Kondylakis-SIGMOD'19

Coconut

- current solution for limited memory devices and streaming time series
  - bottom-up, succinct index construction based on sortable summarizations

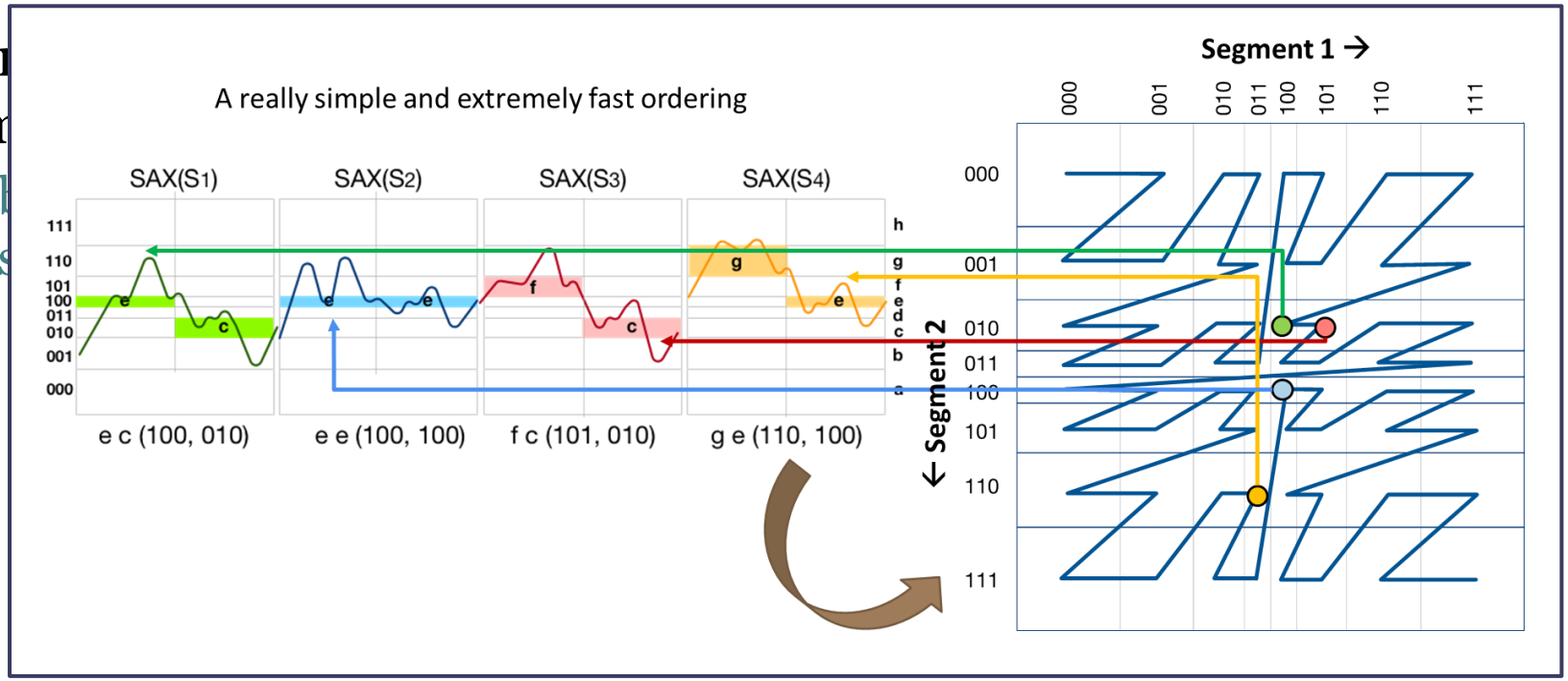
Publications

Kondylakis-PVLDB'18

Kondylakis-SIGMOD'19

Coconut

- cu
  tim
  - I



Publications

Kondylakis-PVLDB'18

Kondylakis-SIGMOD'19

Coconut

- current solution for limited memory devices and streaming time series
  - bottom-up, succinct index construction based on sortable summarizations
  - outperforms state-of-the-art in terms of index space, index construction time, and query answering time

Publications

Kondylakis-PVLDB'18

Kondylakis-SIGMOD'19

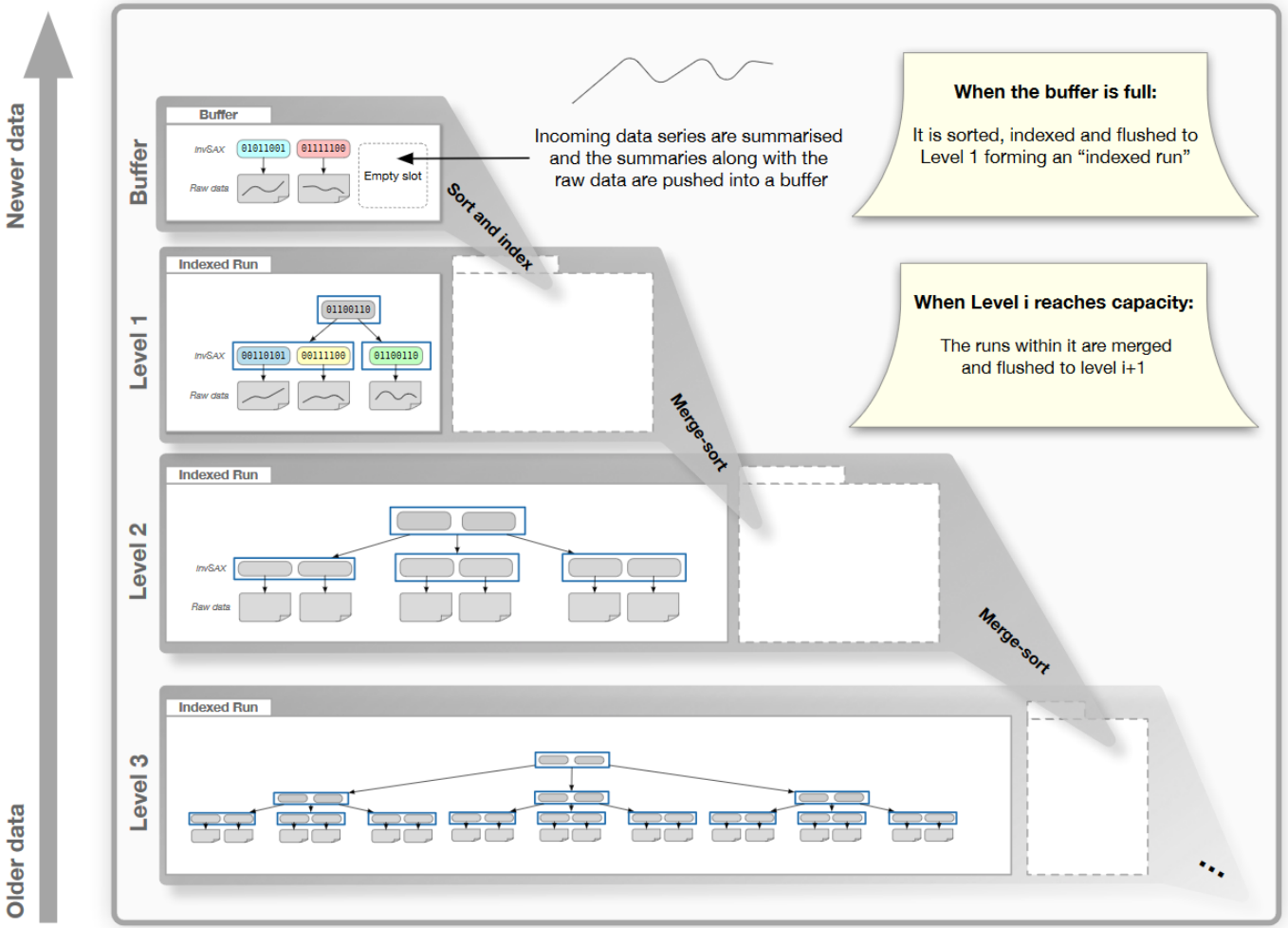
Coconut

- current solution for limited memory devices and streaming time series
  - bottom-up, succinct index construction based on sortable summarizations
  - outperforms state-of-the-art in terms of index space, index construction time, and query answering time
  - compatible with traditional single-dimensional balanced indexes
    - B+-tree, LSM-tree, …

Publications

- Kondylakis-PVLDB'18
- Kondylakis-SIGMOD'19
- Kondylakis-VLDBJ'20

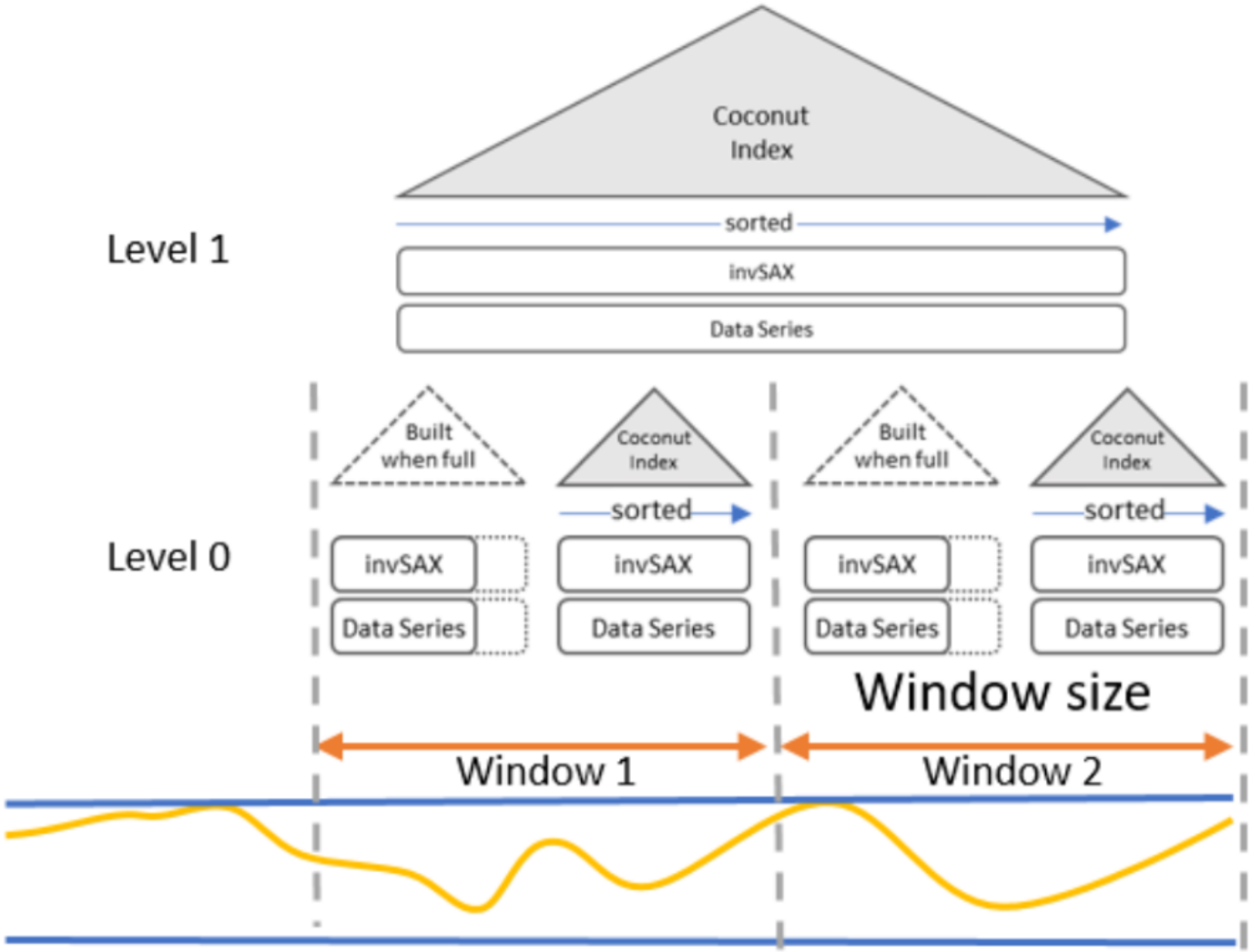
Coconut-LSM



Publications

- Kondylakis-PVLDB'18
- Kondylakis-SIGMOD'19
- Kondylakis-VLDBJ'20

Coconut-LSM



Publications

- Linardi-ICDE'18
- Linardi-PVLDB'19
- Linardi-VLDBJ'20

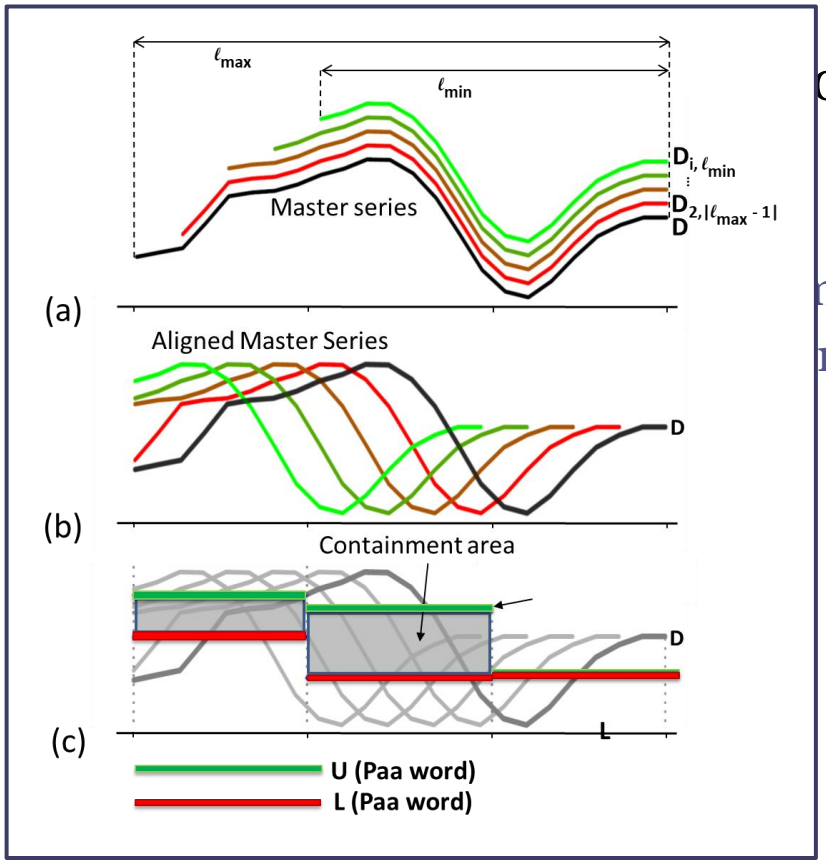
ULISSE

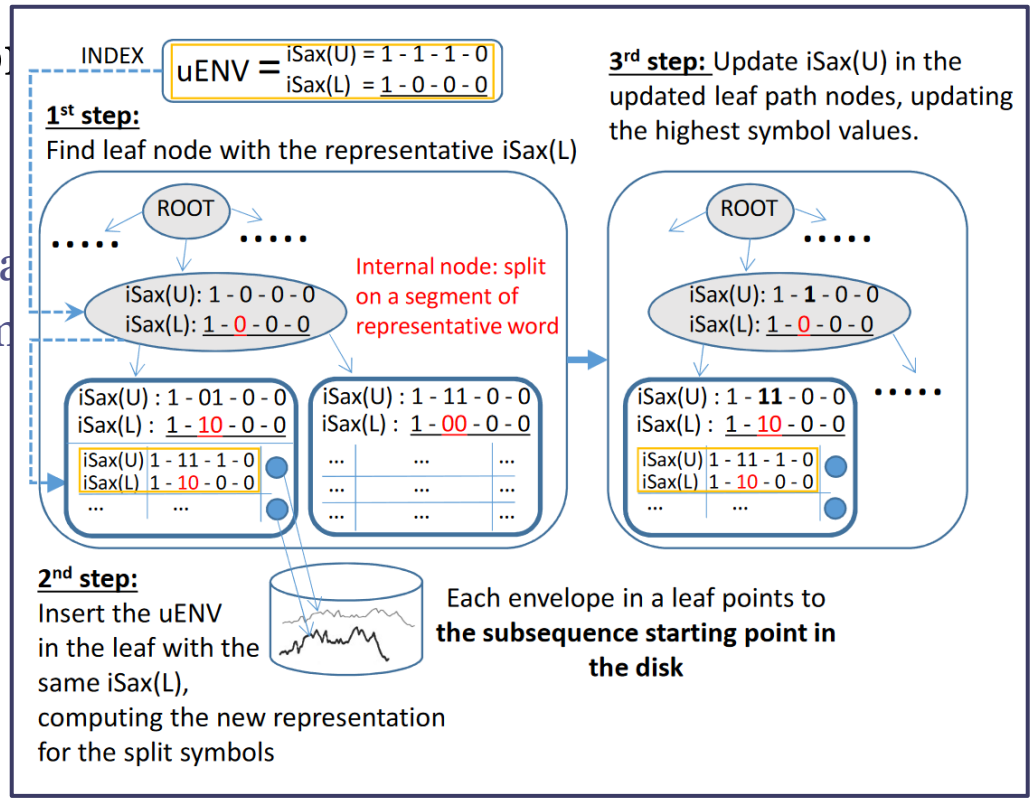
- ULISSE: current solution for variable-length queries
  - single-index support for
    - queries of variable lengths
    - Z-normalized + non Z-normalized data
    - Euclidean + DTW distance measures

ULISSE

Publications

- Linardi-ICDE'18
- Linardi-PVLDB'19
- Linardi-VLDBJ'20





Publications

- Linardi-ICDE'18
- Linardi-PVLDB'19
- Linardi-VLDBJ'20

ULISSE

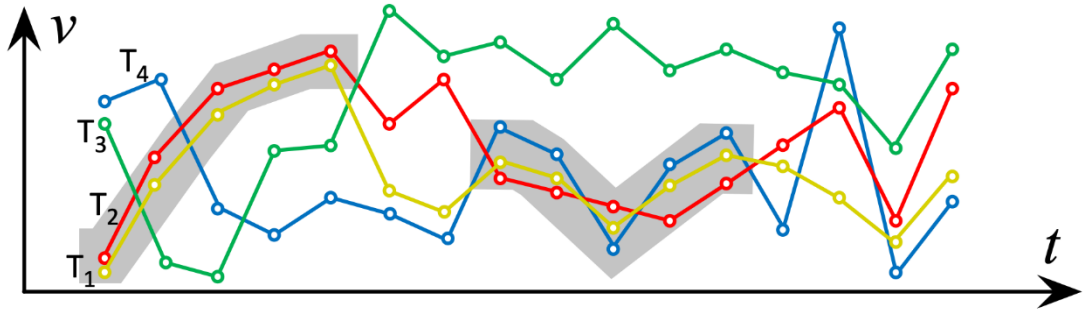
- ULISSE: current solution for variable-length queries
  - single-index support for
    - queries of variable lengths
    - Z-normalized + non Z-normalized data
    - Euclidean + DTW distance measures
  - orders of magnitude faster than competing approaches

Publications

Chatzigeorgakidis et al. SSTD'19

Co-Evolving Subsequences

- discover subsequences, where distance between points is always < ε



Publications

Chatzigeorgakidis et al. SSTD'19

Co-Evolving Subsequences

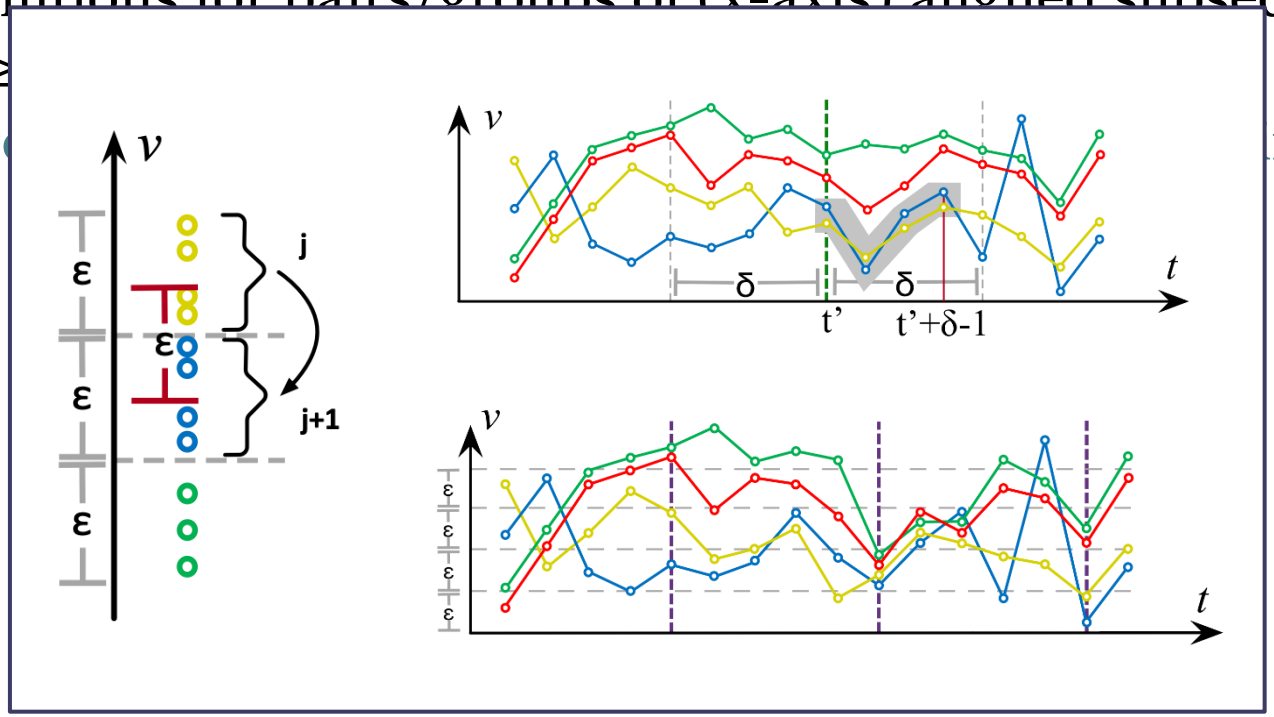
- discover subsequences, where distance between points is always $< \varepsilon$
- SL/CP: solutions for pairs/groups of (x-axis) aligned subsequences of length $\geq \delta$, within large collections of (short) data series
  - prunes search space by discretizing values, and using checkpoints

Publications

Chatzigeorgakidis et al. SSTD'19

Co-Evolving Subsequences

- discover subsequences, where distance between points is always < ε
- SL/CP: solutions for pairs/groups of (x-axis) aligned subsequences of length ≥
  - prunes s… …nts



Publications

Chatzigeorgakidis et al. SSTD'19

Chatzigeorgakidis et al. SIGSPATIAL/GIS'19

Co-Evolving Subsequences

- discover subsequences, where distance between points is always $< \varepsilon$
- SL/CP: solutions for pairs/groups of (x-axis) aligned subsequences of length $\geq \delta$, within large collections of (short) data series
  - prunes search space by discretizing values, and using checkpoints
- SBTSR-Tree: solution for (x-axis) aligned subsequences within large collections of (short) data series, which are geo-located
  - R-Tree like index on segmented data series, with bit-vectors that mark continuity of same series across segments

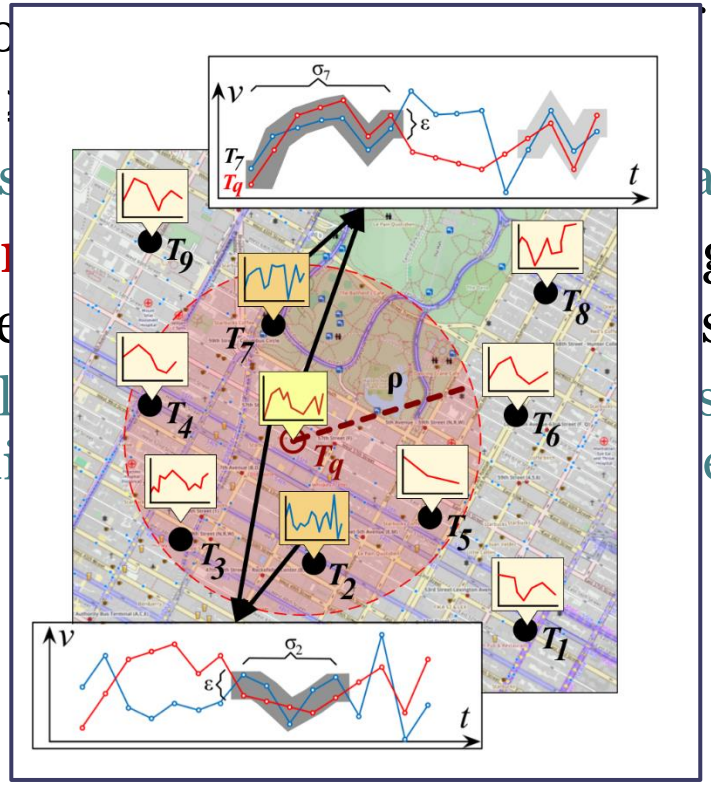
Publications

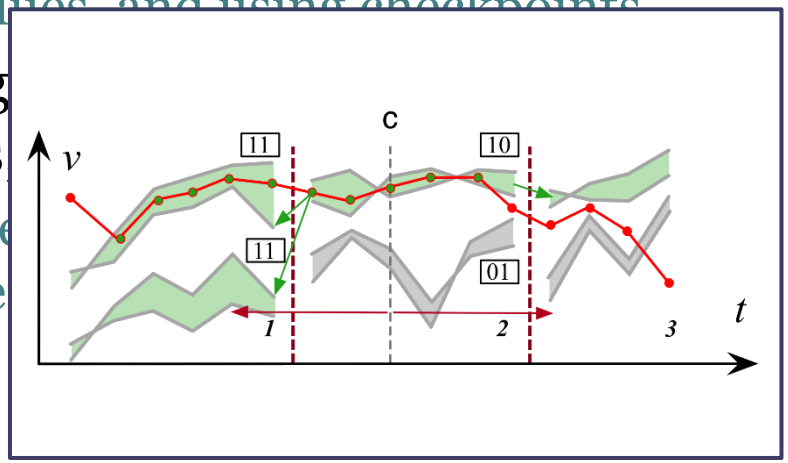
Chatzigeorgakidis et al. SSTD'19

Chatzigeorgakidis et al. SIGSPATIAL/GIS'19

Co-Evolving Subsequences

- discover subsequences, where distance between points is always < ε
- SL/CP: sc (x-axis) aligned subsequences of length of (short) data series
  - prunes s alues, and using checkpoints
- SBTSR-Tr g large colle s
  - R-Tree l se continui e





Publications

- Chatzigeorgakidis et al. SSTD'19
- Chatzigeorgakidis et al. SIGSPATIAL/GIS'19
- Chatzigeorgakidis et al. EDBT'21

Co-Evolving Subsequences

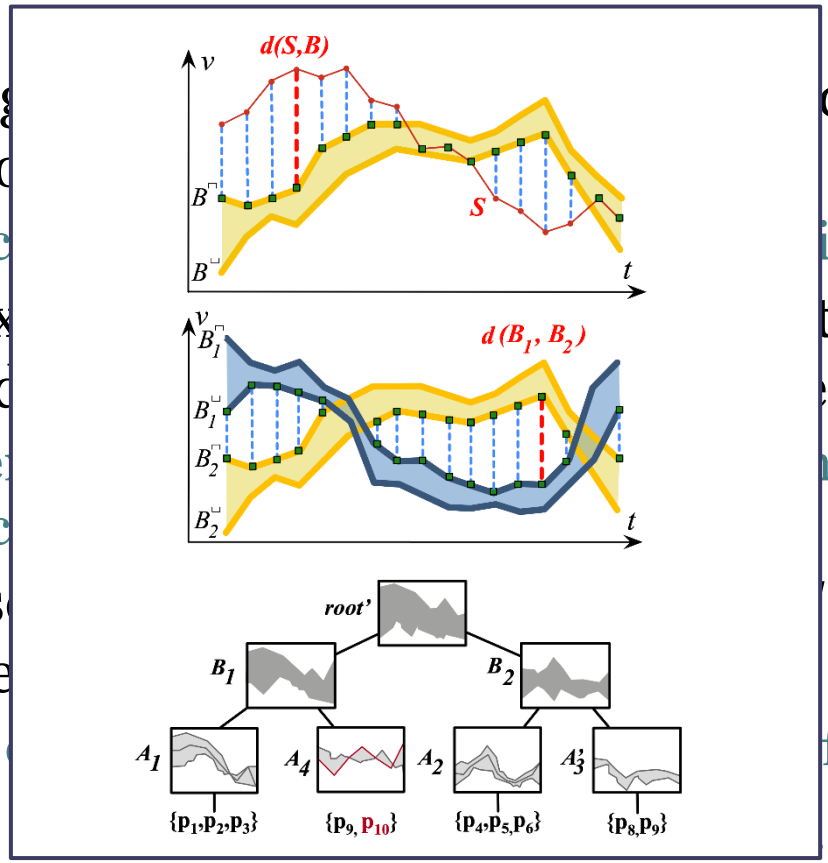
- discover subsequences, where distance between points is always $< \varepsilon$
- SL/CP: solutions for pairs/groups of (x-axis) aligned subsequences of length $\geq \delta$, within large collections of (short) data series
  - prunes search space by discretizing values, and using checkpoints
- SBTSR-Tree: solution for (x-axis) aligned subsequences within large collections of (short) data series, which are geo-located
  - R-Tree like index on segmented data series, with bit-vectors that mark continuity of same series across segments
- TS-Index: solution for subsequences of a long data series T that are similar to a (short) query sequence of length l
  - k-ary balanced index, built on per-point min/max envelopes of all l-length subsequences of T

Publications

- Chatzigeorgakidis et al. SSTD'19
- Chatzigeorgakidis et al. SIGSPATIAL/GIS'19
- Chatzigeorgakidis et al. EDBT'21

Co-Evolving Subsequences

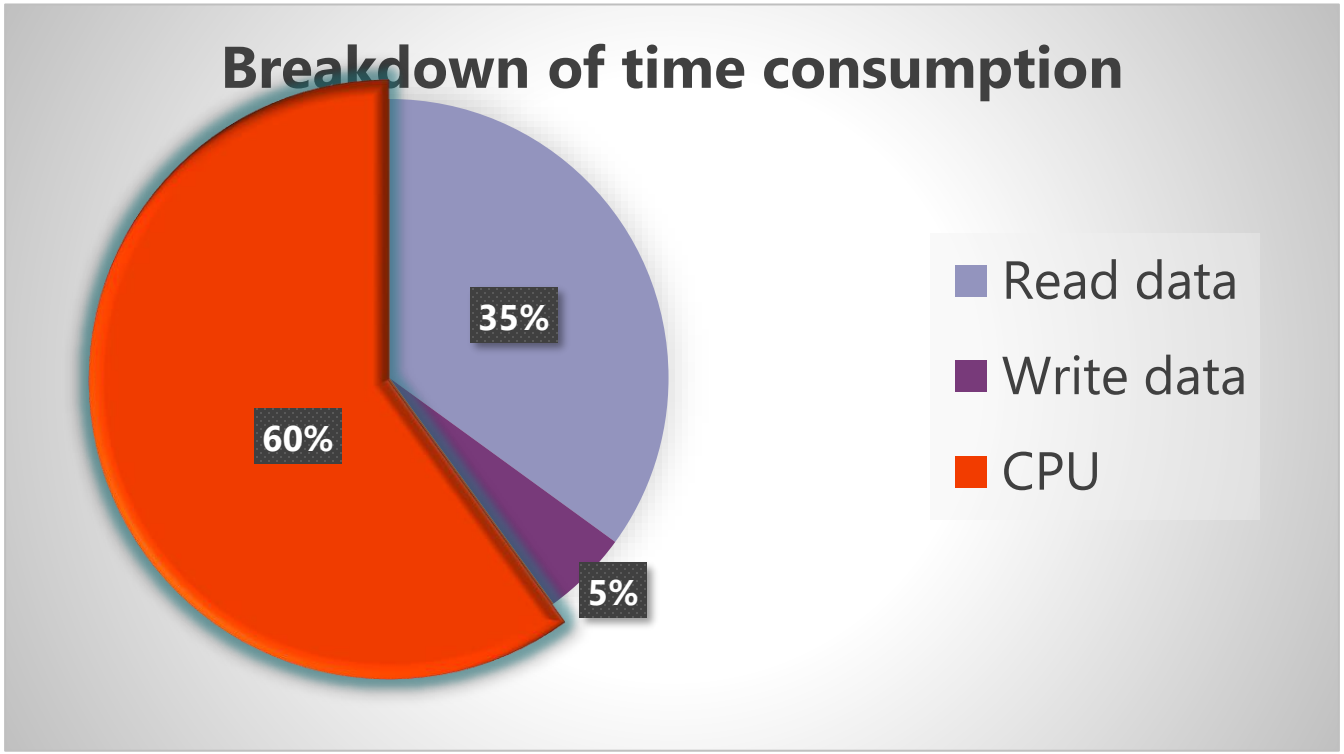
- discover subsequences, where distance between points is always < ε
- SL/CP: solutions for pairs/g … quences of length ≥δ, within large co …
  - prunes search space by disc … ints
- SBTSR-Tree: solution for (x … thin large collections of (short) c … d
  - R-Tree like index on segme … at mark continuity of same series ac …
- TS-Index: solution for subs … that are similar to a (short) query se …
  - k-ary balanced index, built … f all *l*-length subsequences of *T*



Questions?

Data Series Indexing
Parallel & Distributed

ADS Index creation



~60% of time spent in CPU: potential for improvement!

Publications

Peng-BigData'18

Peng-TKDE'20

ParIS+
Parallel Indexing of Sequences

- solution for SIMD, multi-core, multi-socket architectures
  - completely masks out the CPU cost during index creation
  - answers exact queries in the order of few secs on 100GB dataset
    - up to 3 orders of magnitude faster then single-core solutions

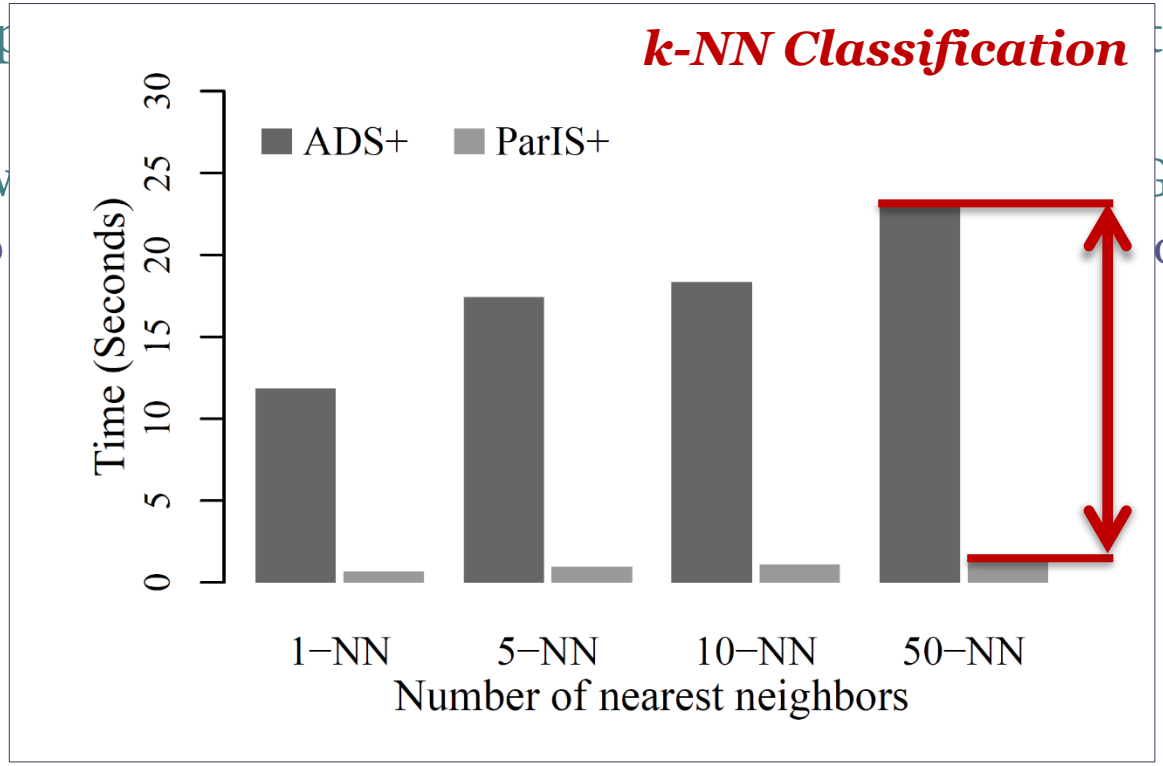
Publications

Peng-BigData'18

Peng-TKDE'20

ParIS+
Parallel Indexing of Sequences

- solution for SIMD, multi-core, multi-socket architectures
  - comp … tion
  - answ … GB dataset
    - up … ons



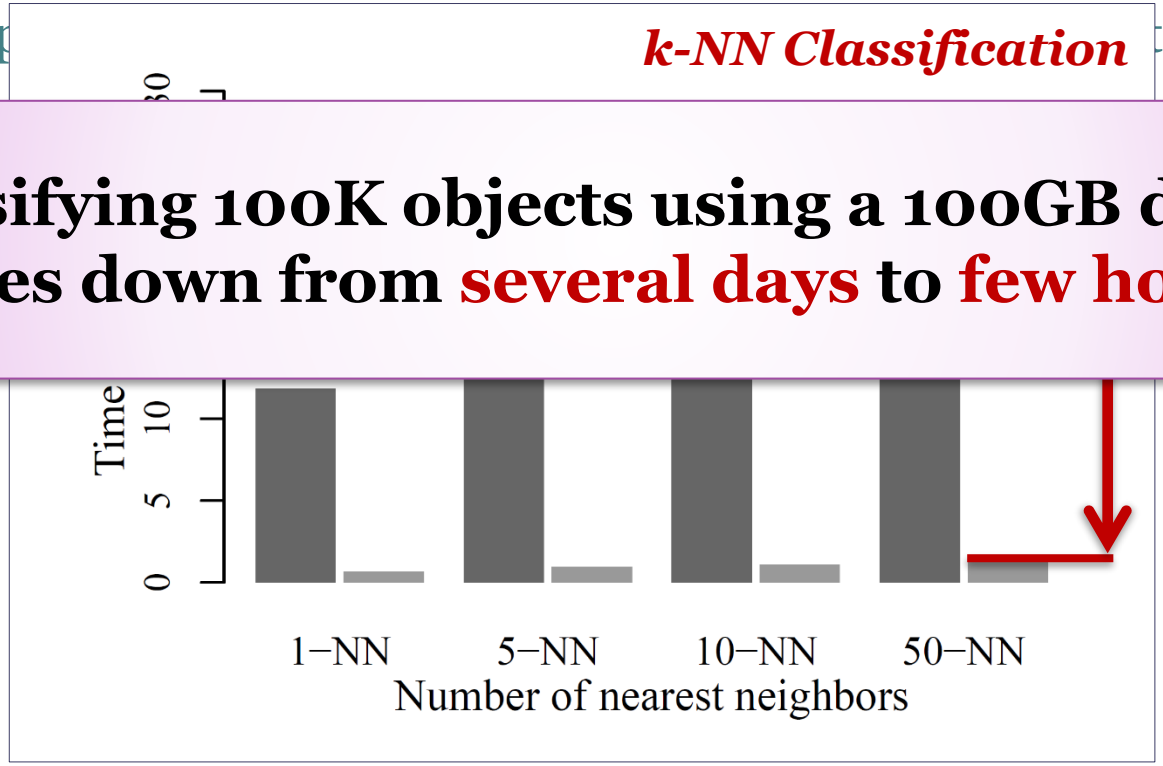
Publications

Peng-BigData'18

Peng-TKDE'20

ParIS+
Parallel Indexing of Sequences

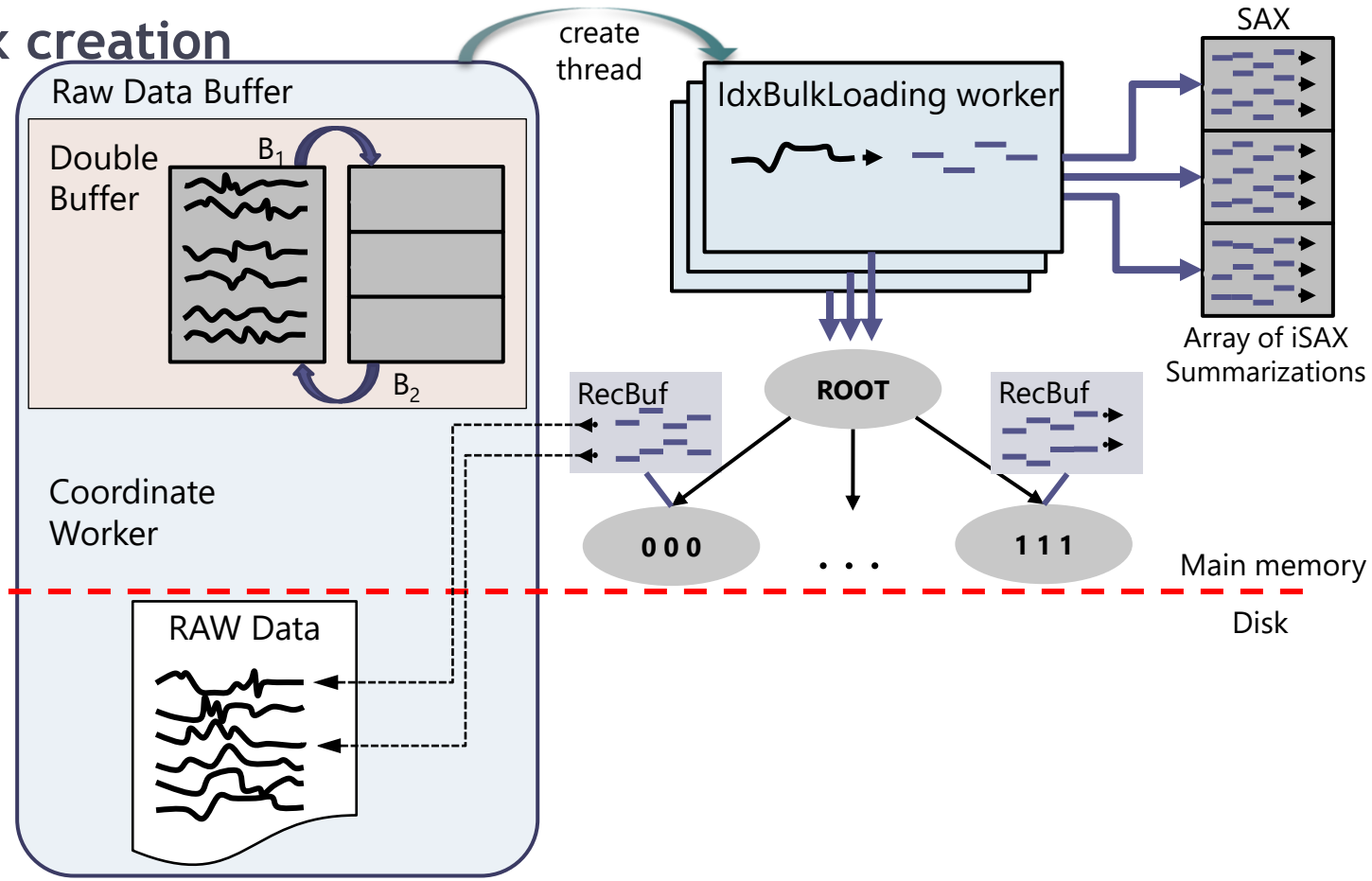
- solution for SIMD, multi-core, multi-socket architectures
  - comp… …tion



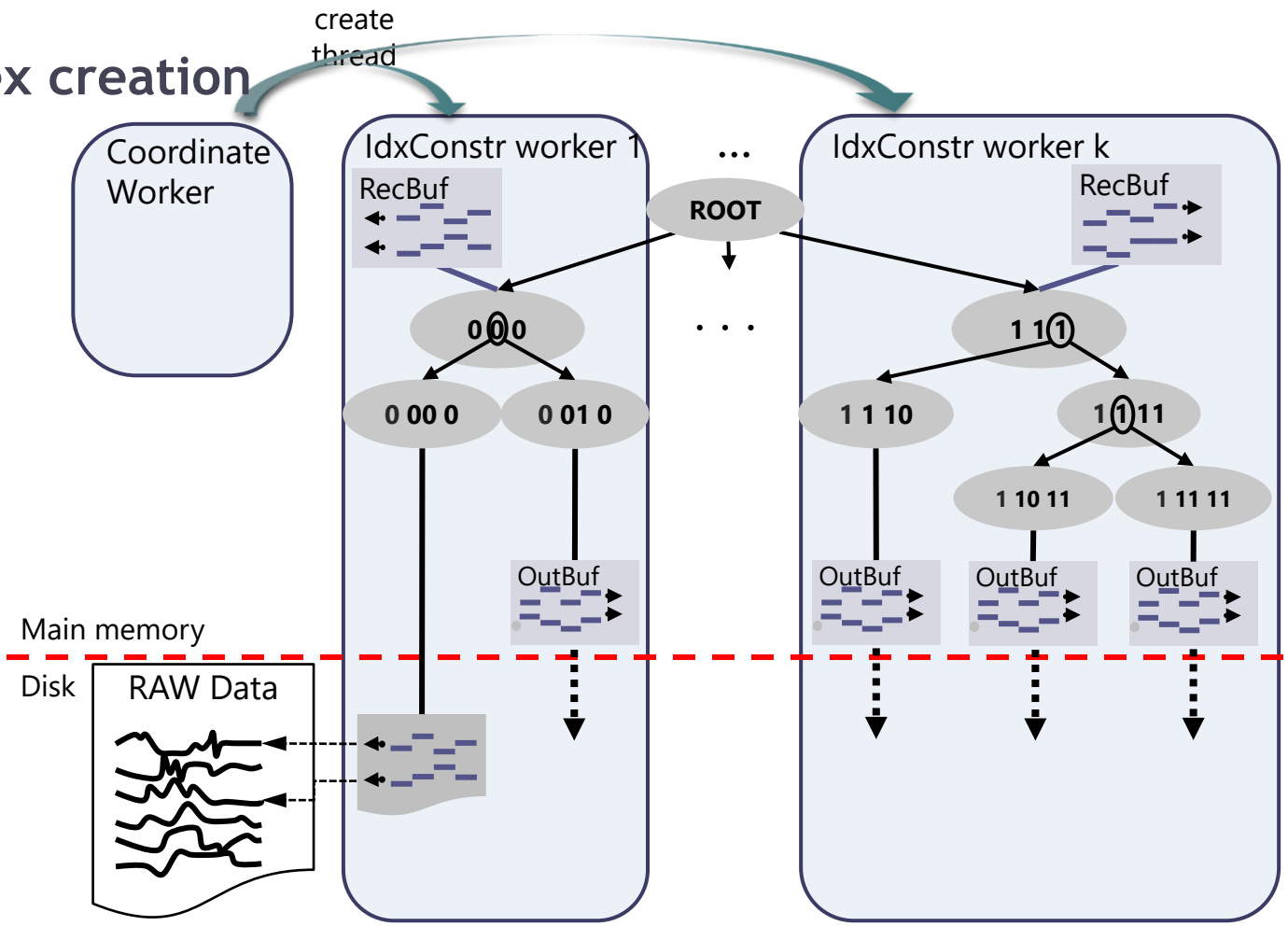
classifying 100K objects using a 100GB dataset goes down from several days to few hours!

18x faster

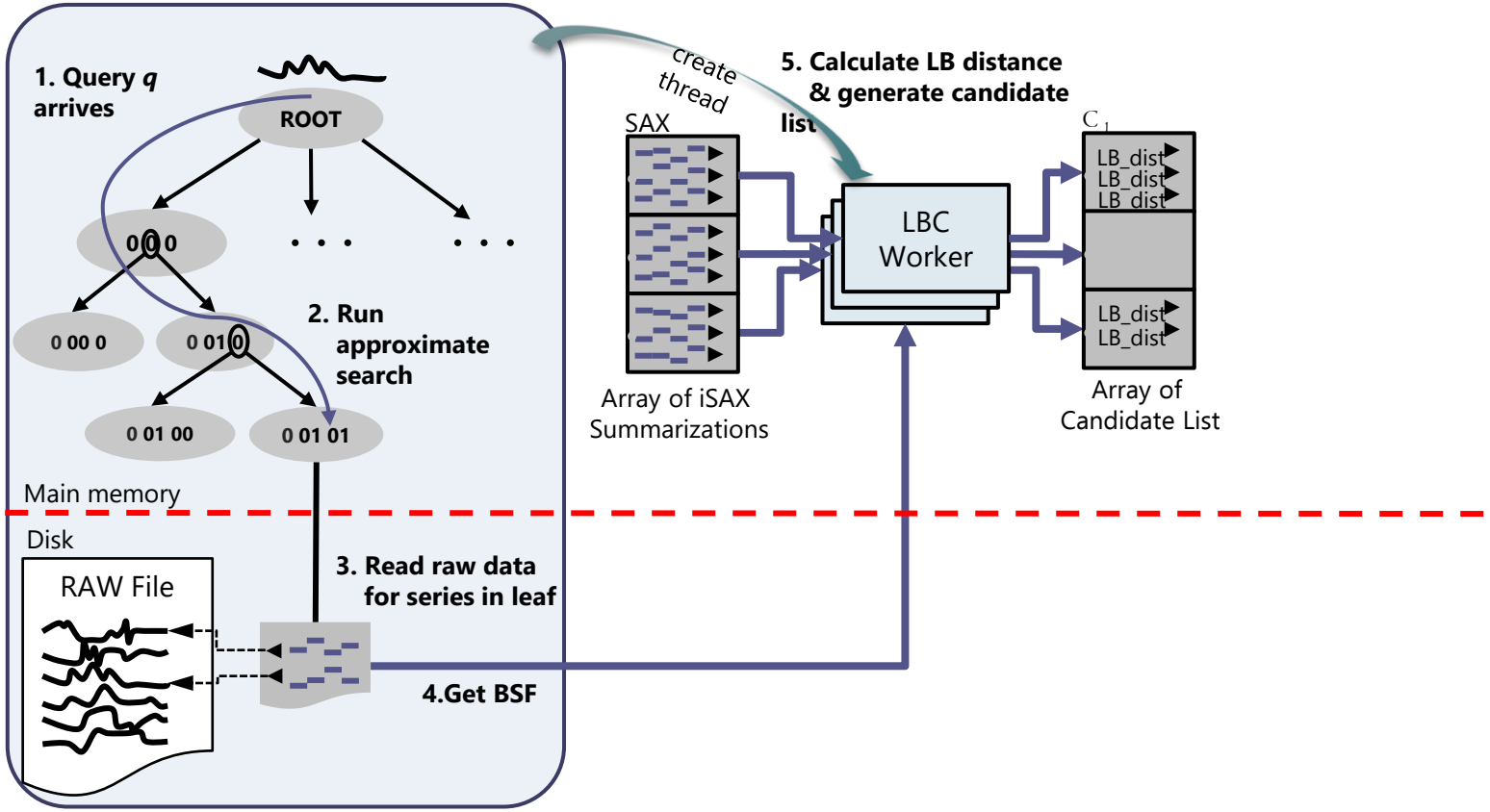
Index creation



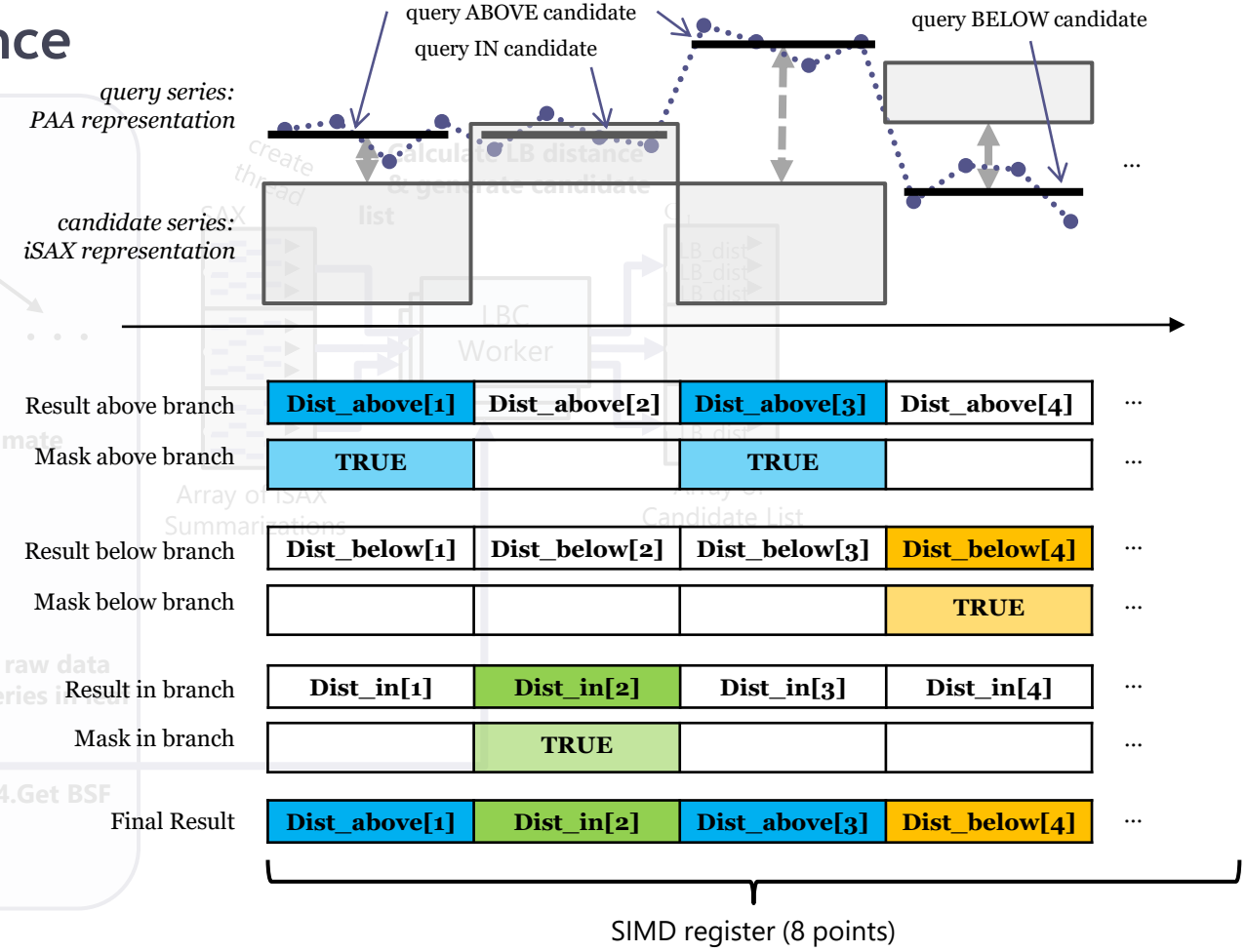
Index creation



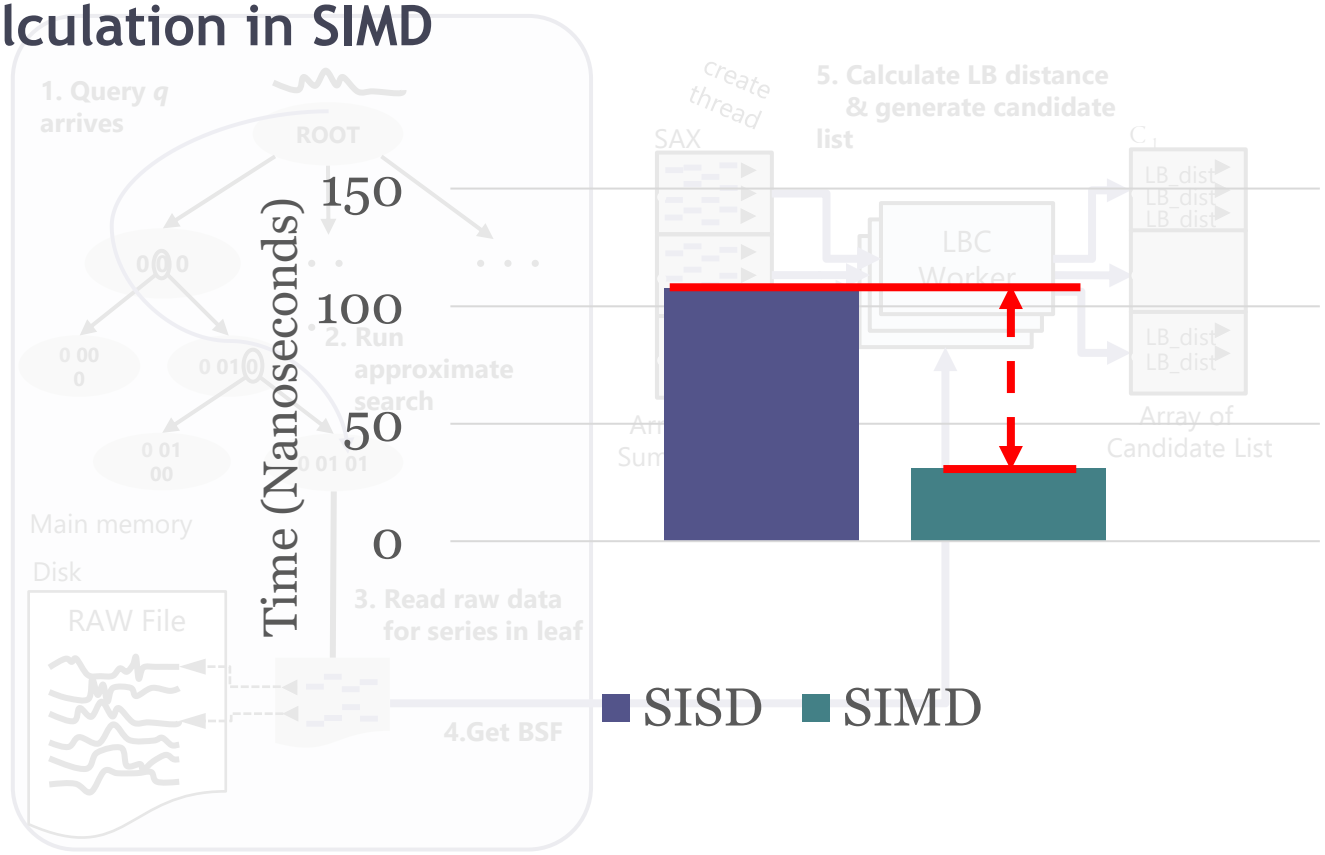
ParIS+ exact query answering



Lower-Bound Distance Calculation in SIMD

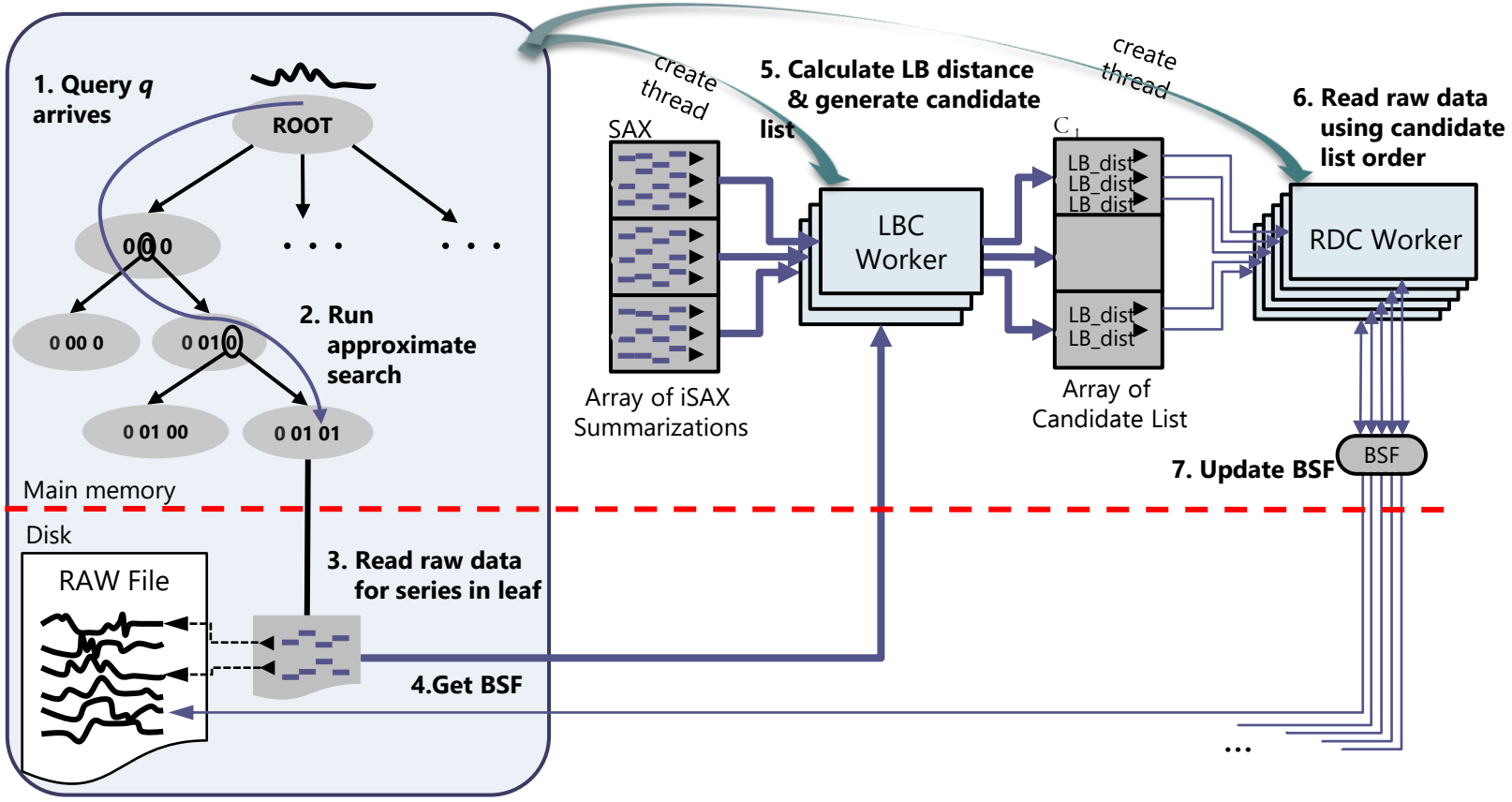


Lower-Bound Distance Calculation in SIMD



SIMD lower bounds are 3.4x faster

ParIS+ exact query answering



Publications

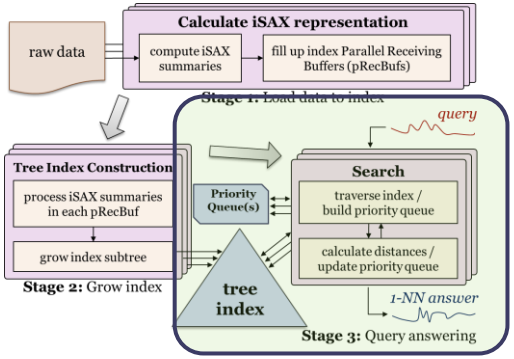
Peng-ICDE'20

Peng-VLDBJ'21

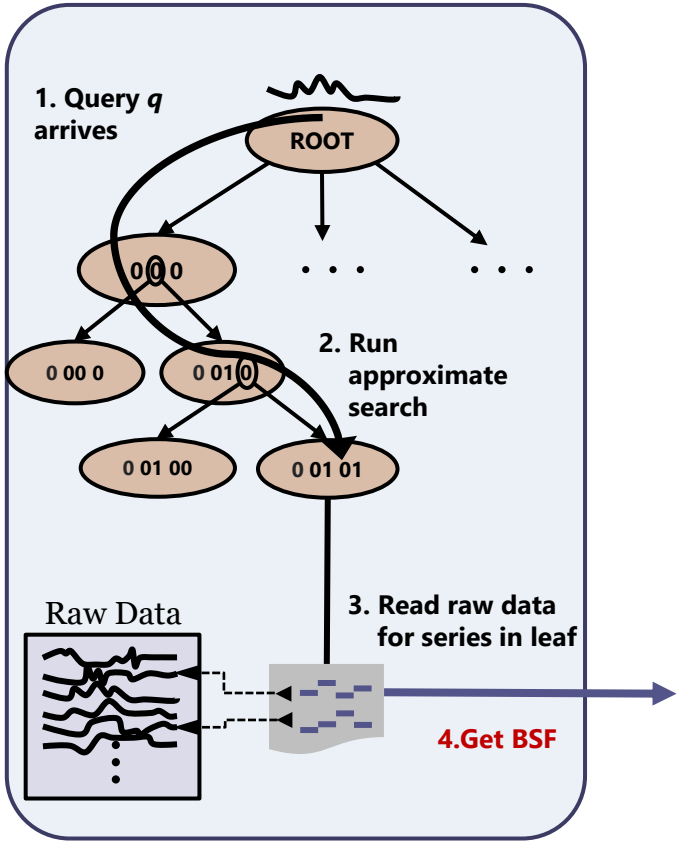
MESSI
In-Memory Data Series Index

- in-memory solution for SIMD, multi-core, multi-socket architectures
  - index-creation algorithm
    - balances workload of different workers, minimizes synchronization cost
  - exact query answering algorithm
    - optimizes tree traversal and pruning
    - minimizes number of lower-bound and real distance calculations
  - answers exact queries at interactive speeds: ~50msec on 100GB
    - up to 11x faster than competing approaches

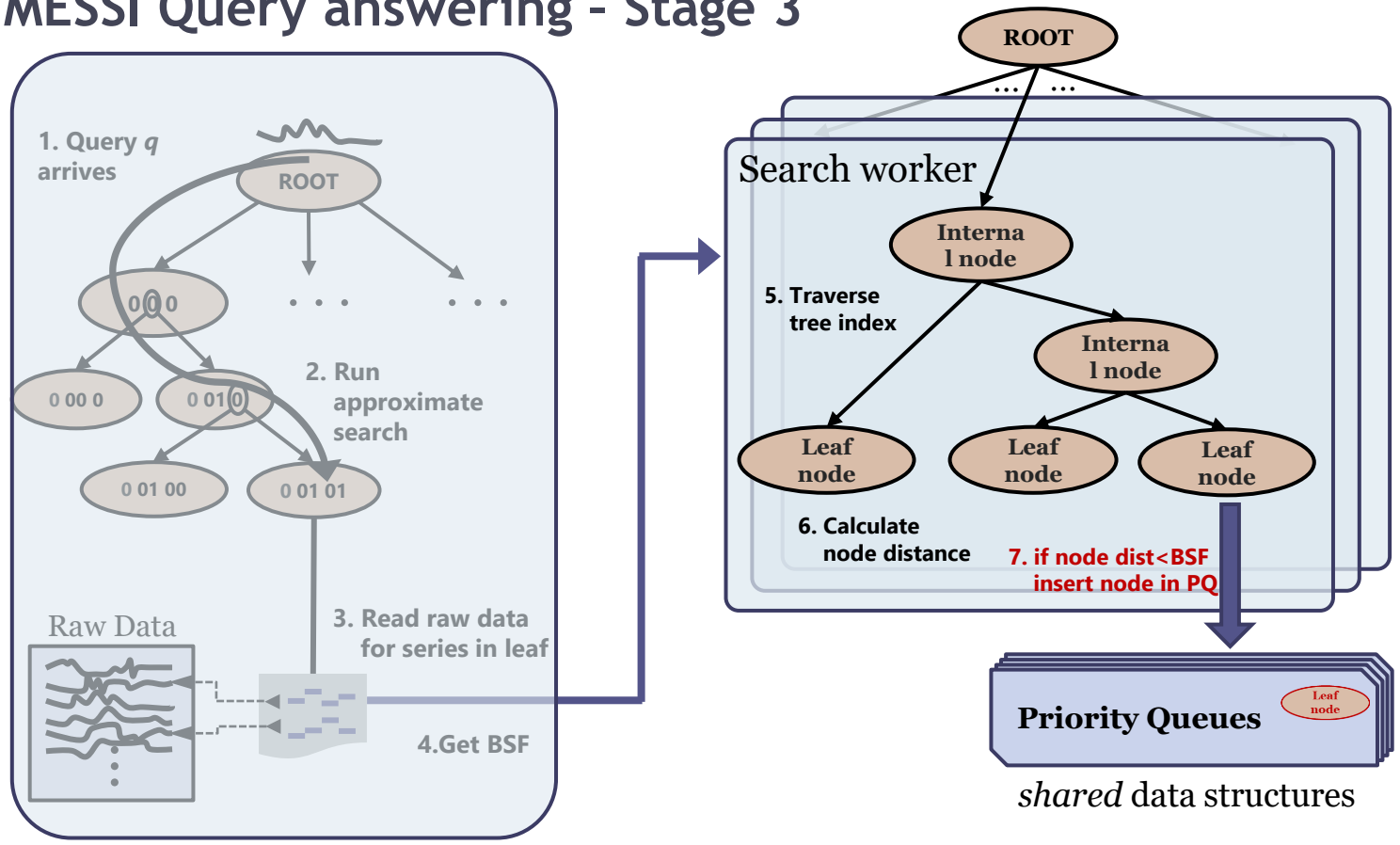
MESSI Query answering - Stage 3



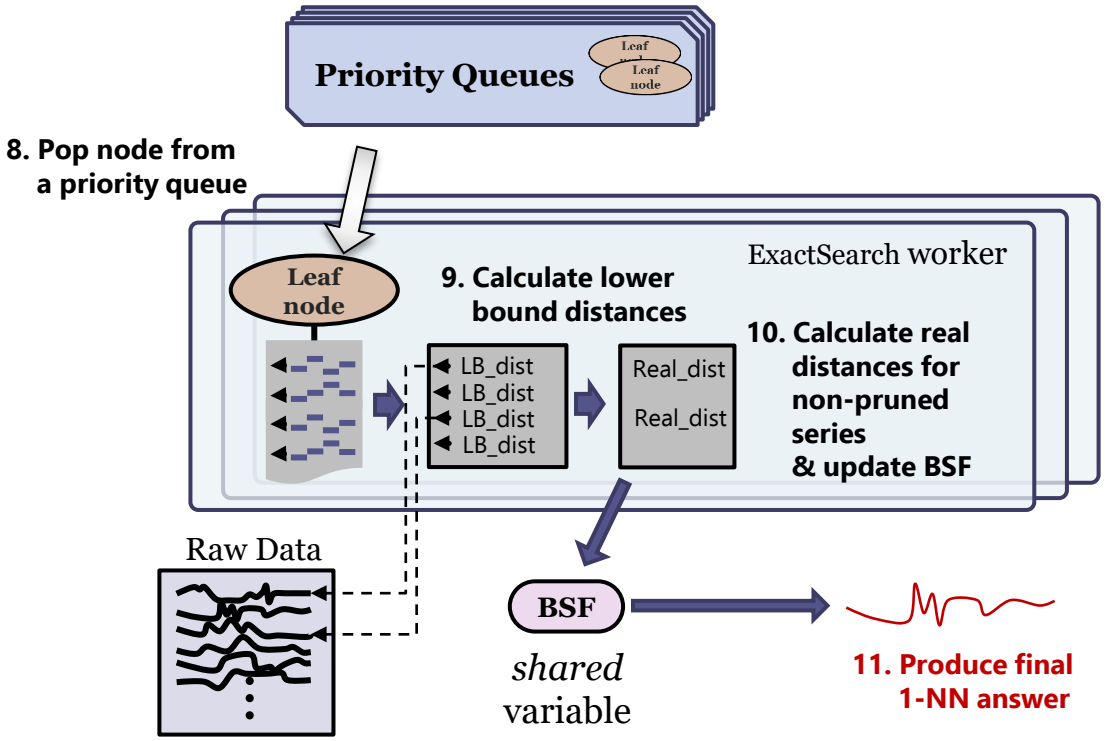
MESSI Query answering - Stage 3



MESSI Query answering - Stage 3



MESSI Query answering - Stage 3



Publications

Peng-
ICDE'21

SING
Sequence Indexing Using GPUs

- in-memory solution for SIMD, multi-core, multi-socket architectures with GPUs (Graphical Processing Units)
  - new exact query answering algorithm
    - CPU-GPU co-processing framework
    - new GPU-friendly lower bound distance calculation algorithm
  - answers exact queries at interactive speeds: ~32msec on 100GB dataset
    - up to 5x faster than competing approaches

GPUs for Data Series Similarity Search

- a natural solution
  - GPUs typically part of modern hardware
  - GPUs offer massive parallelization opportunities
  - data series operations are massively parallelizable

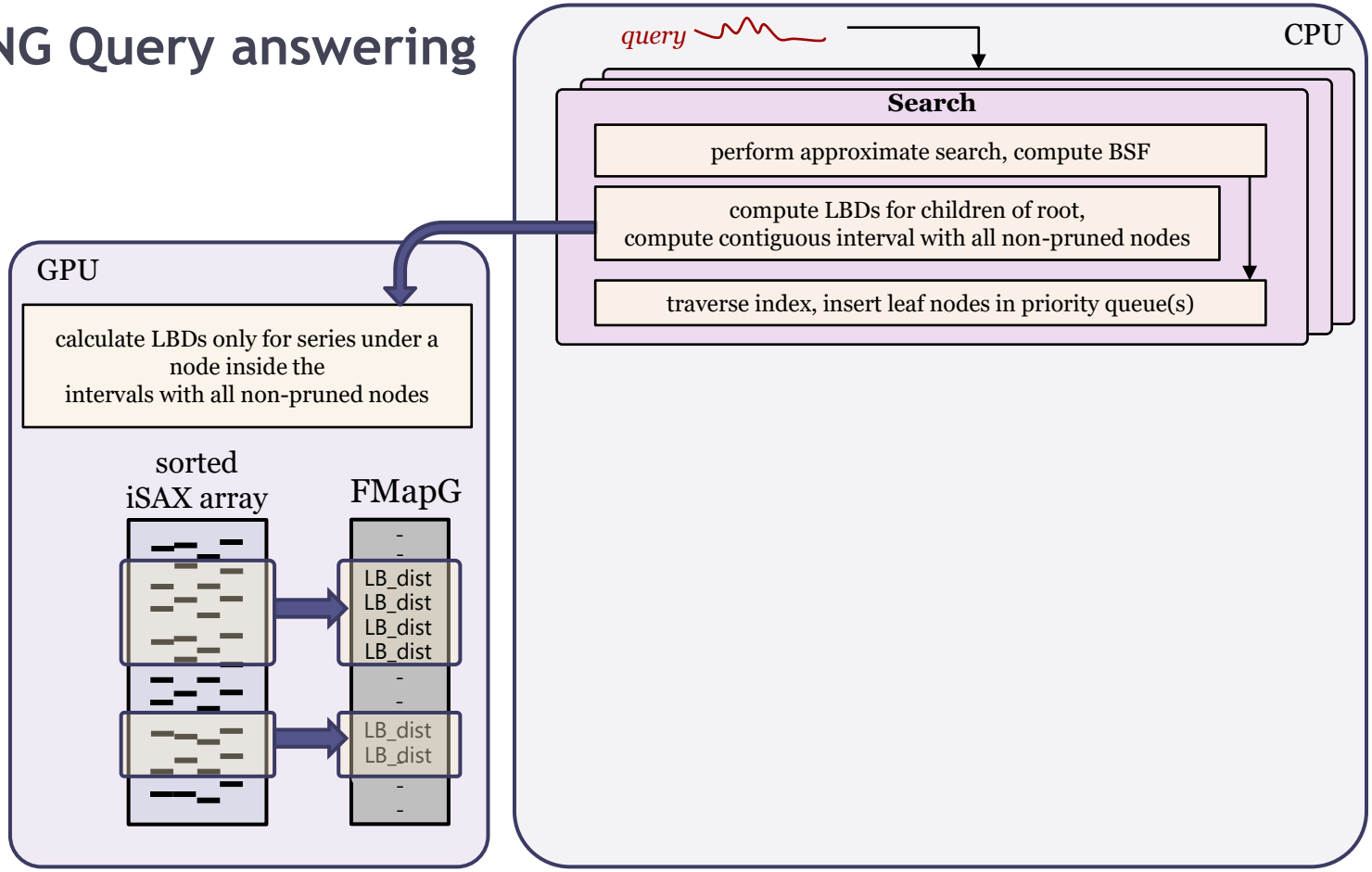
GPUs for Data Series Similarity Search

- a natural solution
  - GPUs typically part of modern hardware
  - GPUs offer massive parallelization opportunities
  - data series operations are massively parallelizable

- challenges
  - Limited GPU memory size (~12GB of RAM for modern GPUs)
    - much smaller than raw data
  - Slow interconnect speeds (PCI-Express 3.0 x16 delivers 10GB/sec)
    - moving raw data needed by individual queries prohibitively expensive
  - non-sophisticated Streaming Processors (GPU cores)
    - not suited for supporting complex data structures/branching
    - very limited in-core fast memory

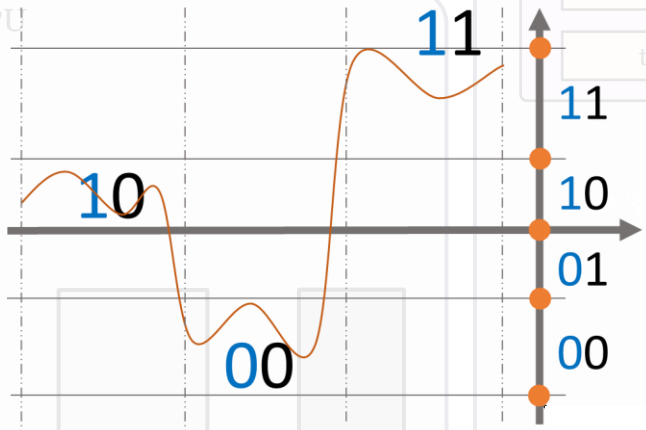
  - trade-offs will change as GPU and interconnects technology advances

SING Query answering



SING Query answering

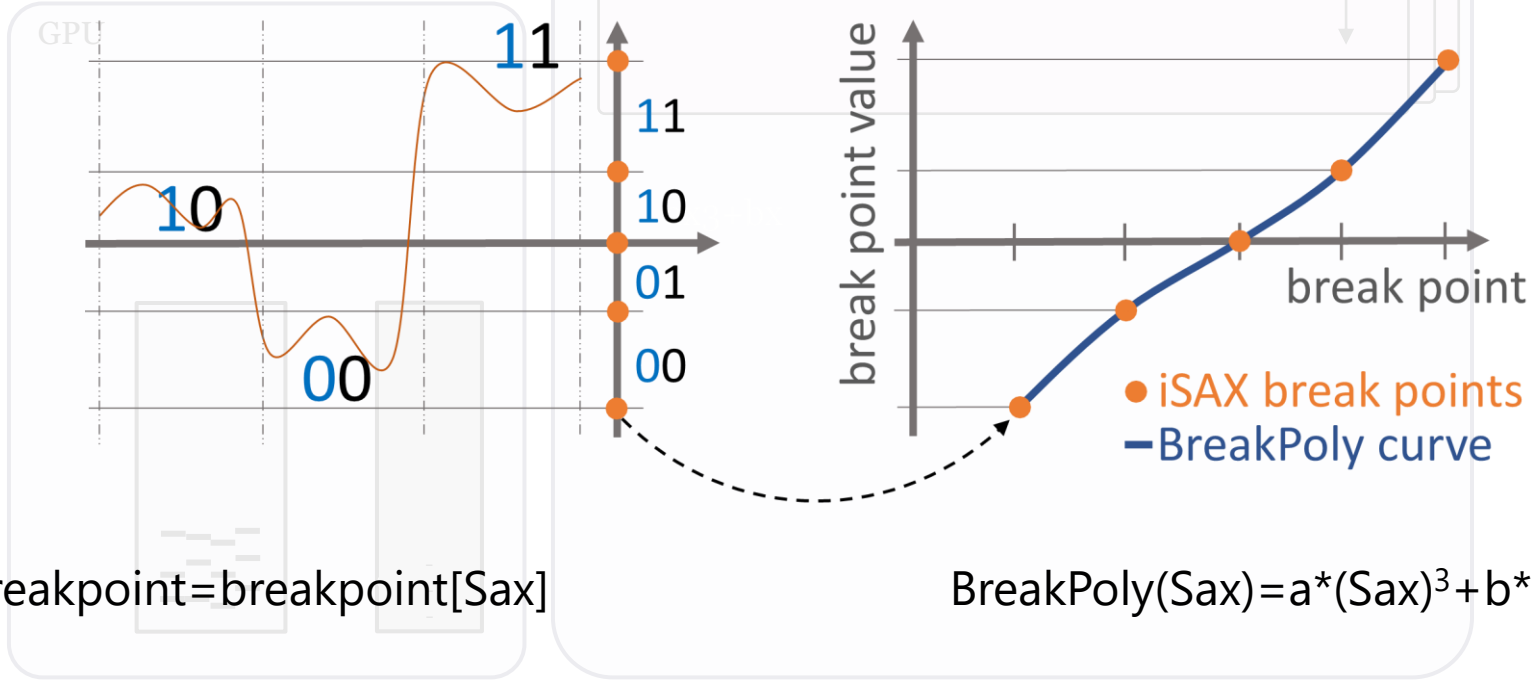
Lower Bound Distance Computations



Breakpoint=breakpoint[Sax]

SING Query answering

GPU-friendly Lower Bound Distance Computations

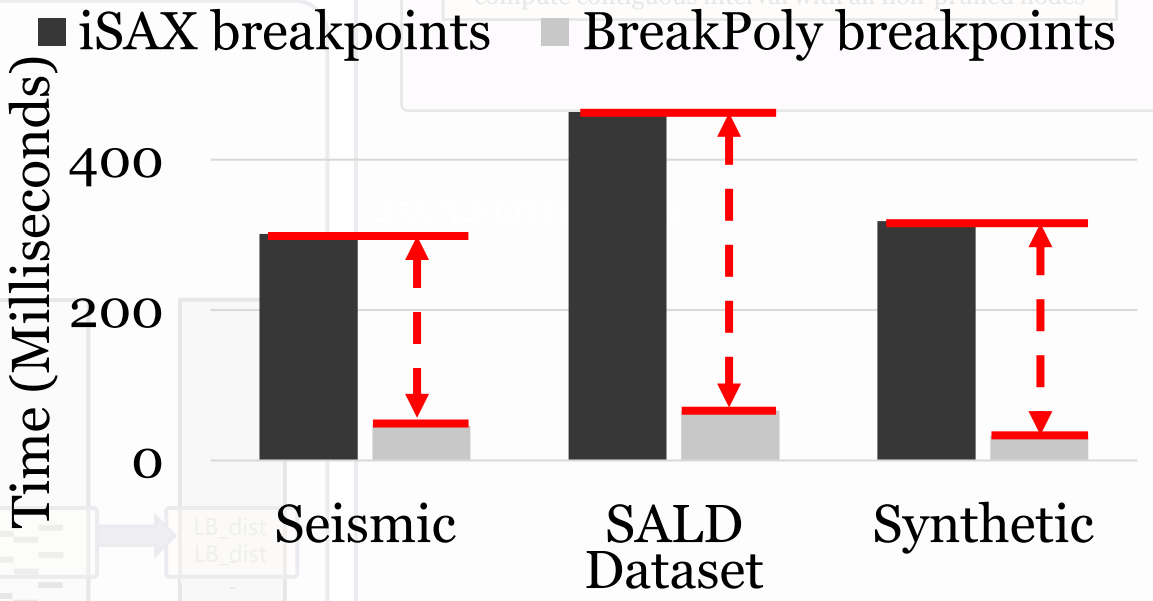


Breakpoint=breakpoint[Sax]

$BreakPoly(Sax)=a*(Sax)^3+b*(Sax)$

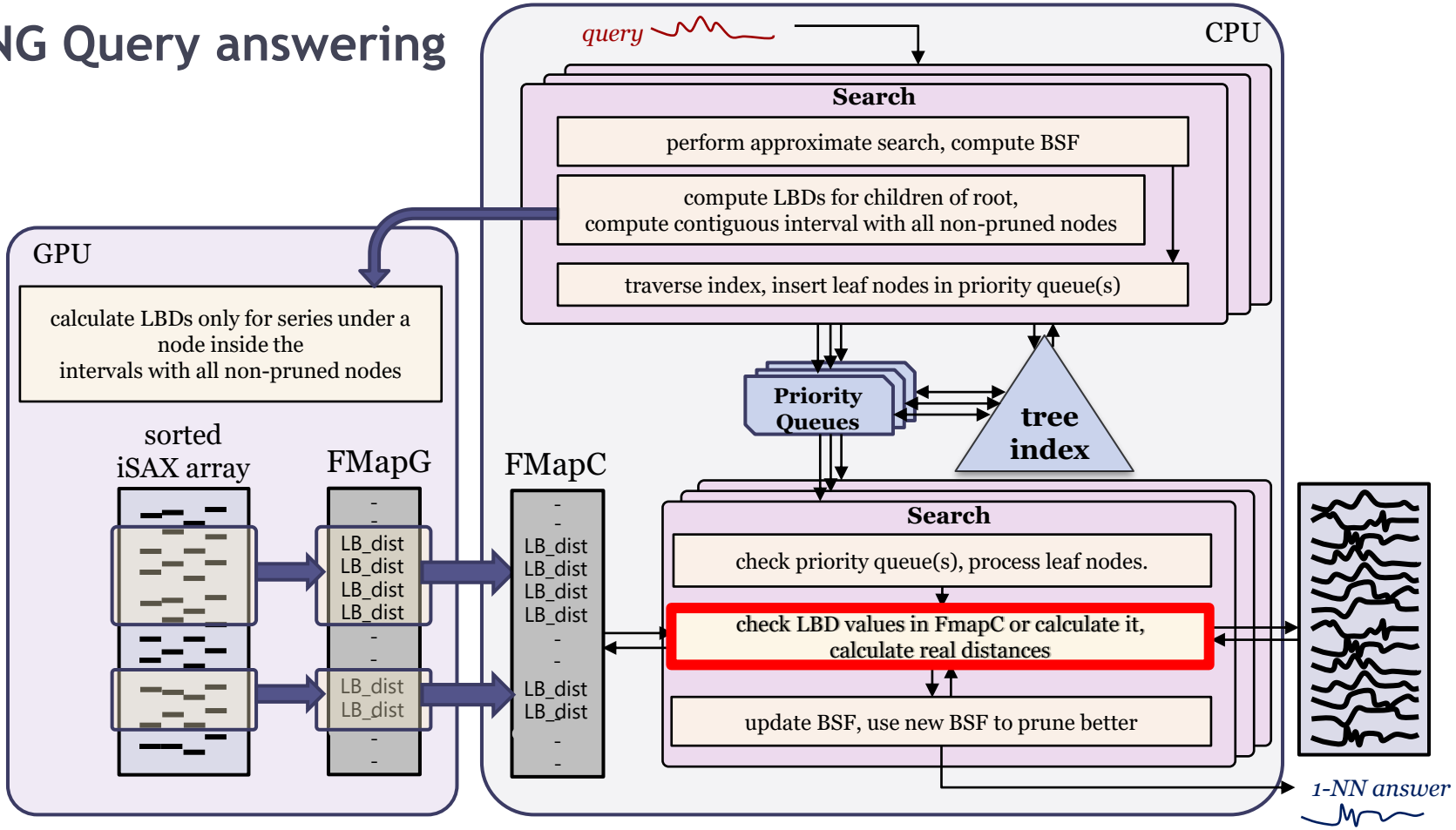
SING Query answering

Time of Lower Bound Distance Computations



GPU with *BreakPoly()* breakpoints is ~10x faster

SING Query answering



DPiSAX
Distributed Partitioned iSAX

Publications

- Yagoubi-ICDM'17
- Yagoubi-TKDE'18
- Lavchenko-KAIS'20

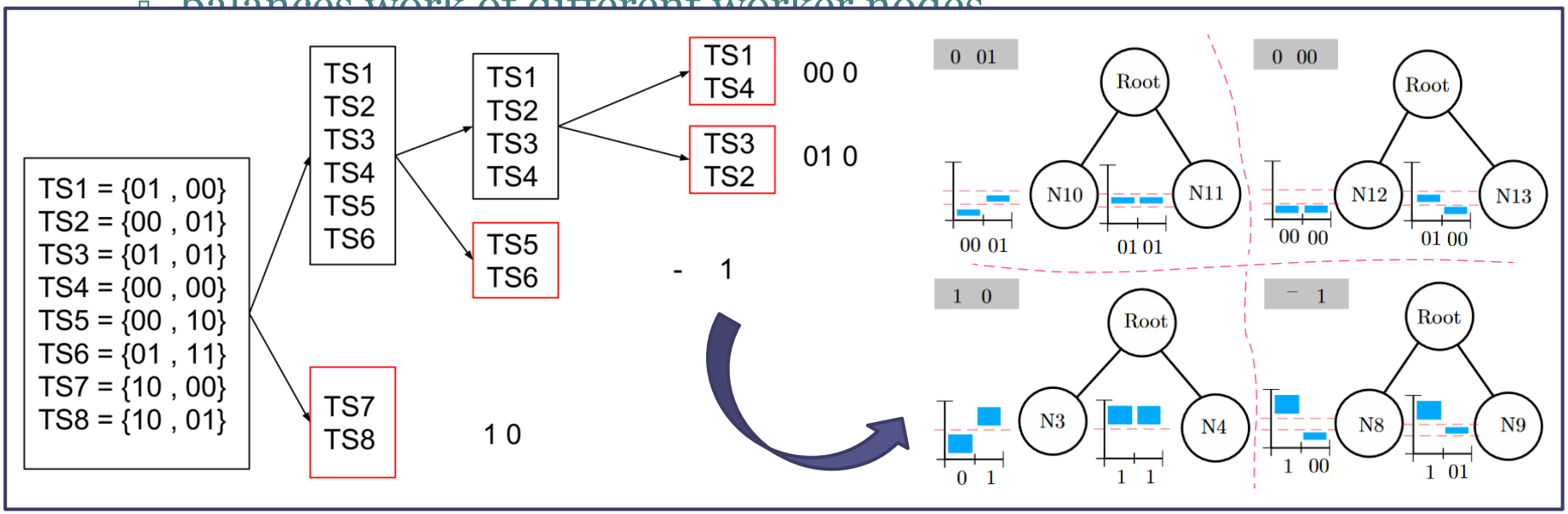
- solution for distributed processing (Spark)
  - balances work of different worker nodes
    - partitions series into uniform groups with parallel sampling (for load balancing)
    - creates in parallel an index for each group (in a different node)

DPiSAX
Distributed Partitioned iSAX

Publications

- Yagoubi-ICDM'17
- Yagoubi-TKDE'18
- Lavchenko-KAIS'20

- solution for distributed processing (Spark)
  - balances work of different worker nodes



DPiSAX
Distributed Partitioned iSAX

Publications

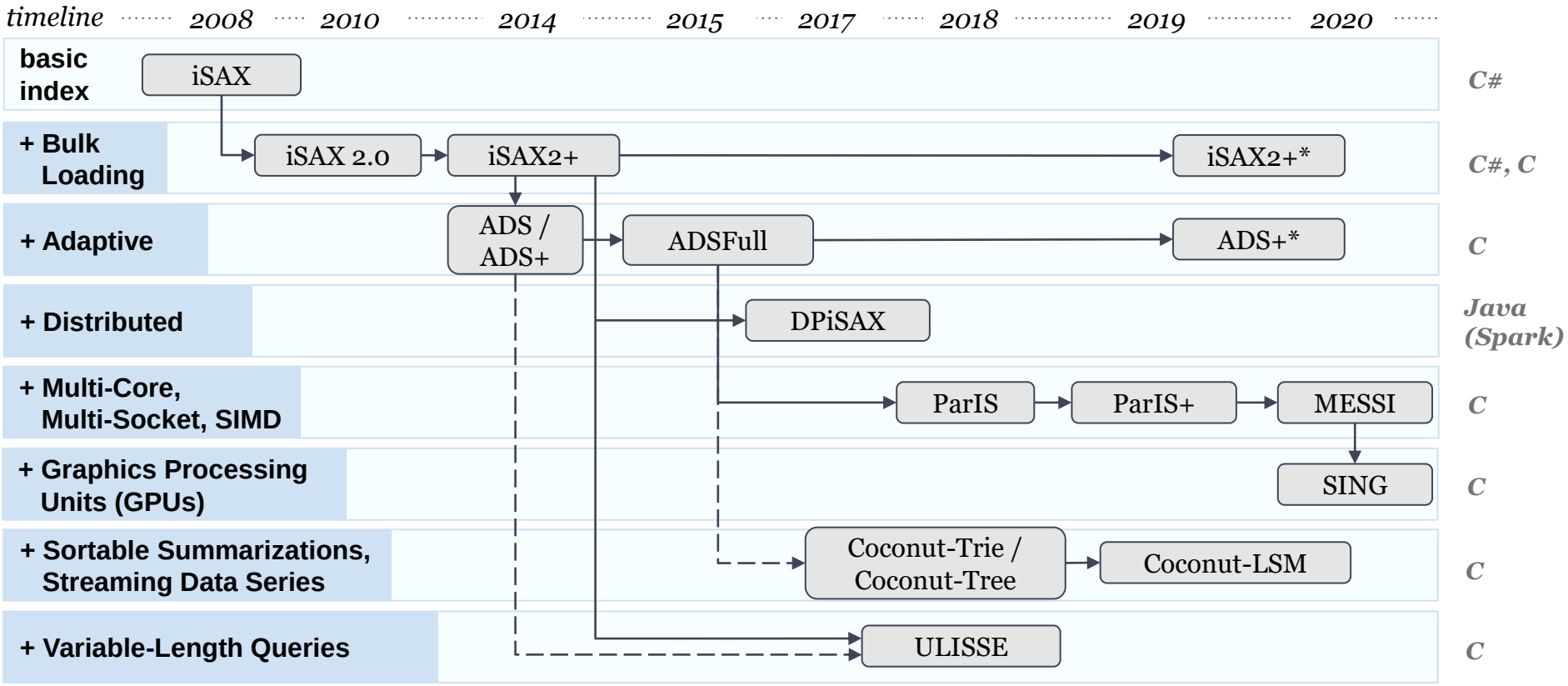
- Yagoubi-ICDM'17
- Yagoubi-TKDE'18
- Lavchenko-KAIS'20

- solution for distributed processing (Spark)
  - balances work of different worker nodes
    - partitions series into uniform groups with parallel sampling (for load balancing)
    - creates in parallel an index for each group (in a different node)
  - speeds-up query answering
    - exact queries are answered by all nodes (parallelize query execution)
    - approximate queries answered only by a single node (parallelize workload execution)

*i*SAX Index Family

Publications

Palpanas-ISIP'19



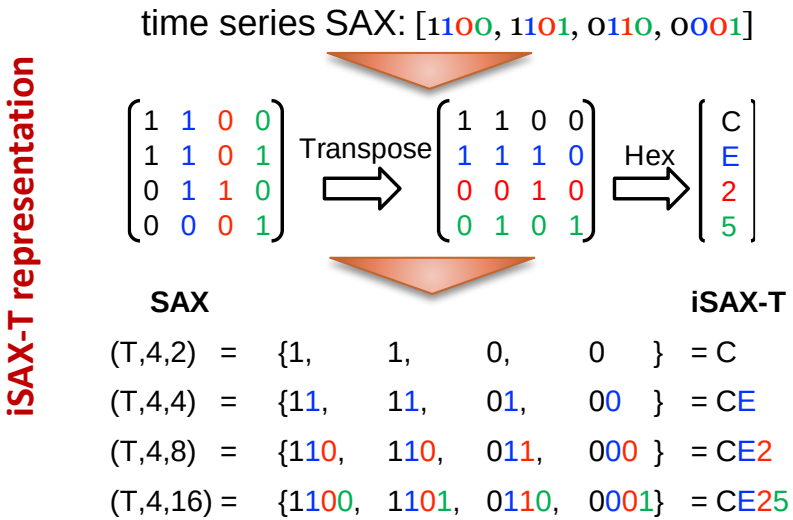
Timeline depicted on top; implementation languages marked on the right. Solid arrows denote inheritance of index design; dashed arrows denote inheritance of some of the design features; two new versions of iSAX2+/ADS+ marked with asterisk support approximate similarity search with deterministic and probabilistic quality guarantees.

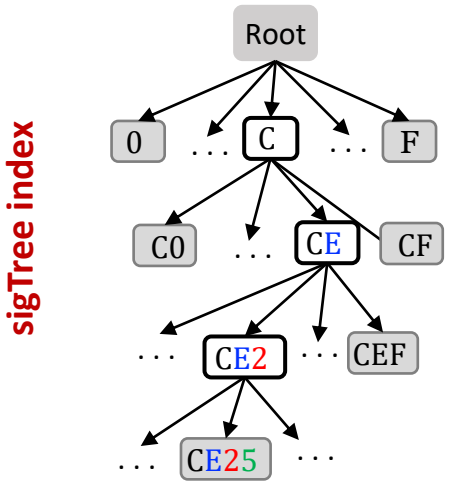
TARDIS

Publications

Zhang et al. ICDE'19

- solution for distributed processing (Spark)
  - based on iSAX-T representation and sigTree index
    - iSAX Transposition: transposes matrix of iSAX words of same cardinality, represents as strings
    - sigTree: prefix k-ary tree on iSAX-T strings





TARDIS

Publications

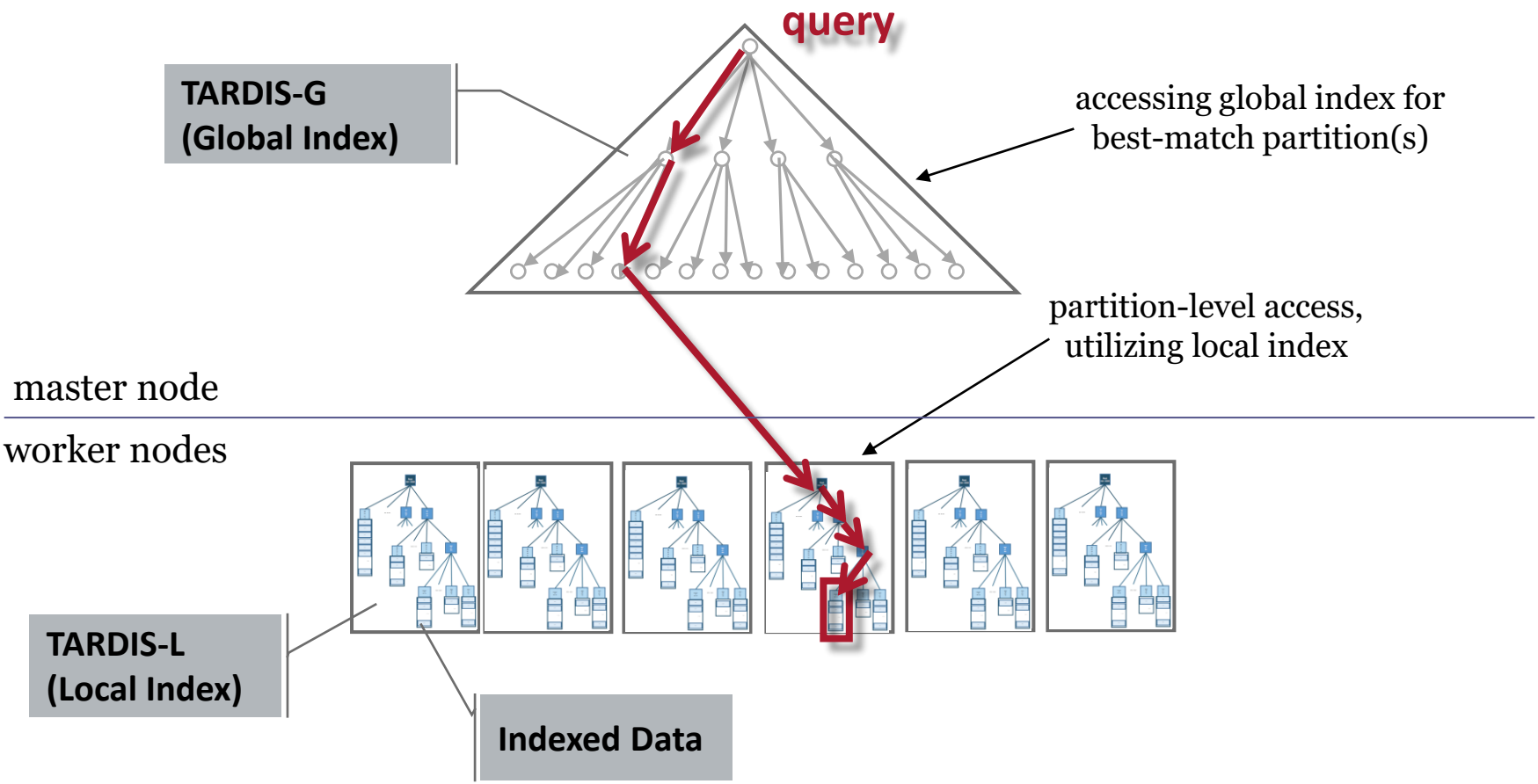
Zhang et al. ICDE'19

- solution for distributed processing (Spark)
  - based on iSAX-T representation and sigTree index
    - iSAX Transposition: transposes matrix of iSAX words of same cardinality, represents as strings
    - sigTree: prefix k-ary tree on iSAX-T strings
  - centralized global sigTree + distributed local sigTrees with raw data
    - global sigTree
      - constructed using statistics from local samples
      - serves as partition scheme for data re-distribution

TARDIS

Publications

Zhang et al. ICDE'19



TARDIS

Publications

Zhang et al. ICDE'19

- solution for distributed processing (Spark)
  - based on iSAX-T representation and sigTree index
    - iSAX Transposition: transposes matrix of iSAX words of same cardinality, represents as strings
    - sigTree: prefix k-ary tree on iSAX-T strings
  - centralized global sigTree + distributed local sigTrees with raw data
    - global sigTree
      - constructed using statistics from local samples
      - serves as partition scheme for data re-distribution
  - query answering
    - ng-approximate k-NN queries
    - exact-match queries (does the query appear exactly the same in the dataset?)

KV-match

Publications

Wu et al. ICDE'19

- solution for distributed (HDFS) subsequence similarity search
  - similarity search problem
    - subsequence similarity search: search for a short query inside a long series
    - ε-range queries
    - exact answers for constrained ε-range queries (using cNSM)
  - cNSM: constrained Normalized Subsequence Matching
    - essentially, constrained similarity search
    - intuitively, Z-normalization with constraints on degrees of amplitude scaling and offset shifting ($\alpha \geq 1$ and $\beta \geq 0$, respectively)

$$D(\hat{S}, \hat{Q}) \leq \varepsilon \quad \cap \quad \frac{1}{\alpha} \leq \frac{\sigma^S}{\sigma^Q} \leq \alpha \quad \cap \quad -\beta \leq \mu^S - \mu^Q \leq \beta$$

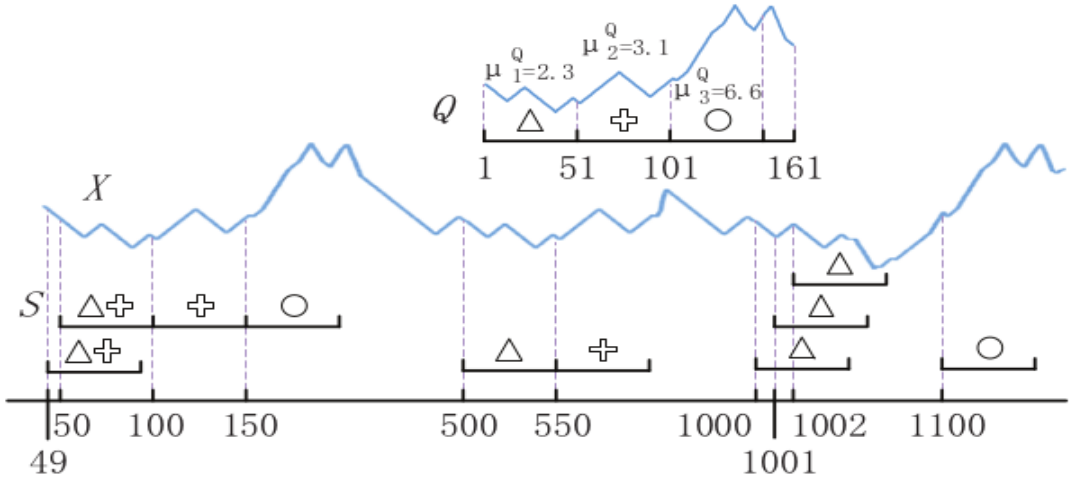
    - users control extent of amplitude scaling and offset shifting
    - normalized subsequence matching is a special case of cNSM

KV-match

Publications

Wu et al. ICDE'19

- solution for distributed (HDFS) subsequence similarity search
  - index creation
    - slide window on input series
    - produce ordered rows of key-value pairs
      - key K_i : a range of mean values, $K_i = [LR_i , UR_i)$
      - value V_i : the set of sliding windows whose mean values fall within K_i



Index

	μ	Window
△	1.7	1000
	1.8	1001
	1.9	1002
✚	2.4	500
	2.6	50
	2.8	49
	3.0	100
	3.3	550
	...	...
○	6.1	150
	7.2	1100

Key	Value
[1. 5, 2. 0)	[1000, 1002]
[2. 0, 3. 0)	[49, 50], [500, 500]
[3. 0, 4. 0)	[100, 100], [550, 550]
.....	.....
[6. 0, 7. 5)	[150, 150], [1100, 1100]

    - key-value table stored in HBase

KV-match

Publications

Wu et al. ICDE'19

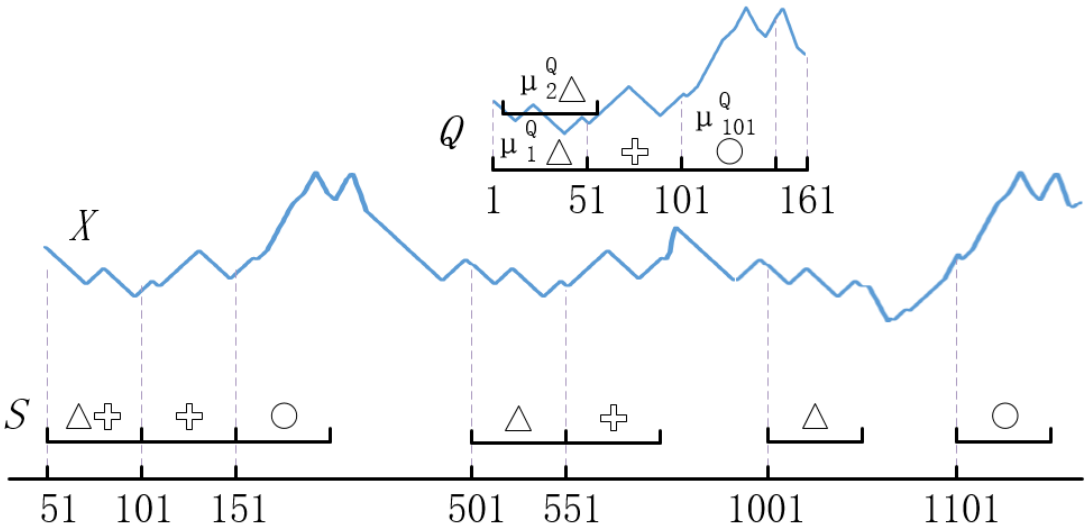
- solution for distributed (HDFS) subsequence similarity search
  - query answering
    - for query Q and corresponding subsequence S
      - segment Q into aligned length-w disjoint windows (requires having several indexes of different lengths)
    - for each window Q_i and S_i
      - filtering condition: S is candidate answer only if all μ_{Si} fall within $[LR_i , UR_i]$
    - Phase 1: **Index-probing**
      - generate set of candidate subsequences CS
    - Phase 2: **Post-processing**
      - verify subsequences in CS by computing actual distance on the raw data

L-match

Publications

Feng et al. IEEE Access'20

- solution for distributed (HDFS) subsequence similarity search
  - L-match improves on KV-match
    - instead of sliding a window to build the index, L-match slides a window on query
    - index is more compact
    - operations are naturally parallelizable (no data-window overlaps among nodes)



Index

μ	Offset
1.8	1001
2.4	501
2.6	51
3	101
6.1	151
7.2	1101

L-match

Publications

Feng et al.
IEEE Access'20

- solution for distributed (HDFS) subsequence similarity search
  - L-match improves on KV-match
    - instead of sliding a window to build the index, L-match slides a window on query
    - index is more compact
    - operations are naturally parallelizable (no data-window overlaps among nodes)
  - compared to KV-match, L-match is slightly slower, but 10x smaller

ParSketch

Publications

- Cole et al. KDD'05
- Yagoubi et al. DMKD'18

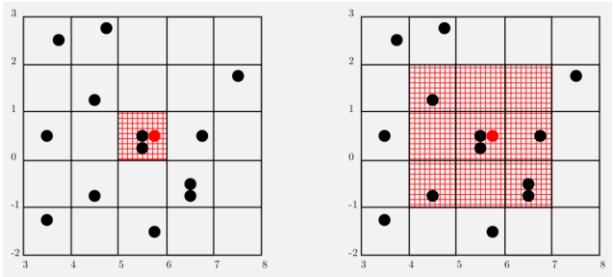
- solution for distributed processing (Spark)
  - represents data series using sketches
    - using a set of random vectors (Johnson-Lindenstrauss lemma)

$$x = (x_1, x_2, x_3, \ldots x_n)$$
$$y = (y_1, y_2, y_3, \ldots y_n)$$
$$z = (z_1, z_2, z_3, \ldots z_n)$$

$$R1 = (r_1 1, r_1 2, r_1 3, \ldots r_1 w)$$
$$R2 = (r_2 1, r_2 2, r_2 3, \ldots r_2 w)$$
$$R3 = (r_3 1, r_3 2, r_3 3, \ldots r_3 w)$$
$$R4 = (r_4 1, r_4 2, r_4 3, \ldots r_4 w)$$

$$(xsk1, xsk2, xsk3, xsk4)$$
$$(ysk1, ysk2, ysk3, ysk4)$$
$$(zsk1, zsk2, zsk3, zsk4)$$

  - define groups of dimensions in sketches
  - store the values of each group in a grid (in parallel)
    - each grid is kept by a node



node 1 node 2

  - for ng-approximate query answering (originally proposed for ε-range queries)
    - find in the grids time series that are close to the query
    - finally, check the real similarity of candidates to find the results
  - performs well for high-frequency series

Questions?

Experimental Comparison: Exact Query Answering Methods

How do these methods compare?

- several methods proposed in last 3 decades
- never carefully compared to one another

- we now present results of extensive experimental comparison

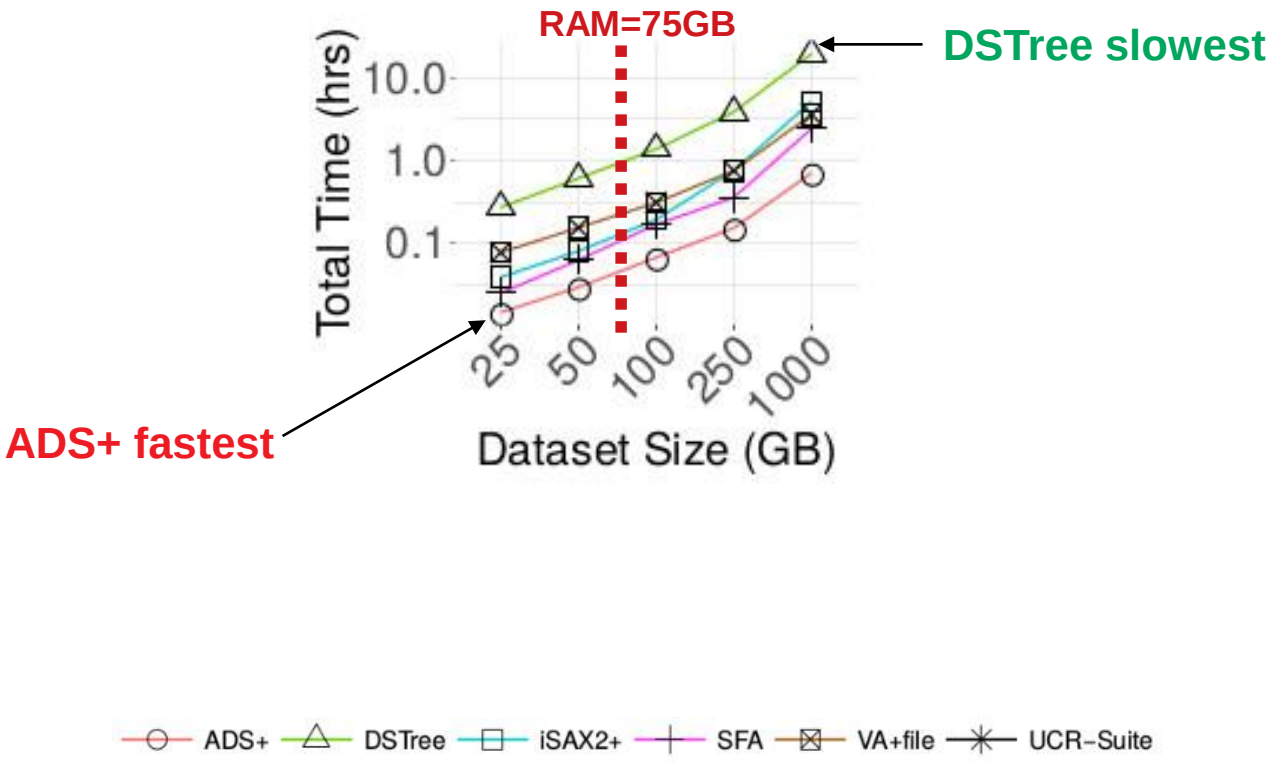
Experimental Framework

Publications

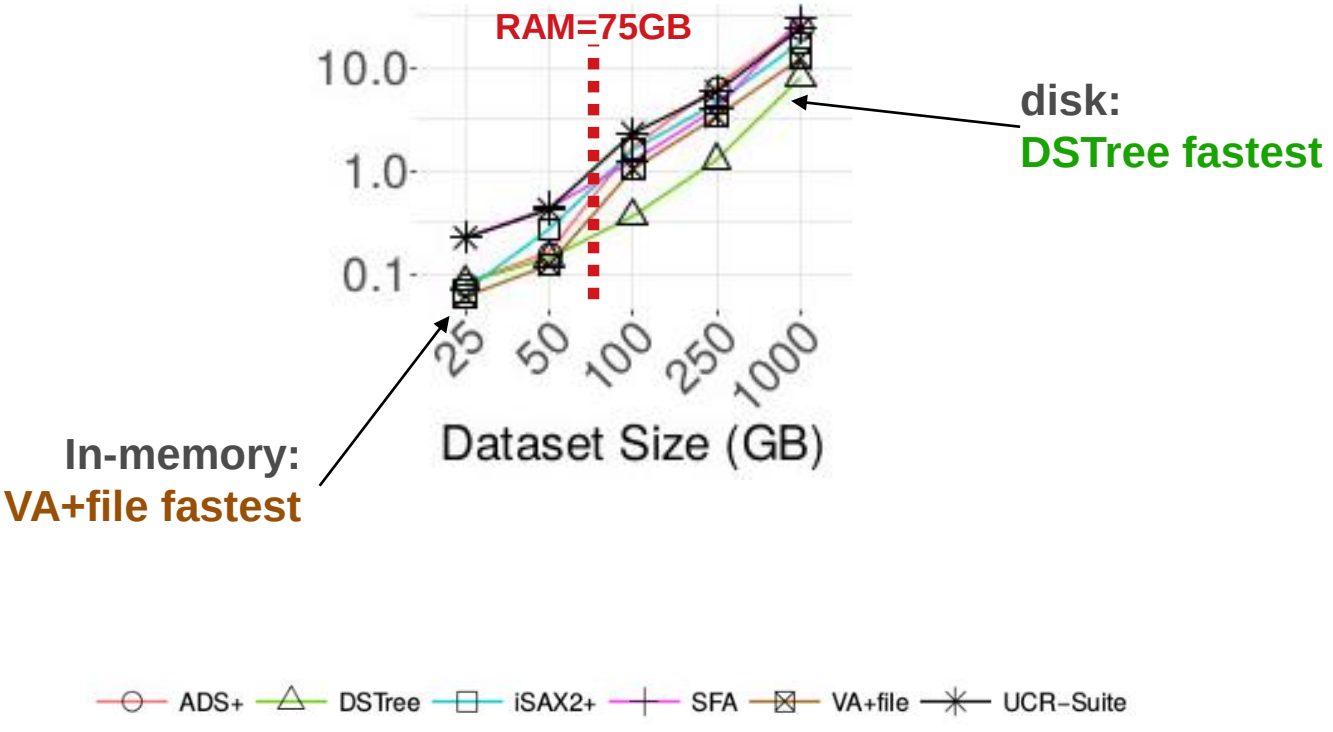
Echihabi-PVLDB'18

- Hardware
  - HDD and SSD
- Datasets
  - Synthetic (25GB to 1TB) and 4 real (100 GB)
- Exact Query Workloads
  - 100 – 10,000 queries
- Performance measures
  - Time, #disk accesses, footprint, pruning, Tightness of Lower Bound (TLB), etc.
- C/C++ methods (4 methods reimplemented from scratch)
- Procedure:
  - Step 1: Parametrization
  - Step 2: Evaluation of individual methods
  - Step 3: Comparison of best methods

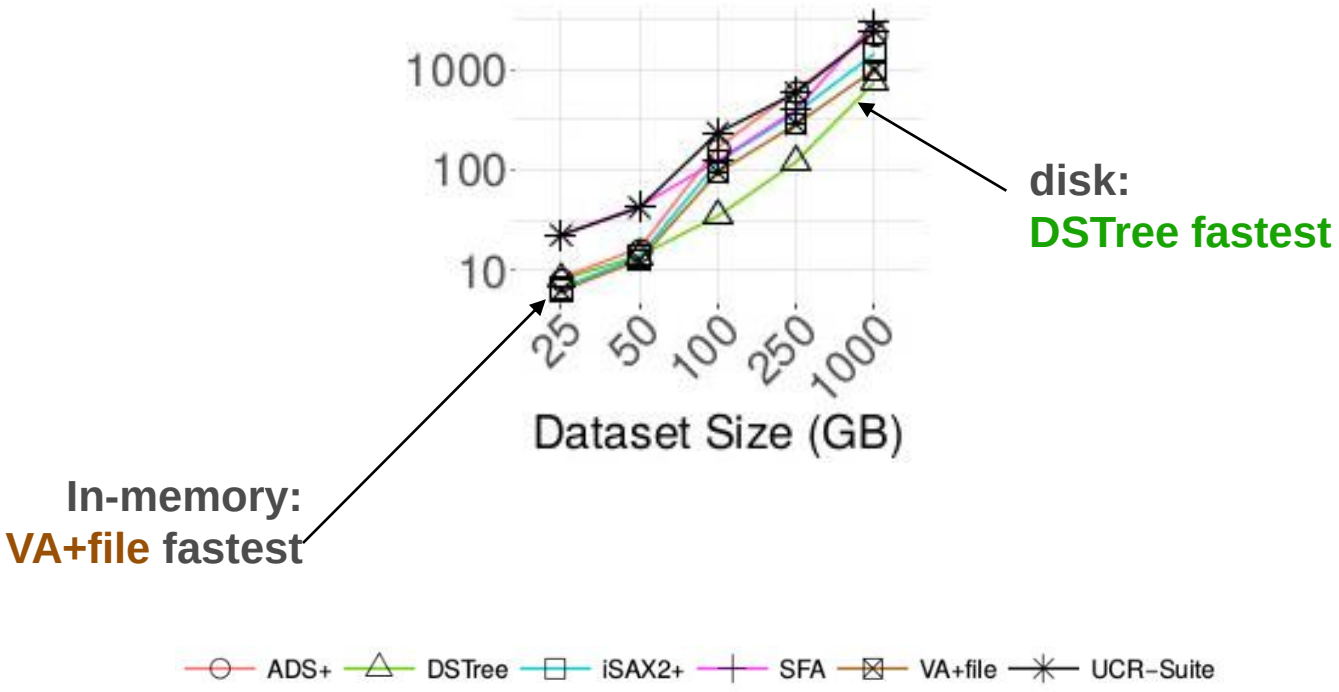
Time for Indexing (Idx) vs. Dataset Size



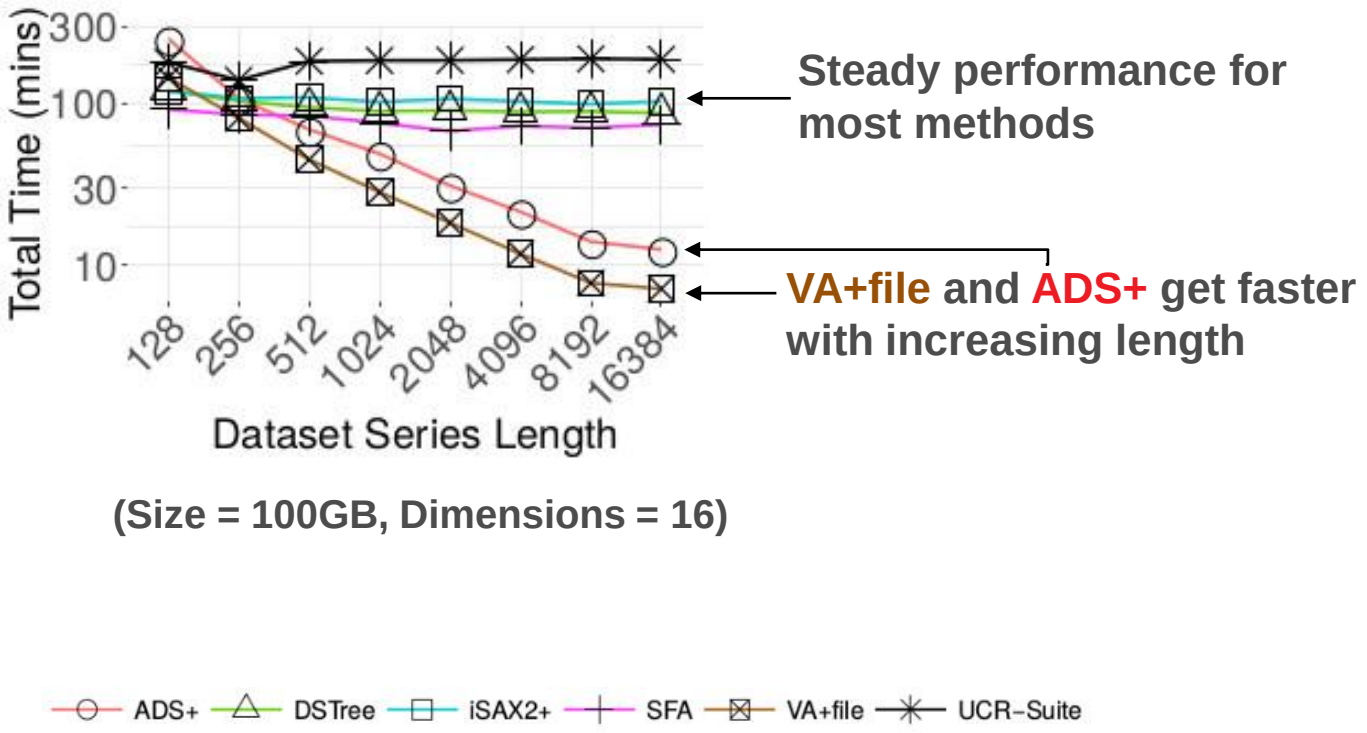
Time for 100 Exact Queries vs. Dataset size



Time for Idx + 10K Exact Queries vs. Dataset size



Time for Idx + 10K Exact Queries vs. Series Length



Publications

Echihabi-PVLDB'18

Unexpected Results

- Some methods do not scale as expected (or not at all!)
- Brought back to the spotlight two older methods VA+file and DSTree
  - Our reimplementations outperform by far the original ones
- Optimal parameters for some methods are different from the ones reported in the original papers
- Tightness of Lower Bound (TLB) does not always predict performance

Publications

Echihabi-PVLDB'18

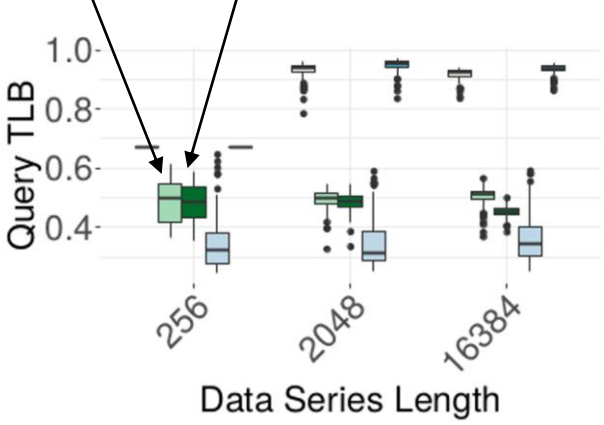
TLB does not always predict performance

The TLB measures the quality of a summarization (higher is better)

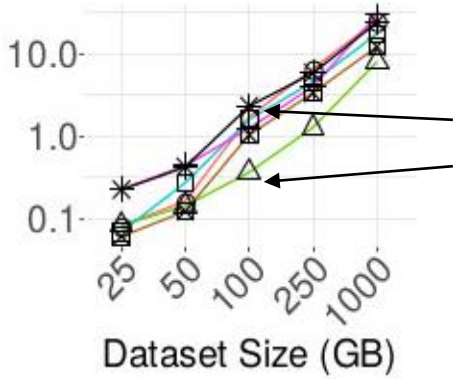
$$\mathbf{0} \leq \text{TLB} = \frac{dist(Query, candidate)\ in\ reduced\ space}{dist(Query, candidate)\ in\ original\ space} \leq \mathbf{1}$$

worst — best

DSTree and iSAX2+ have similar TLB



YET



iSAX2+ 5x slower than DSTree

No bias, same data and same implementation framework

Publications

Echihabi-PVLDB'18

Insights

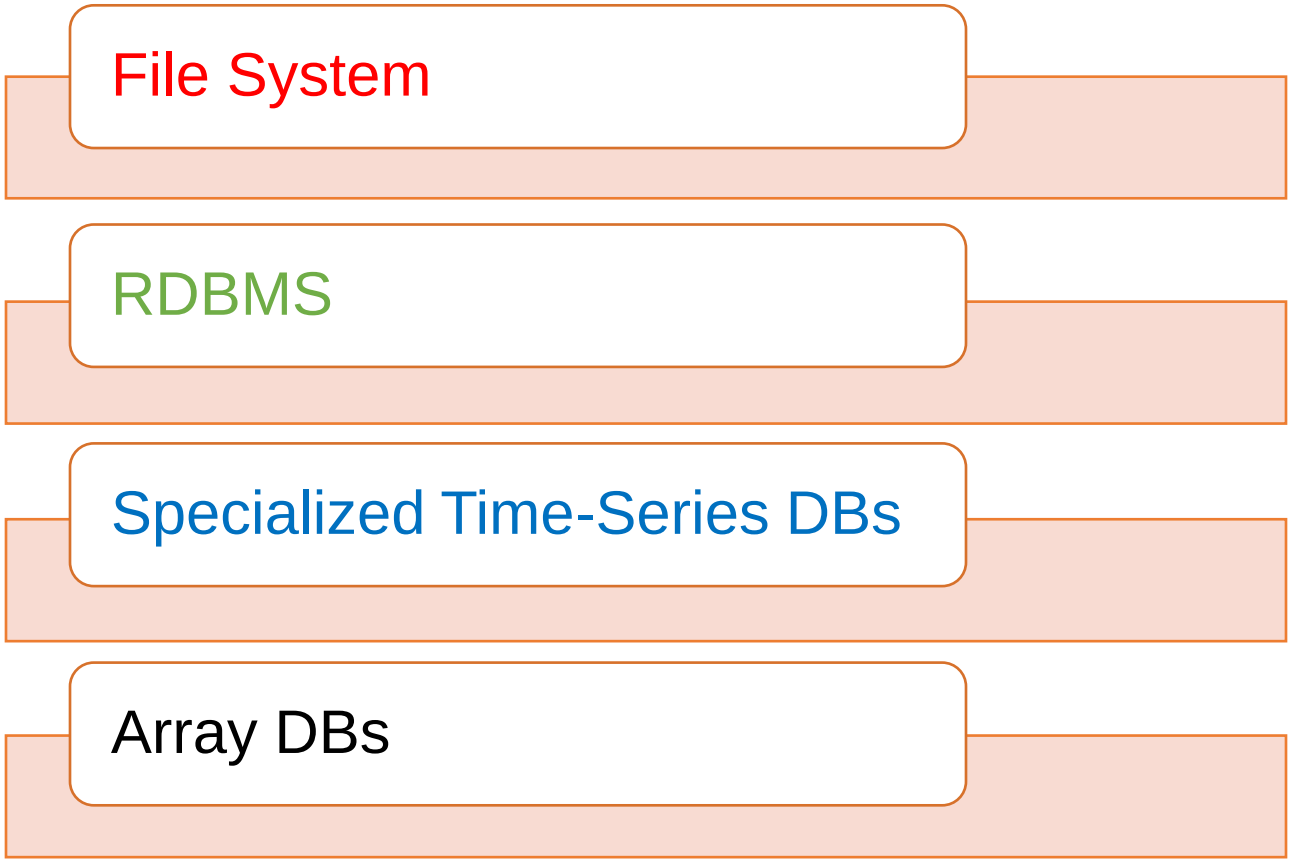
- Results are sensitive to:
  - Parameter tuning
  - Hardware setup
  - Implementation
  - Workload selection
- Results identify methods that would benefit from modern hardware

Questions?

Time Series Management Systems

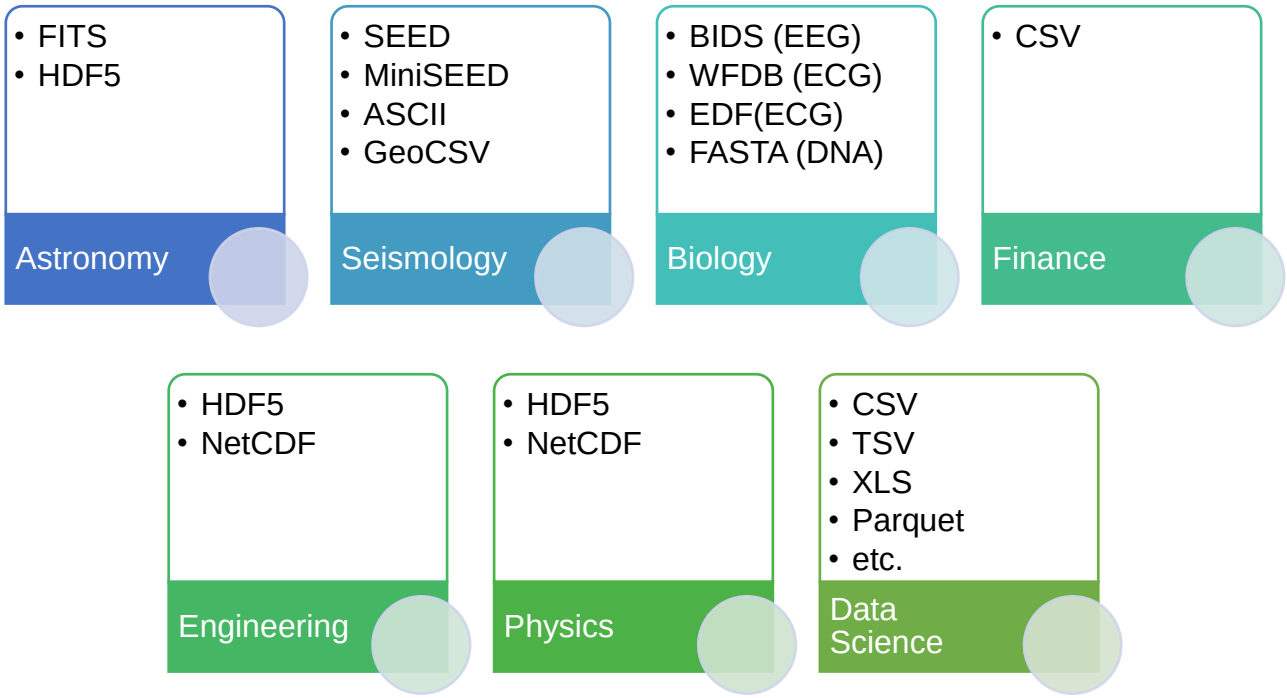
Storing Time-Series

Multiple options. By popularity:



Storing Time-Series: **File-System**

Multiple **different formats** implemented for **various applications**



Storing Time-Series: DBMS

Illustra (1993) → IBM Informix (Time-Series DataBlade):

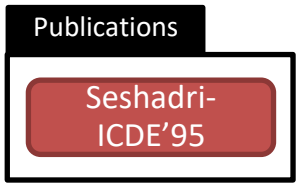
Commercial System

- Users need to define a time-series sub-type, which have a datetime as the first column in the definition
- Can encode both regular and irregular time-series (fixed of variable intervals)
- Can describe meta-data
- Supports: running aggregates, prev, next value reasoning, horizontal and vertical mathematical operations, lags, etc.

Shore → SEQ

Academic System

- Custom Time-Series Data Type
- Various time-series operators (order, correlation, etc.)



Oracle:

Commercial System

- Introduced Time-Series functionality in Oracle8
- Now merged into the main product.
- It is in the form of time-series analytics functions (e.g., forecasting)

Storing Time-Series: DBMS

Illustra (1993) → IBM Inform… …es DataBlade): Commercial System

- Users need to define a… …which have a datetime as the first column in the d…
- Can encode … …d of variable interv…
- Ca…
- Supp… …zontal and vertica…

Most people use DBMSs merely for storing and retrieving time-series.

All analysis is performed externally.

Shore → SE… Academic System

- Custom Tir…
- Various time…

Oracle: Commercial System

- Introduced Ti… …ality…
- Now merged into the main product.
- It is in the form of time-series analytics functions (e.g., forecasting)

Storing Time-Series: **Specialized Time-Series DBs**

InfluxDB
- Storage: Custom (TSM-Tree)

TimeScaleDB
- Storage: PostgreSQL

Beringei
- Storage: Compressed Arrays on Disk

Druid
- Metadata Storage: DBMS
- Data Storage: HDFS, S3, Custom format

Prometheus
- Storage: Custom (TSDB Format)

CrateDB
- Storage: Custom (Column-oriented)

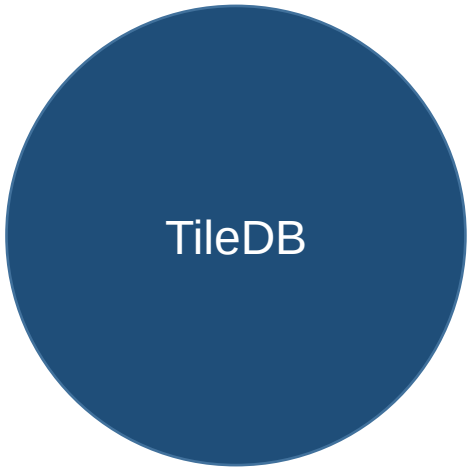
IoTDB
- Storage: Custom: (TsFile – compression + stats)

OpenTSDB
- Storage: HBase

QuasarDB
- Storage: RocksDB

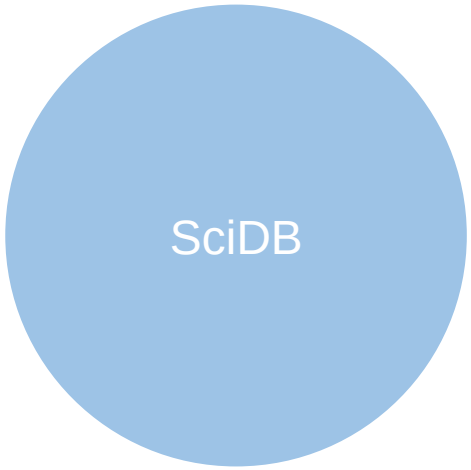
Amazon TimeStream
- Storage: Unknown

Storing Time-Series: **ArrayDBs**



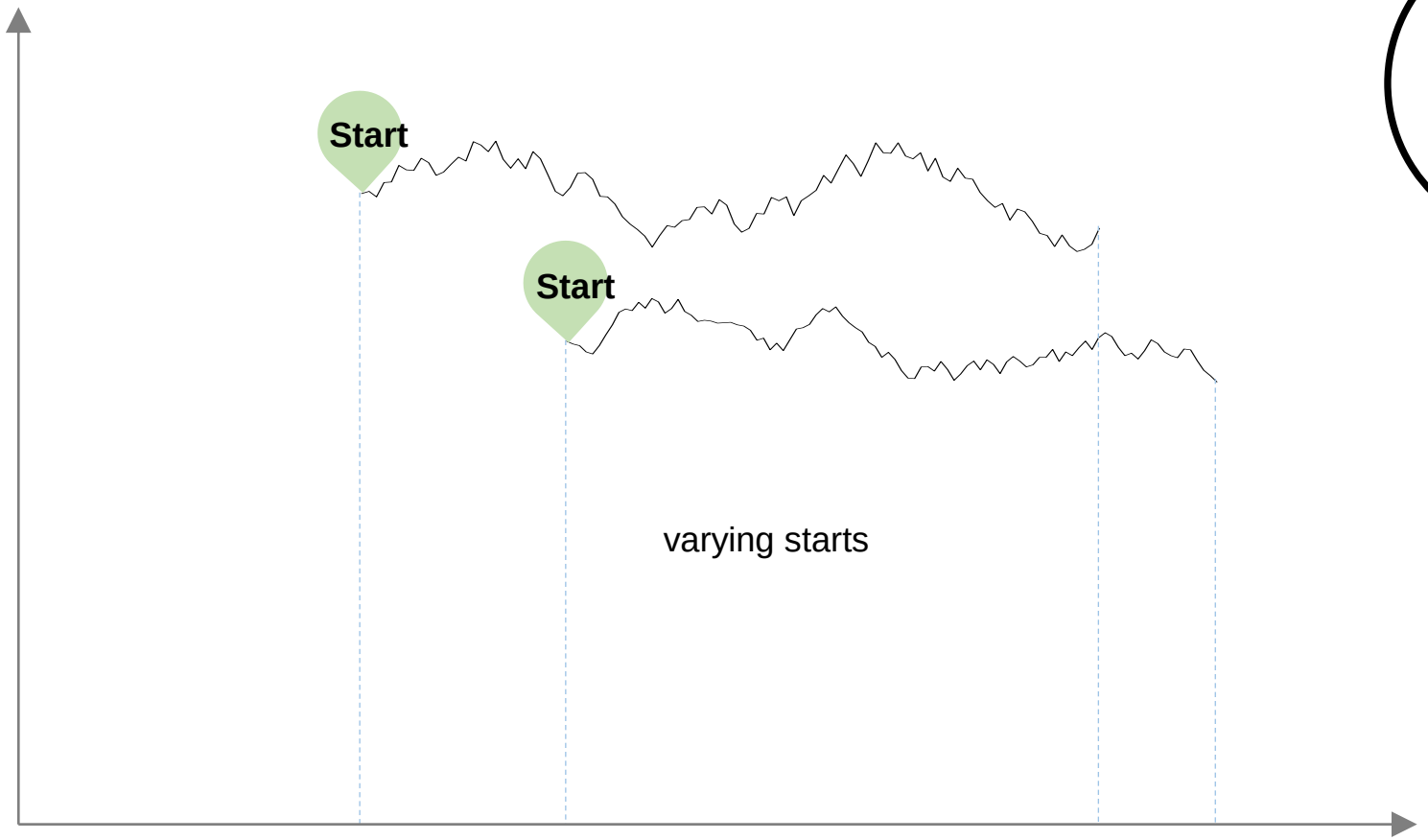
Custom Log-Structured storage

Sits on top of existing DBMSs



Custom storage

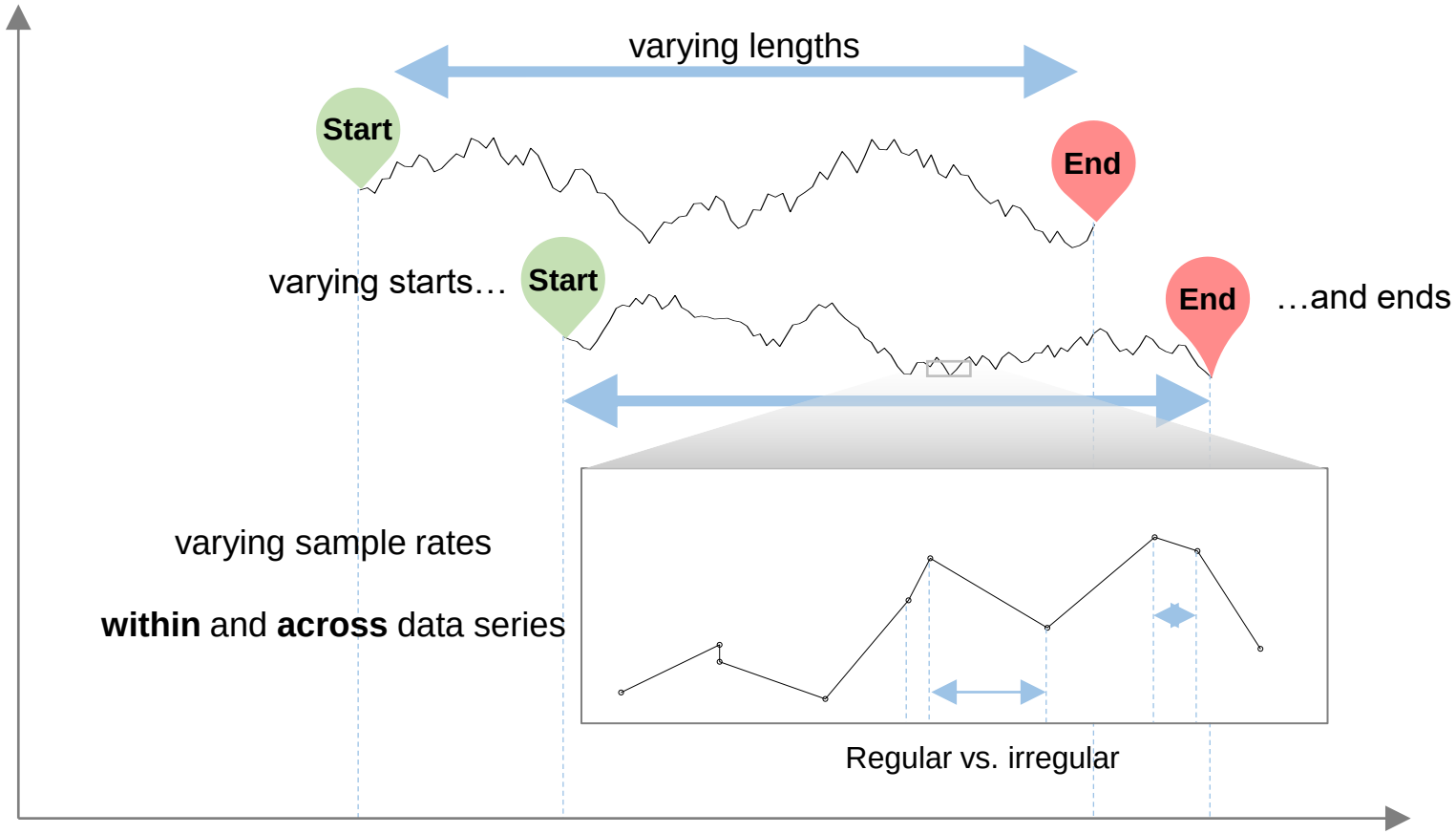
Time-Series Characteristics



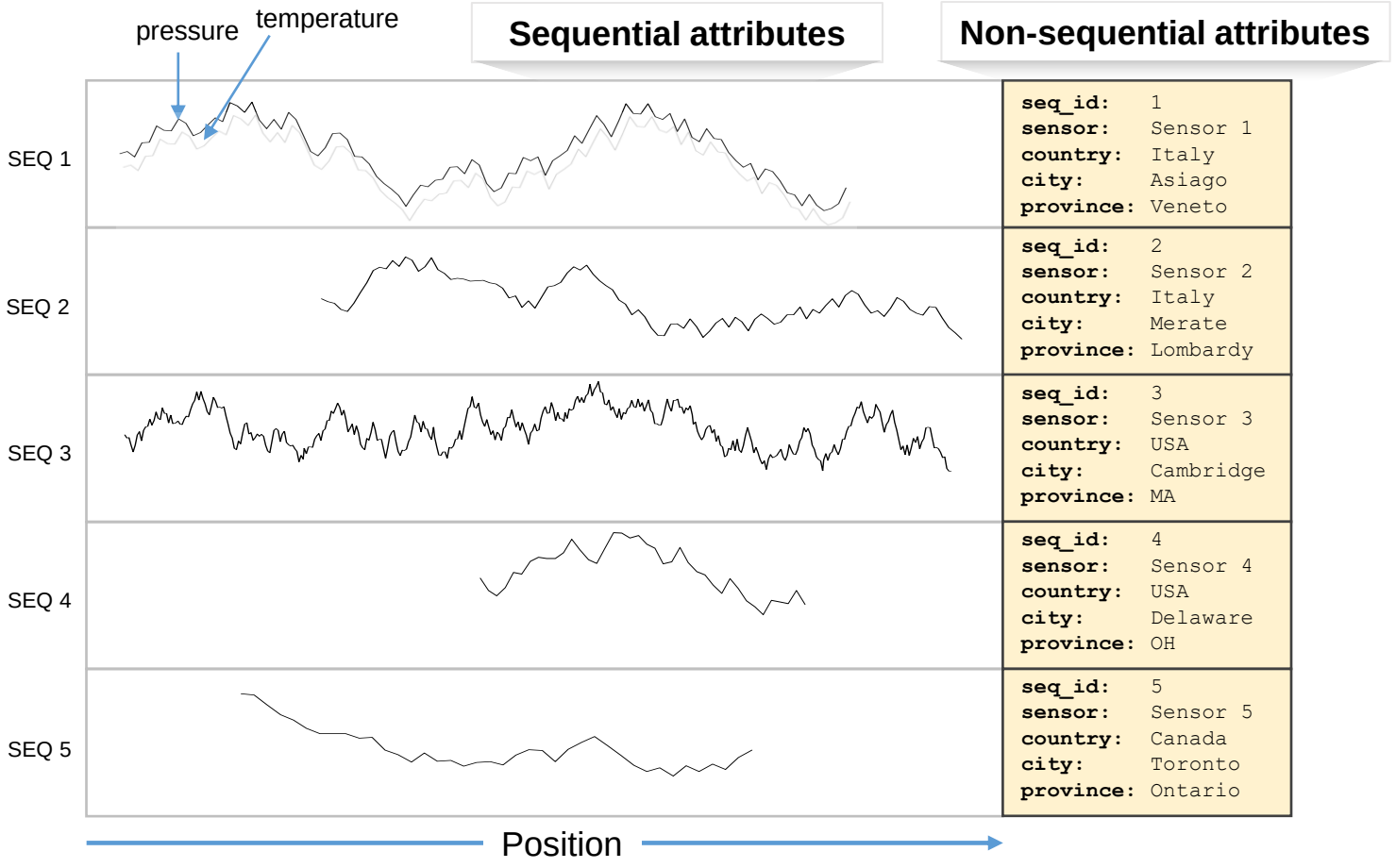
The Data-Type Characteristics

What are the properties of data series?

Time-Series Characteristics

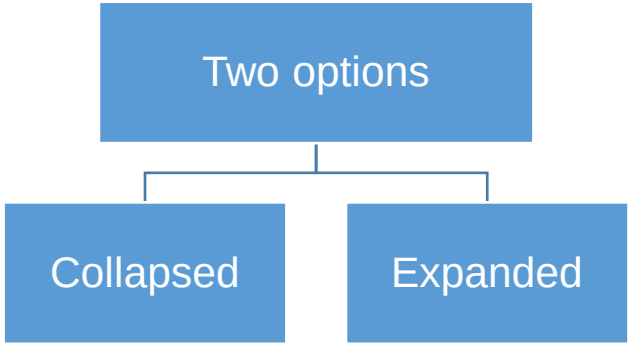


Time-Series Characteristics



Storage

sensor_id	country	city	province	temperature	pressure
102	Italy	Asiago	Veneto		
104	Italy	Merate	Lombardy		
201	USA	Cambridge	MA		
202	USA	Delaware	OH		
303	Canada	Toronto	Ontario		



Storage: **Collapsed** Design

Time-series-as-a-record

sensor_id	country	city	province	temperature	pressure
102	Italy	Asiago	Veneto		
104	Italy	Merate	Lombardy		
201	USA	Cambridge	MA		
202	USA	Delaware	OH		
303	Canada	Toronto	Ontario		

UW–MADISON COMPUTER SCIENCES
SEQ Project

PREDATOR
ENHANCED DATA TYPE OBJECT-RELATIONAL DBMS

ILLUSTRA

Informix SOFTWARE

IBM

nestor

Our own experimental system, currently under development at the **University of Paris**

Université de Paris

Specialized Data Type

- With specialized UDFs
- Compression algos
- Stored **inline** or in **containers**
  e.g. in Informix small time series (<1500 bytes) are stored in-row

UW–MADISON COMPUTER SCIENCES
SEQ Project

PREDATOR
ENHANCED DATA TYPE OBJECT-RELATIONAL DBMS

sensor_id	country	city	province	temperature	pressure
102	Italy	Asiago	Veneto	1 **Shore Large Object** per series (contains 1 **Sparse Array** per series)	
104	Italy	Merate	Lombardy		
….					

Informix SOFTWARE

IBM

sensor_id	country	city	province	temperature	pressure
102	Italy	Asiago	Veneto	inline up to 1500 bytes	
104	Italy	Merate	Lombardy	>1500 bytes → ref to external container (each time series is sorted by time)	
….					

Storage: **Expanded** Design

Time-series-point-as-a-record

sensor_id	country	city	province	timestamp	temperature	pressure
102	Italy	Asiago	Veneto	1616264642	26	1013.5
102	Italy	Asiago	Veneto	1616268642	26.5	1013.1
…	…	…	…	…	…	…
104	Italy	Merate	Lombardy	1616264682	24	1012.2
…	…	…	…	…	…	…
201	USA	Cambridge	MA	1616274642	22	1011.6
202	USA	Delaware	OH	1616294642	21	1005.8
303	Canada	Toronto	Ontario	1616274642	18	1008.2

teradata.

```
CREATE TABLE
   buoyinfo( buoyid sid, salinity integer,
             temperature integer )
PRIMARY TIME INDEX
  (TIMESTAMP(1), DATE '2016-04-19',
   HOURS(1), COLUMNS(buoyid, salinity),
   SEQUENCED);
```

Timescale

```
SELECT create_hypertable('table', 'time');
```

Special Table

- Index and partition by timestamp
- Tags are compressed (and sometimes bitmap indexed)

Storage: **Expanded** Design

Time-series-point-as-a-record

sensor_id	country	city	province	timestamp	temperature	pressure
102	**Usually also encoded with inverted indexes** e.g. in druid, bitmaps CountryItalyBmp = […] CountryUSABmp = […] CityAsiagoBmp = […] ….			1616264642	26	1013.5
102				1616268642	26.5	1013.1
…				…	…	…
104				1616264682	24	1012.2
…				…	…	…
201				1616274642	22	1011.6
202				1616294642	21	1005.8
303				1616274642	18	1008.2

teradata.

```
CREATE TABLE
   buoyinfo( buoyid sid, salinity integer,
             temperature integer )
PRIMARY TIME INDEX
  (TIMESTAMP(1), DATE '2016-04-19',
   HOURS(1), COLUMNS(buoyid, salinity),
   SEQUENCED);

SELECT create_hypertable('table', 'time');
```

Timescale

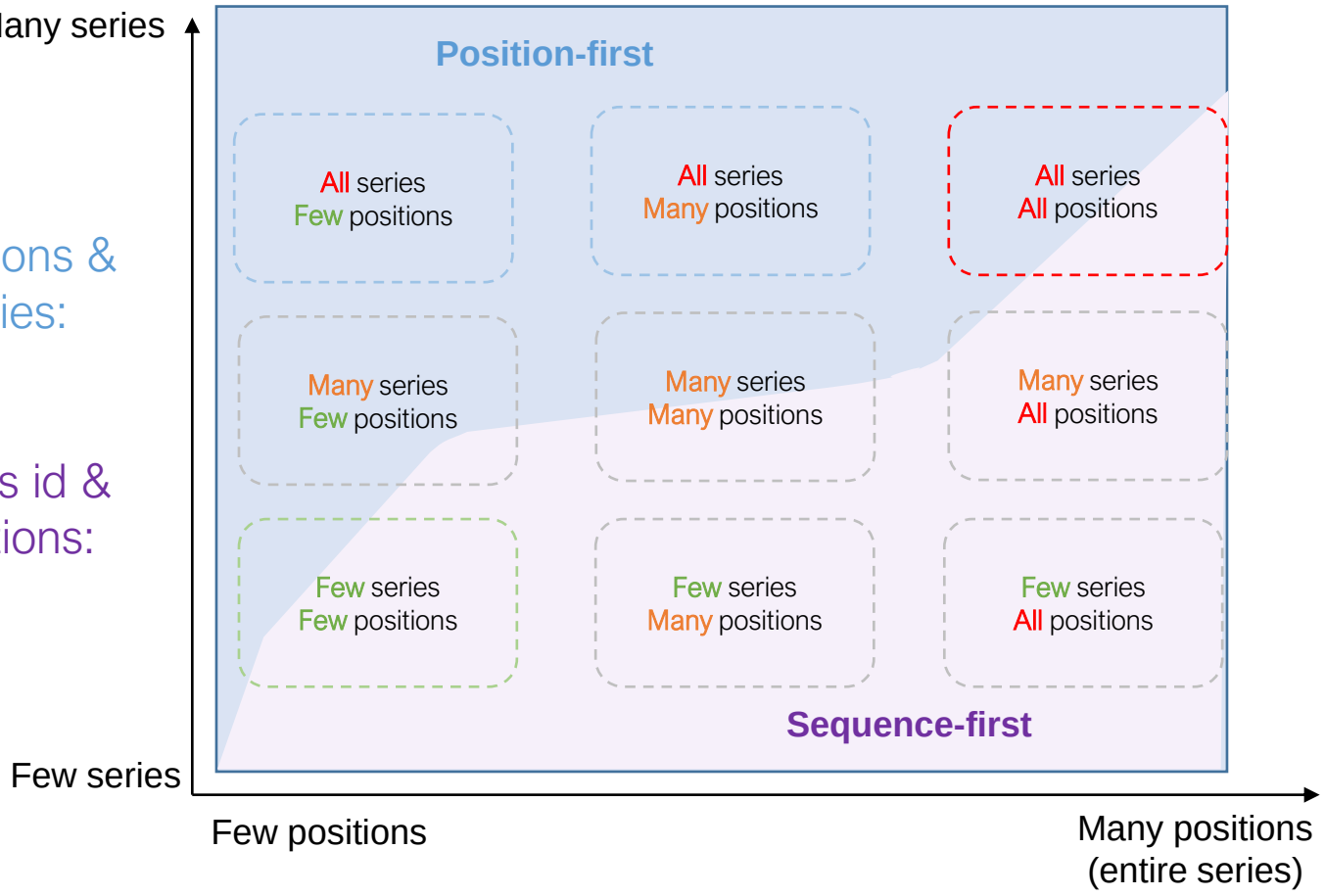
Special Table

- Index and partition by timestamp
- Tags are compressed (and sometimes bitmap indexed)

Expanded Design: Clustering

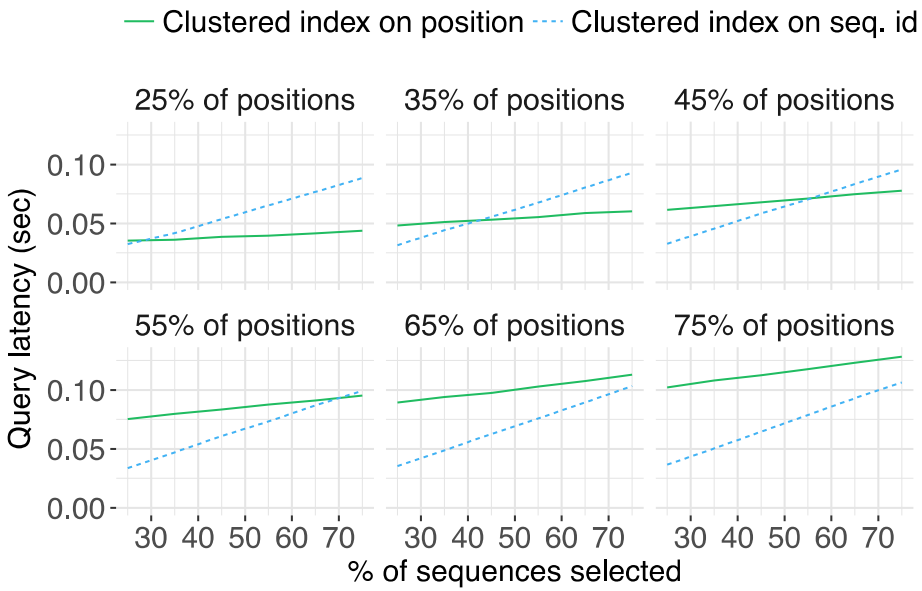
Heavy filtering on positions & Accessing lots of series: **position-first**

Heavy filtering on series id & accessing lots of positions: **sequence-first**



Heavy filtering on positions & Accessing lots of series: **position-first**

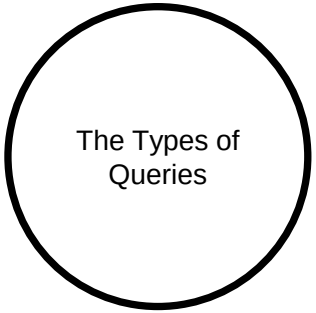
Heavy filtering on series id & accessing lots of positions: **sequence-first**



*DBMS-X


ed index on position
Clustered index on seq. id
itions
35% of positions
45% of positions
75% of positions
0 50 60 70
30 40 50 60 70
ences selected
*DBMS-X
Heavy filtering on positions &
Accessing
position
Heavy filtering on series id
accessing lots of positions:
sequence-first
Most existing systems
sort data by series

Query Types



Simple

Selection-Projection-Transformation

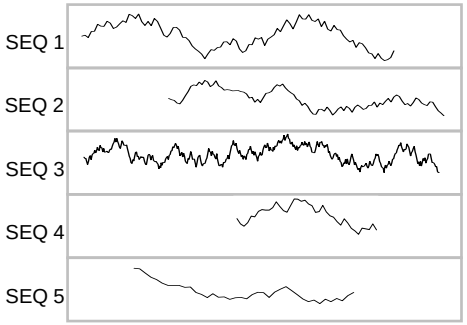
Complex

Analytical/Mining Queries

Query Types

Simple

Selection-Projection-Transformation



Query Types

Simple

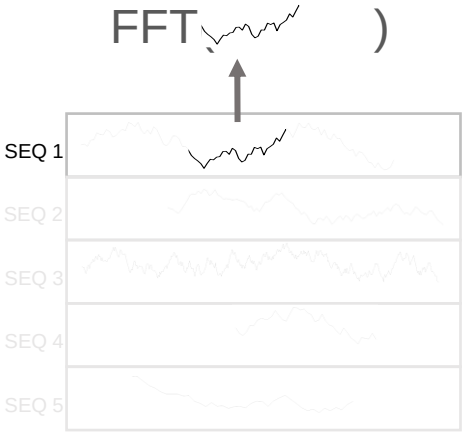
Selection-Projection-Transformation

Select some series → **Project** some points → **Apply** a **function** on them

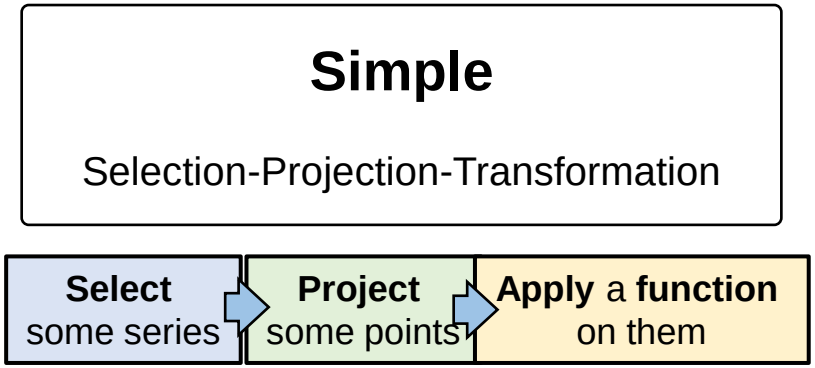
Query Type 1: Find all points of a subset of data series
*e.g., Bring me the **whole history** of "pressure" for "Sensor 1"*

Query Type 2: Look at the points at a subset of the positions
*e.g., Compute the **average** pressure for all sensors for the range of positions that cover the 2nd to the 12th of March.*

Query Type 3: Look at a subset of points based on a value
*e.g., Bring me **all pressure** values above a **threshold***



Query Types



Query Type 1: Find all points of a subset of data series
*e.g., Bring me the **whole history** of "pressure" for "Sensor 1"*

Query Type 2: Look at the points at a subset of the positions
*e.g., Compute the **average** pressure for all sensors for the range of positions that cover the 2nd to the 12th of March.*

Query Type 3: Look at a subset of points based on a value
*e.g., Bring me **all pressure** values above a **threshold***

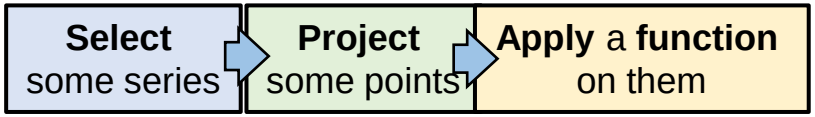
Query Types

Classic **1/n-dimensional indexes & layouts** for **point** and **range queries:**

Point get: Get seq id = 1
Range: Get positions 10 - 100

Simple

Selection-Projection-Transformation



Query Type 1: Find all points of a subset of data series
*e.g., Bring me the **whole history** of "pressure" for "Sensor 1"*

Query Type 2: Look at the points at a subset of the positions
*e.g., Compute the **average** pressure for all sensors for the range of positions that cover the 2nd to the 12th of March.*

Query Type 3: Look at a subset of points based on a value
*e.g., Bring me **all pressure** values above a **threshold***

Query Types

Simple

Selection-Projection-Transformation

Select some series → **Project** some points → **Apply** a **function** on them

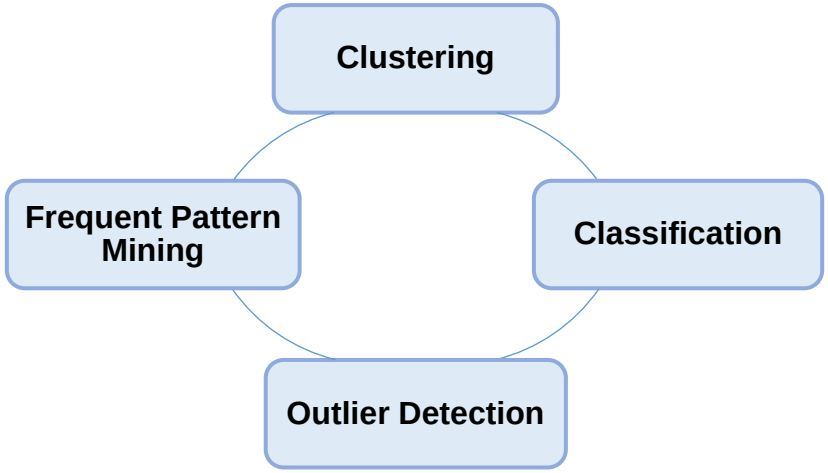
Query Type 1: Find all points of a subset of data series
*e.g., Bring me the **whole history** of "pressure" for "Sensor 1"*

Query Type 2: Look at the points at a subset of the positions
*e.g., Compute the **average** pressure for all sensors for the range of positions that cover the 2nd to the 12th of March.*

Query Type 3: Look at a subset of points based on a value
*e.g., Bring me **all pressure** values above a **threshold***

Complex

Analytical/Mining Queries



Query Types

Simple

Selection-Projection-Transformation

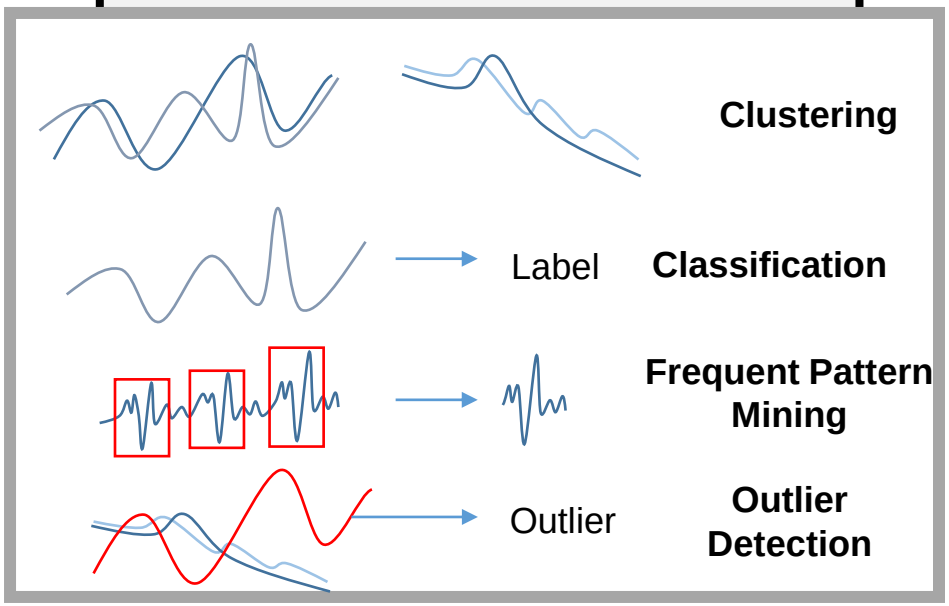
Select some series → **Project** some points → **Apply** a **function** on them

Query Type 1: Find all points of a subset of data series
*e.g., Bring me the **whole history** of "pressure" for "Sensor 1"*

Query Type 2: Look at the points at a subset of the positions
*e.g., Compute the **average** pressure for all sensors for the range of positions that cover the 2nd to the 12th of March.*

Query Type 3: Look at a subset of points based on a value
*e.g., Bring me **all pressure** values above a **threshold***

Complex



Query Types

Simple

Selection-Projection-Transformation

Select some series → **Project** some points → **Apply** a **function** on them

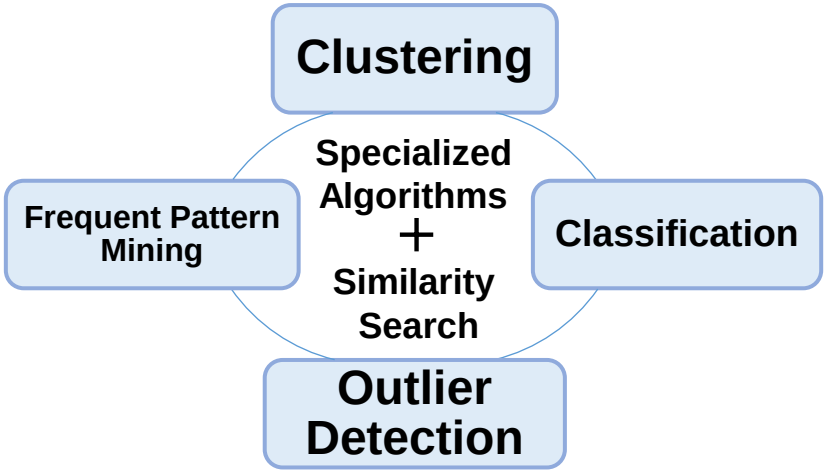
Query Type 1: Find all points of a subset of data series
*e.g., Bring me the **whole history** of "pressure" for "Sensor 1"*

Query Type 2: Look at the points at a subset of the positions
*e.g., Compute the **average** pressure for all sensors for the range of positions that cover the 2nd to the 12th of March.*

Query Type 3: Look at a subset of points based on a value
*e.g., Bring me **all pressure** values above a **threshold***

Complex

Analytical/Mining Queries



Query Types

Simple

Selection-Projection-Transformation

Select some series → **Project** some points → **Apply** a **function** on them

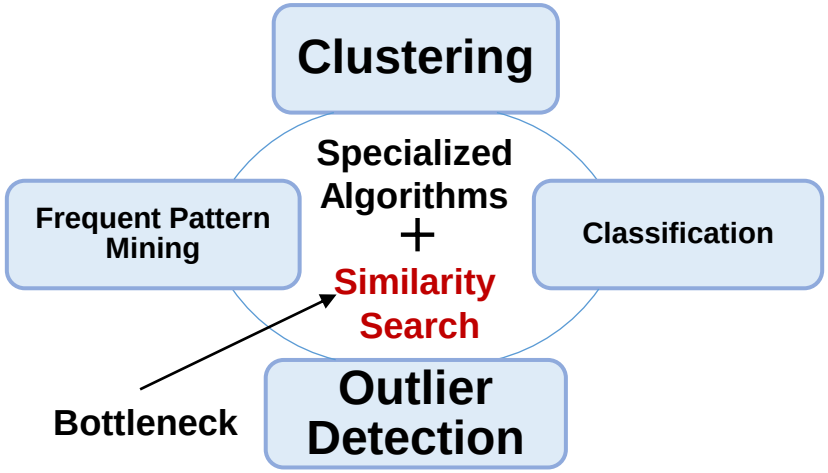
Query Type 1: Find all points of a subset of data series
*e.g., Bring me the **whole history** of "pressure" for "Sensor 1"*

Query Type 2: Look at the points at a subset of the positions
*e.g., Compute the **average** pressure for all sensors for the range of positions that cover the 2nd to the 12th of March.*

Query Type 3: Look at a subset of points based on a value
*e.g., Bring me **all pressure** values above a **threshold***

Complex

Analytical/Mining Queries



Time-Series Management Systems

a few more details on the popular systems:

- InfluxDB
- TimescaleDB

InfluxDB

- Storage Engine:
  - **Log Structured Merge Tree: LSM-Tree** variant that expects data to arrive ordered by time and partitions them by distinct sequence. It then stores each series contiguously.
- Schema:
  - Tags and fields. Tags are used to describe meta-data and fields are used to store quantities that change over time.
- Queries
  - It supports group by (only on tags), join (on timestamps and fields), selections, projections, and aggregations.
  - It also supports continuous queries

TimescaleDB

- **Storage:** Uses PostgreSQL as the backend.
  - It partitions time-series into multiple tables, forming a single virtual entity called a **hypertable**.
  - It allows for the **compression** of data, something that Postgres does not do by default.

- **Schema:** Tables are **normal Postgres tables**, where one has to specify a time column in order to create a hypertable.

- **Queries: Full SQL support**, with the addition of custom time-series functions.
  - **Custom time-series operators:** first, last, histogram, interpolation, time bucketing, gap filling, etc.
  - It also supports **continuous queries**

Challenges and Open Problems

Challenges and Open Problems

- we are still far from having solved the problem
- several challenges remain in terms of
  - usability, ease of use
  - scalability, distribution
  - benchmarking
- these challenges derive from modern data series applications

Massive Data Series Collections

Publications

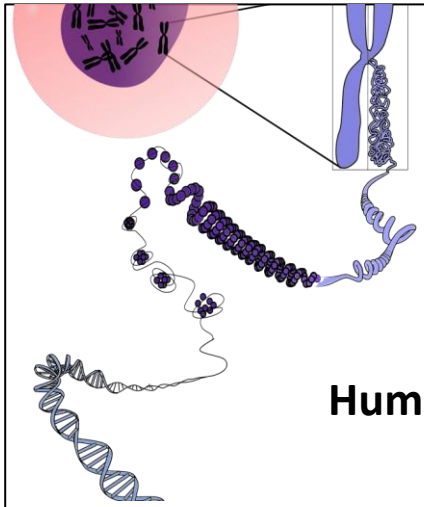
Palpanas-SIGREC'19

NASA's Solar Observatory

1.5 TB per day

Large Synoptic Survey Telescope (2019)

~30 TB per night



Human Genome project

130 TB

passenger aircrafts

20 TB per hour

data center and services monitoring

2B data series

4M points/sec

Challenges and Open Problems Outline

- sequence management system
- benchmarking
- interactive analytics
- parallelization and distribution
- general high-dimensional vectors
- deep learning

Publications

- Zoumbatianos ICDE'18
- Palpanas-HPCS'17
- Palpanas-SIGREC'15

Management System

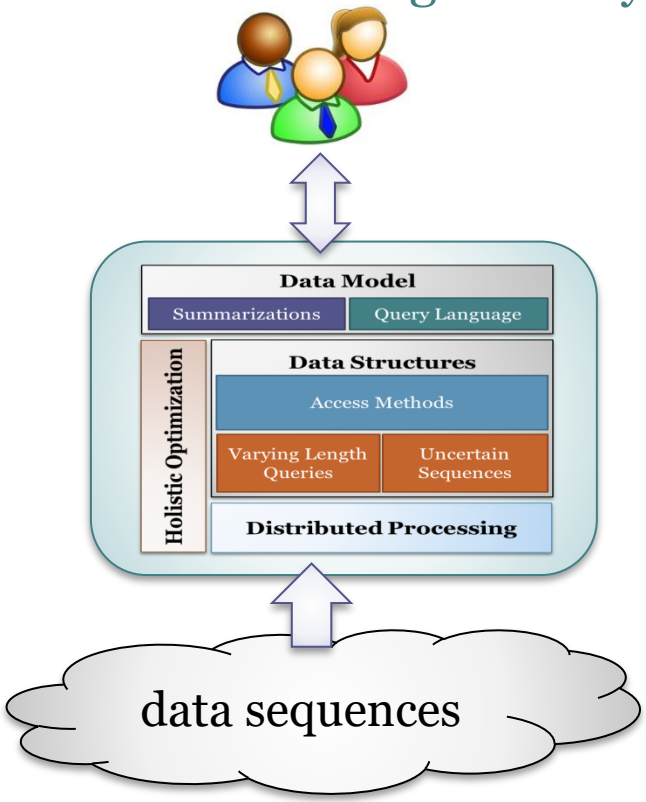
"enable practitioners and non-expert users to easily and efficiently manage and analyze massive data series collections"

Publications

- Zoumbatianos ICDE'18
- Palpanas-HPCS'17
- Palpanas-SIGREC'15

Management System

- Big Sequence Management System
  - general purpose data series management system

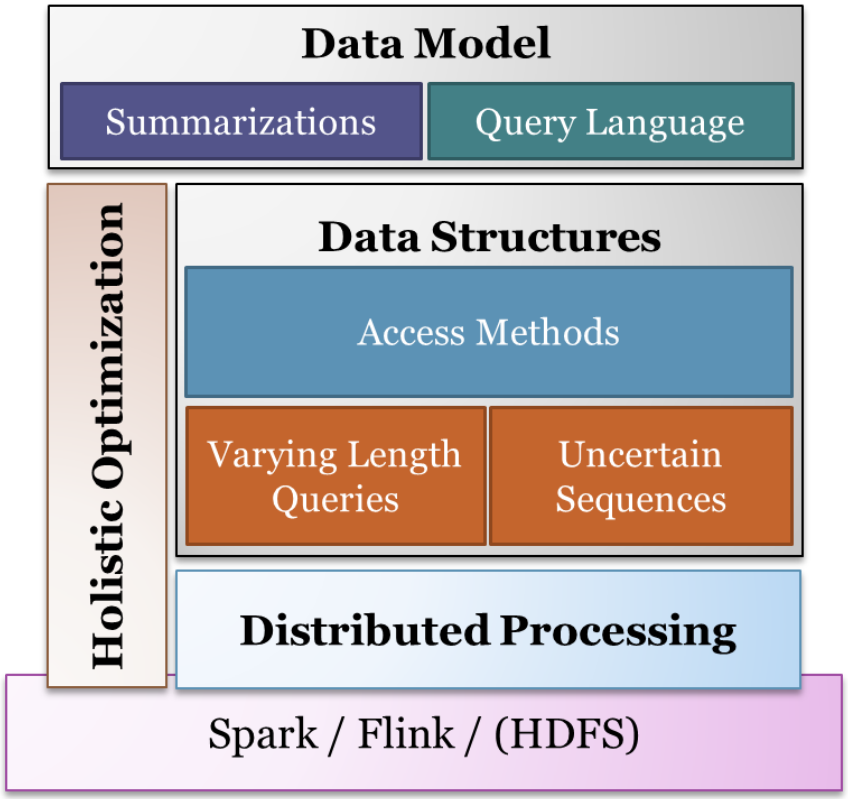


Publications

- Zoumbatianos ICDE'18
- Palpanas-HPCS'17
- Palpanas-SIGREC'15

Management System

- Big Sequence Management System

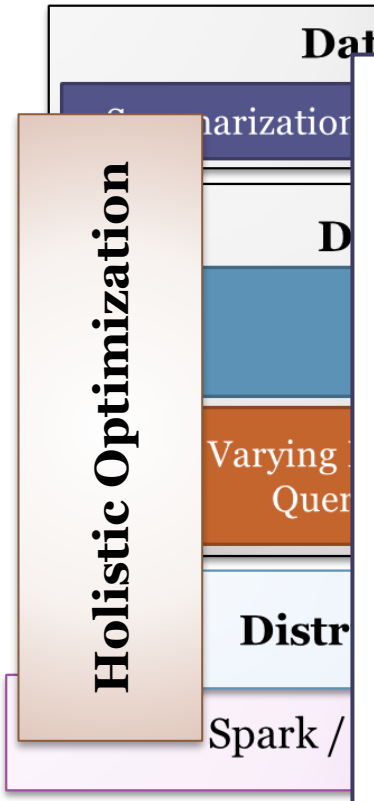


Publications

- Zoumbatianos ICDE'18
- Palpanas-HPCS'17
- Palpanas-SIGREC'15
- Echihabi-PVLDB'18

Management System

- Big Sequence Management System



	Dataset	Idx	Exact 100	Idx+ Exact 100	Idx+ Exact 10K	Exact Easy-20	Exact Hard-20
		Scenarios					
HDD	Small	A	D	S	D	D	D
HDD	Large	A	D	S	D	D	D
HDD	Astro	A	U	U	V	V	U
HDD	Deep1B	A	U	U	U	D	U
HDD	SALD	A	D	I	D	D	D
HDD	Seismic	A	D	S	D	D	U
SSD	Small	S	D	I	D	I	D
SSD	Large	S	D	I	D	I	D
SSD	Astro	I	V	V	V	V	V
SSD	Deep1B	S	I	I	V	I	U
SSD	SALD	S	I	I	I	I	V
SSD	Seismic	A	V	V	V	D	V

A: ADS, D: DSTree, I: iSAX2+
S: SFA, U: UCR-Suite, V: VA+file

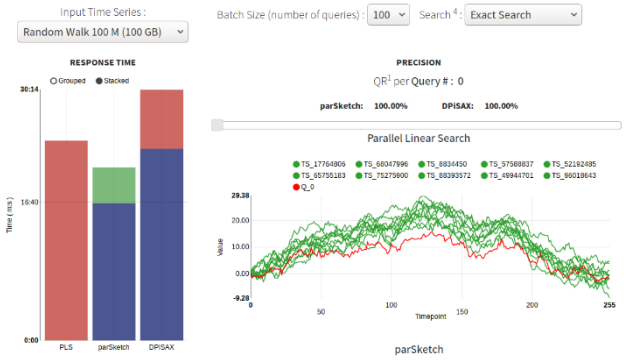
BestNeighbor: Choosing Indexing Method for Given Dataset

Publications

Lavchenko-KAIS'20

- method to choose between DPiSAX and ParSketch
  - based on data power spectrum
    - iSAX less efficient than ParSketch for high-frequency data
- BestNeighbor uses dataset characteristics (Fourier coefficients), and chooses
  - ParSketch: if there is substantial power at least up to the 30th coefficient
  - DPiSAX: otherwise (most of energy in low order Fourier coefficients)

http://imitates.gforge.inria.fr/



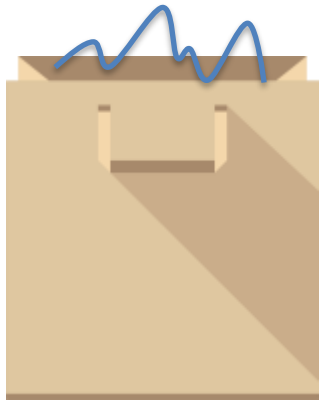
Challenges and Open Problems Outline

- sequence management system
- benchmarking
- interactive analytics
- parallelization and distribution
- general high-dimensional vectors
- deep learning

Previous Studies

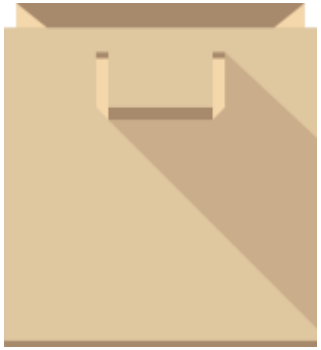
evaluate **performance** of **indexing methods** using **random queries**

- chosen from the data (with/without noise)



Previous Studies

With or without noise



Problem with Random Queries

No control on their ***characteristics***

We **cannot properly evaluate** summarizations and indexes

We need queries that cover the entire range from easy to hard

Publications

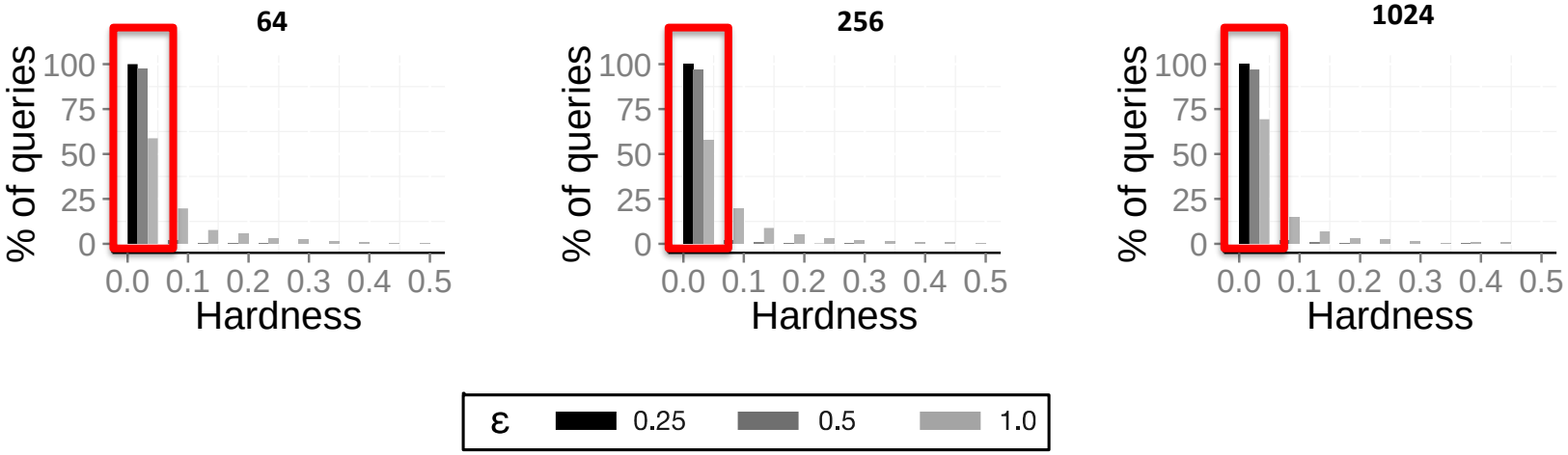
Zoumbatianos KDD ‘15

Zoumbatianos TKDE ‘18

Previous Workloads

Most previous workloads are ***skewed*** to ***easy*** queries

DNA



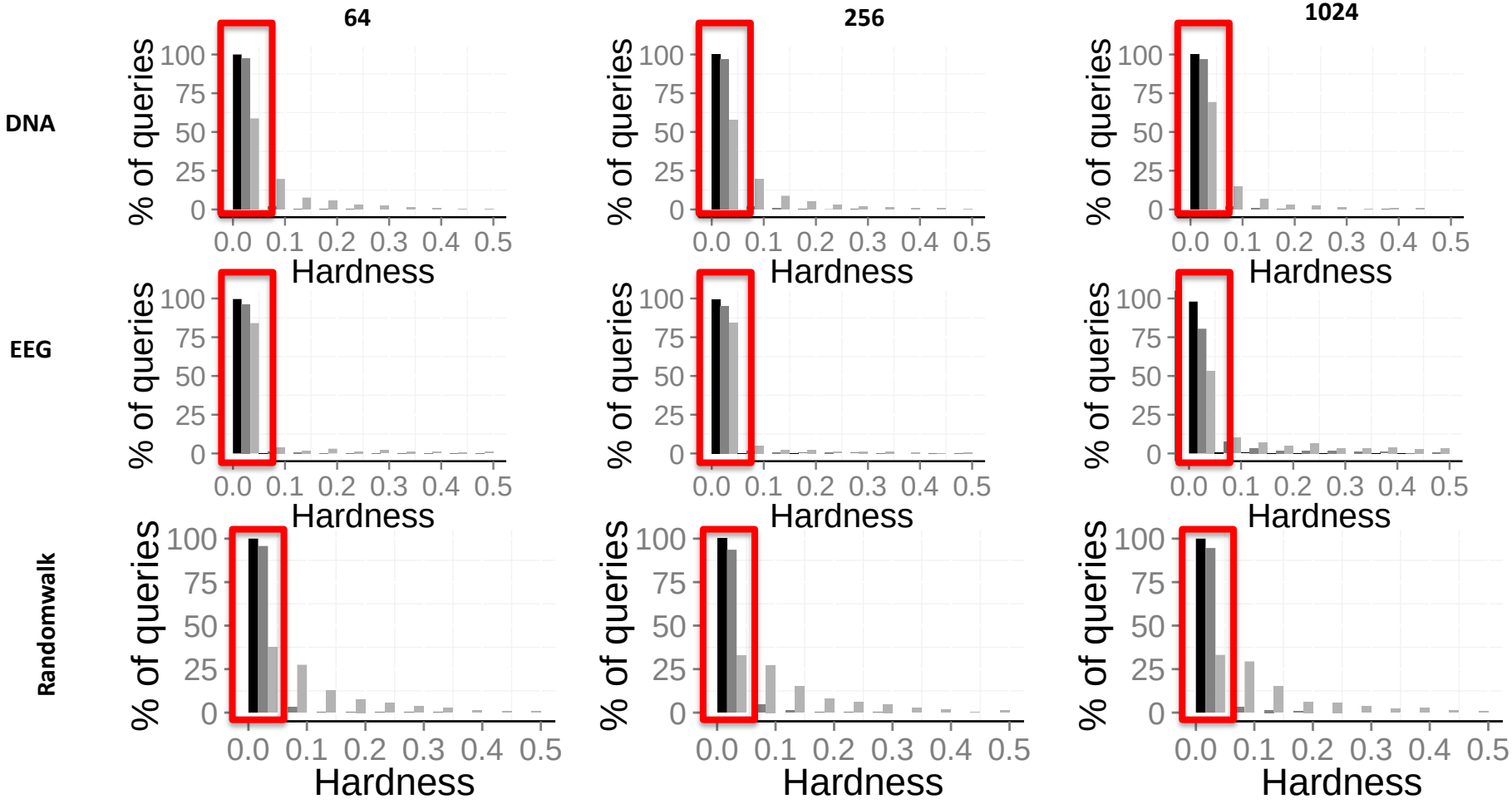
Publications

Zoumbatianos KDD '15

Zoumbatianos TKDE '18

Previous Workloads

Most previous workloads are ***skewed*** to ***easy*** queries



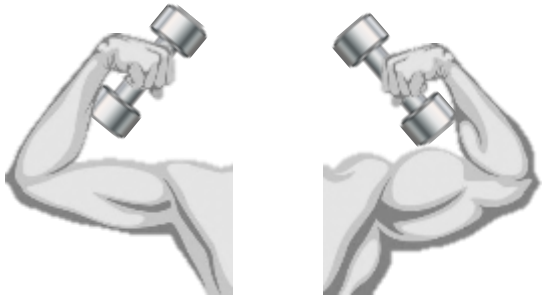
Publications

Zoumbatianos KDD ‘15

Zoumbatianos TKDE ‘18

Benchmark Workloads

If all queries are **easy**
all indexes look **good**



If all queries are **hard**
all indexes look **bad**

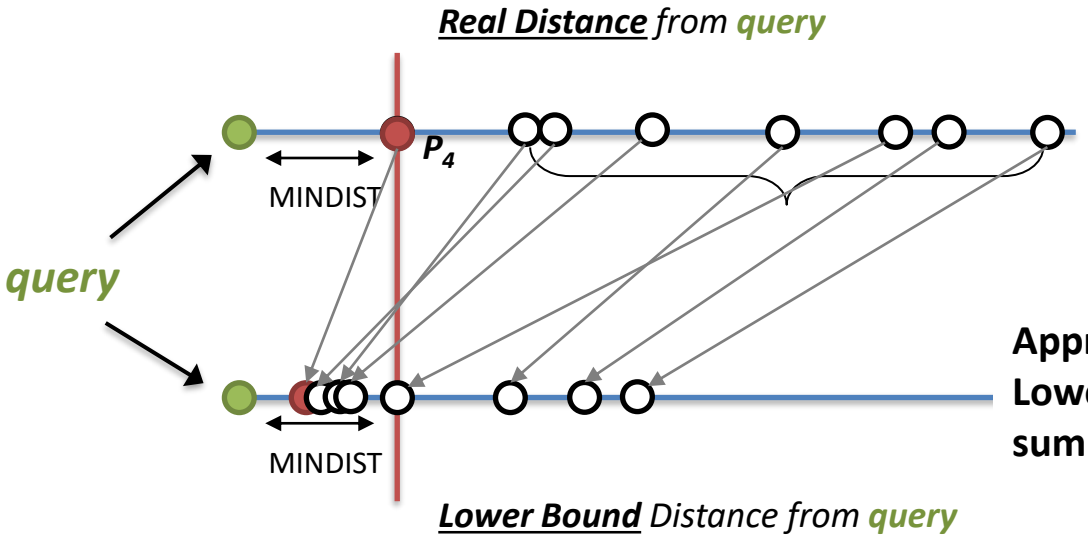
need **methods** for **generating** queries of **varying hardness**

Publications

Zoumbatianos KDD ‘15

Zoumbatianos TKDE ‘18

Characterizing Queries



Approximating distances using **Lower Bounding functions** on **summarizations**.

Publications

Zoumbatianos KDD '15

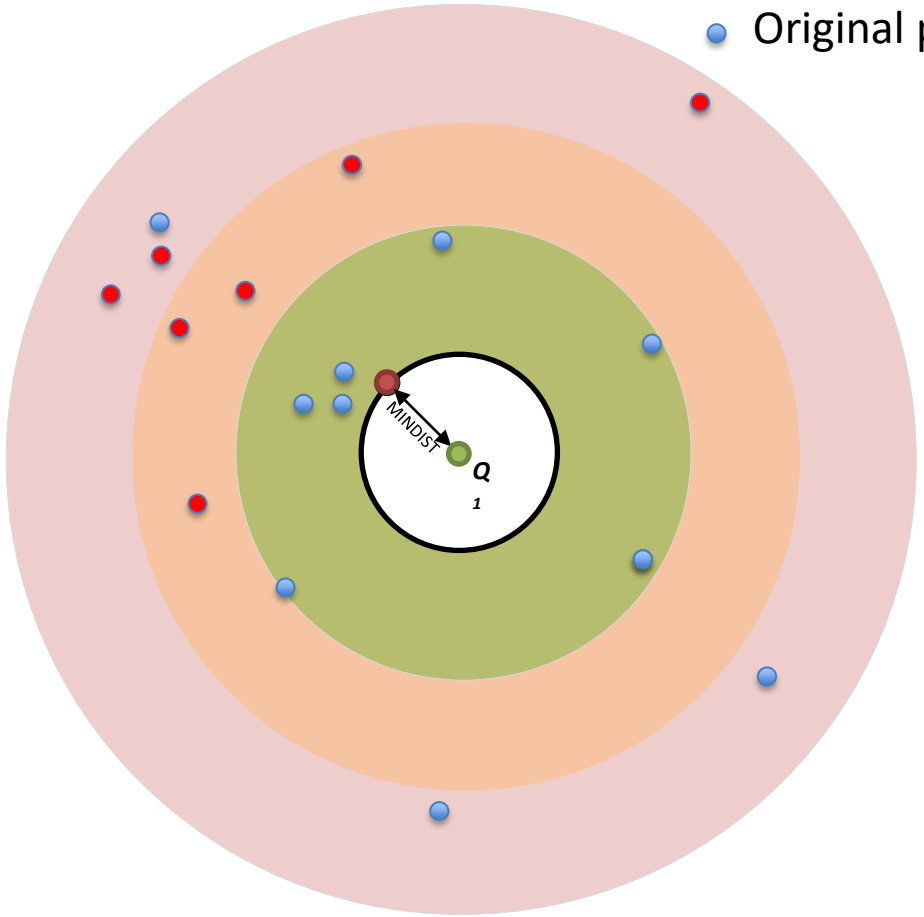
Zoumbatianos TKDE '18

Densification Method:

Equi-densification

Distribute points such that:
The **worse** a summarization
the more data it checks

Equal number of points in every "zone"



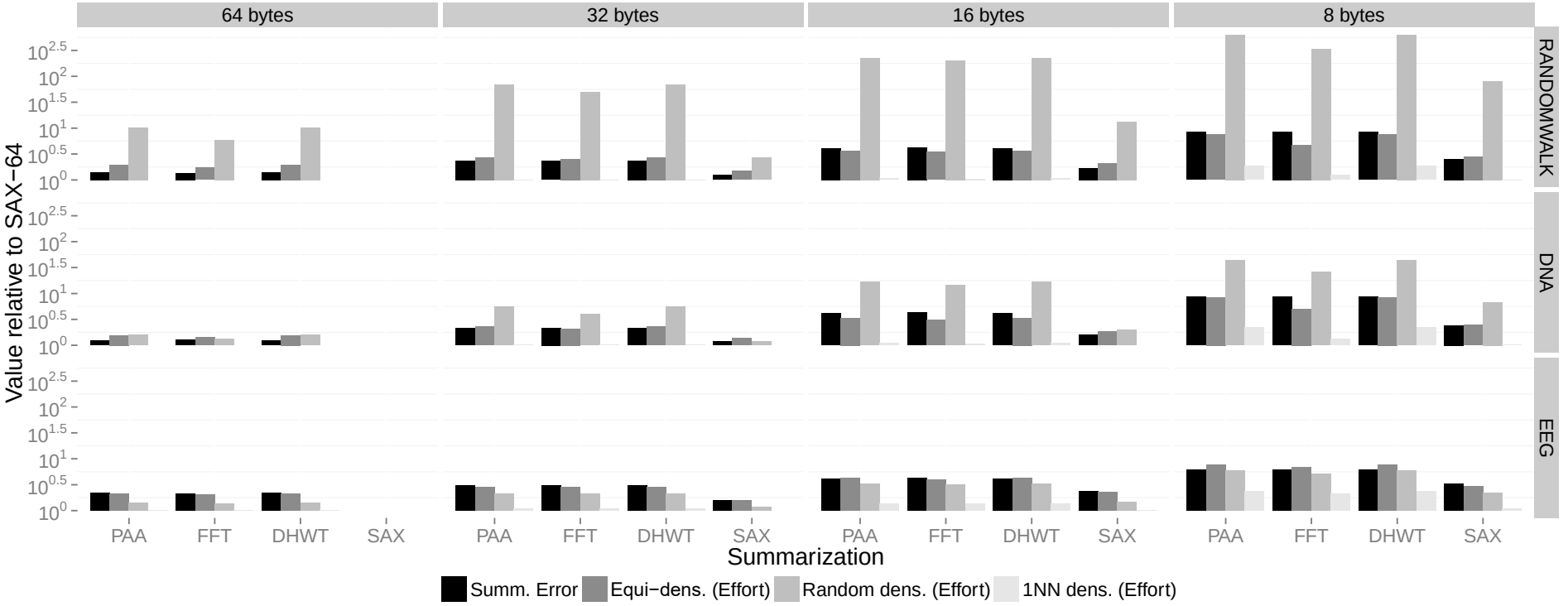
Experiments
Densification Methods

Publications
- Zoumbatianos KDD '15
- Zoumbatianos TKDE '18

Using all datasets of size 256 (100 queries for each dens. method), we measured the:

- ***1-TLB: Summarization Error*** *(0: perfect bound, 1: worst possible bound)*
- ***Minimum Effort*** *for a set of summarizations using 8 – 64 bytes.*

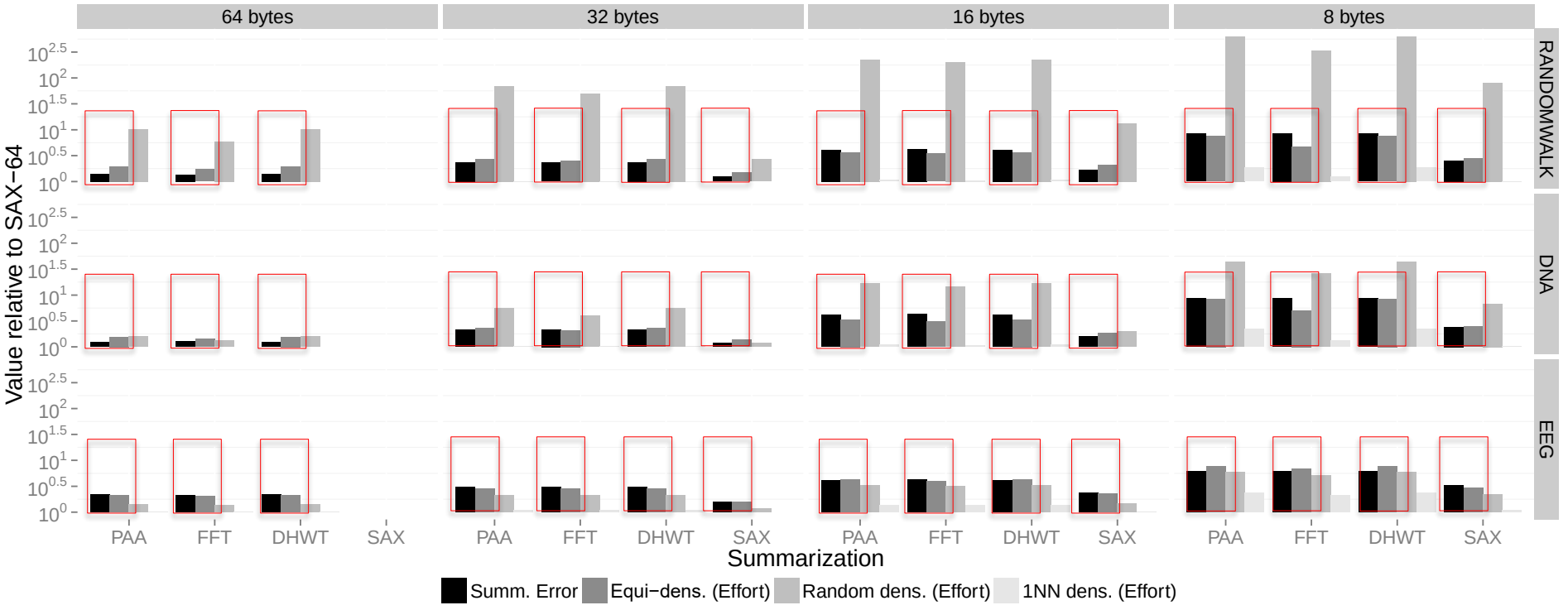
Normalized over SAX-64



Experiments
Densification Methods

Publications
- Zoumbatianos KDD '15
- Zoumbatianos TKDE '18

For **equi-densification**
normalized Effort is closer to the **normalized Summarization Error**
The worse a summarization the bigger effort it does



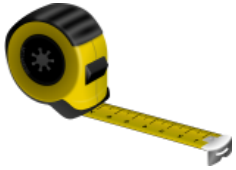
Publications

Zoumbatianos KDD ‘15

Zoumbatianos TKDE ‘18

Summary

Pros:



Theoretical background
Methodology for characterizing
NN queries for data series indexes

Nearest neighbor query workload generator
Designed to stress-test data series indexes
at varying levels of difficulty

Cons:

Time complexity
Need new approach to scale to very large datasets

Challenges and Open Problems Outline

- sequence management system
- benchmarking
- interactive analytics
- parallelization and distribution
- general high-dimensional vectors
- deep learning

Interactive Analytics?

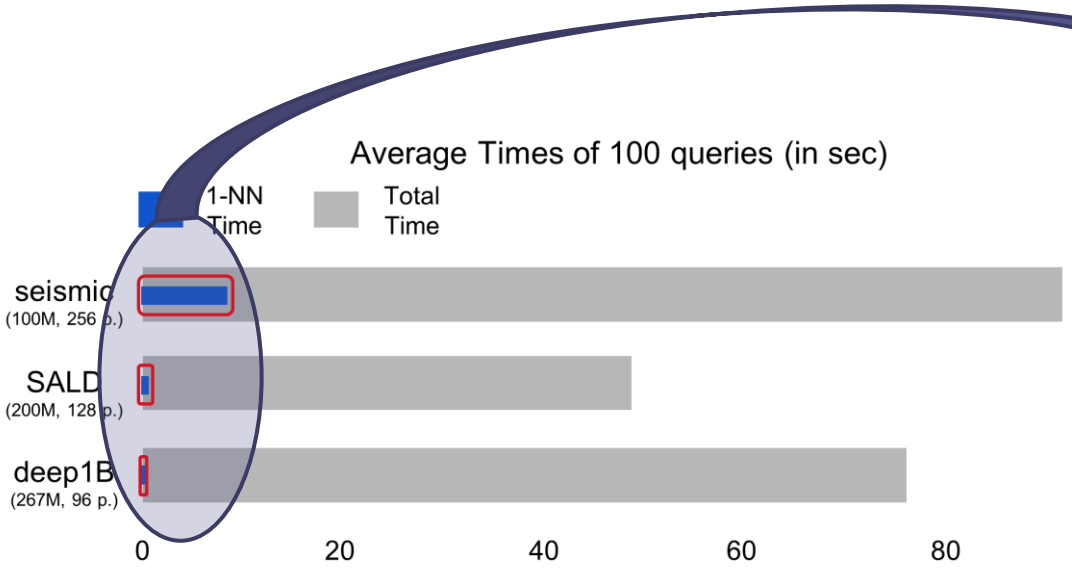
- data series analytics is computationally expensive
  - very high inherent complexity
- may not always be possible to remove delays
  - but could try to hide them!

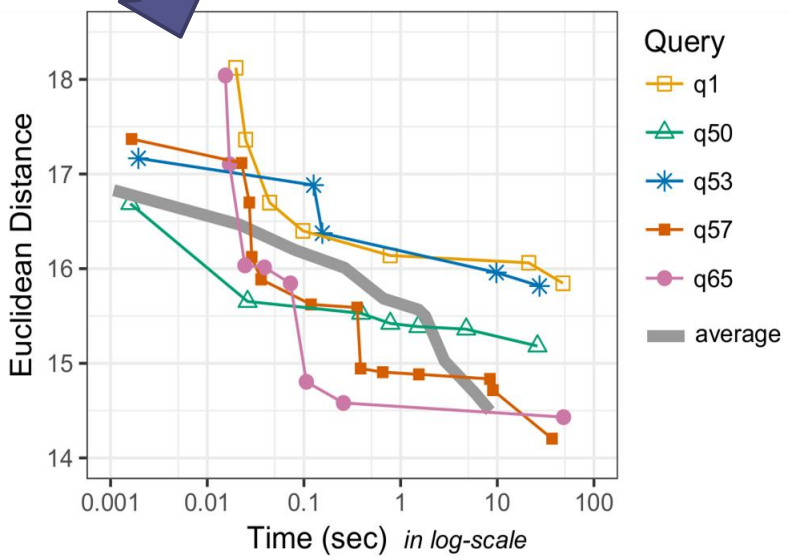
Need for Interactive Analytics

Publications

Gogolou-BigVis'19

- interaction with users offers new opportunities
  - progressive answers
    - produce intermediate results
      - iteratively converge to final, correct solution



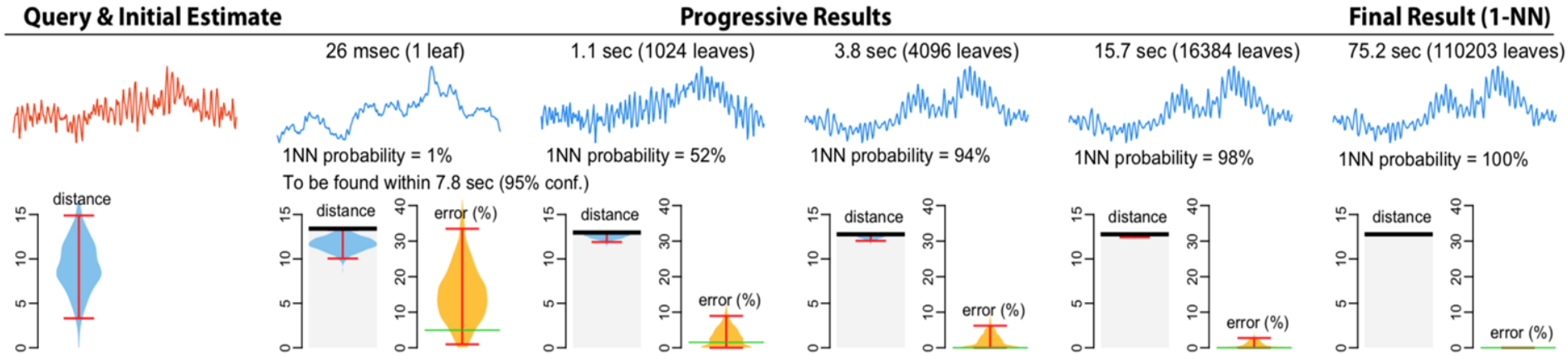


Need for Interactive Analytics

Publications

- Gogolou-BigVis'19
- Gogolou-SIGMOD'20

- interaction with users offers new opportunities
  - progressive answers
    - produce intermediate results
      - iteratively converge to final, correct solution
    - provide bounds on the errors (of the intermediate results) along the way



Need for Interactive Analytics

Publications

Gogolou-Vis'18

- interaction with users offers new opportunities
  - progressive answers
    - produce intermediate results
      - iteratively converge to final, correct solution
    - provide bounds on the errors (of the intermediate results) along the way

- several exciting research problems in intersection of visualization and data management
  - *frontend*: HCI/visualizations for querying/results display
  - *backend*: efficiently supporting these operations

Challenges and Open Problems Outline

- sequence management system
- benchmarking
- interactive analytics
- parallelization and distribution
- general high-dimensional vectors
- deep learning

Publications

Palpanas-HPCS'17

Need for Parallelization/Distribution

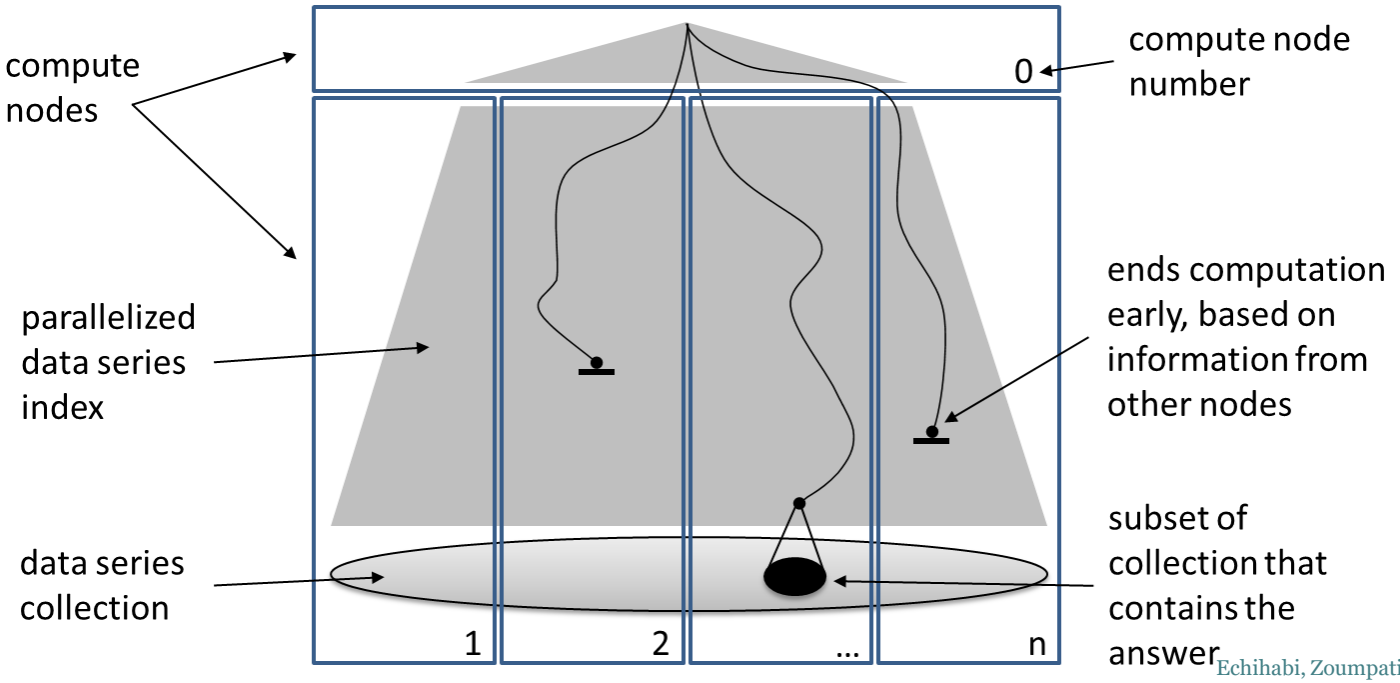
- take advantage of all modern hardware opportunities!
  - Single Instruction Multiple Data (SIMD)
    - natural for data series operations
  - multi-tier CPU caches
    - design data structures aligned to cache lines
  - multi-core and multi-socket architectures
    - use parallelism inside each computation server
  - Graphics Processing Units (GPUs)
    - propose massively parallel techniques for GPUs
  - new storage solutions: NVRAMs, FPGAs
    - develop algorithms that take these new characteristics/tradeoffs into account
  - compute clusters
    - distribute operation over many machines

Publications

Palpanas-HPCS'17

Need for Parallelization/Distribution

- further scale-up and scale-out possible!
  - techniques inherently parallelizable
    - across cores, across machines



Publications

Palpanas-HPCS'17

Need for Parallelization/Distribution

- further scale-up and scale-out possible!
  - techniques inherently parallelizable
    - across cores, across machines
- need to
  - propose methods for concurrent query answering
  - combine multi-core and distributed methods
  - examine FPGA and NVM technologies
- more involved solutions required when optimizing for energy
  - reducing execution time is relatively easy
  - minimizing total work (energy) is more challenging

Challenges and Open Problems Outline

- sequence management system
- benchmarking
- interactive analytics
- parallelization and distribution
- general high-dimensional vectors
- deep learning

Data Series vs. high-d Vectors

Publications

Echihabi-WIMS'20

- two sides of the same(?) coin
  - data series as multidimensional points
  - for a specific ordering of the dimensions
- several techniques for similarity search in high-d vectors
  - using LSH (SRS), space quantization (IMI), k-NN graphs (HNSW)
- how do these high-d vector techniques compare to data series techniques?
  - conducted extensive experimental comparison

Data Series vs. high-d Vectors

Publications

Echihabi-PVLDB'19

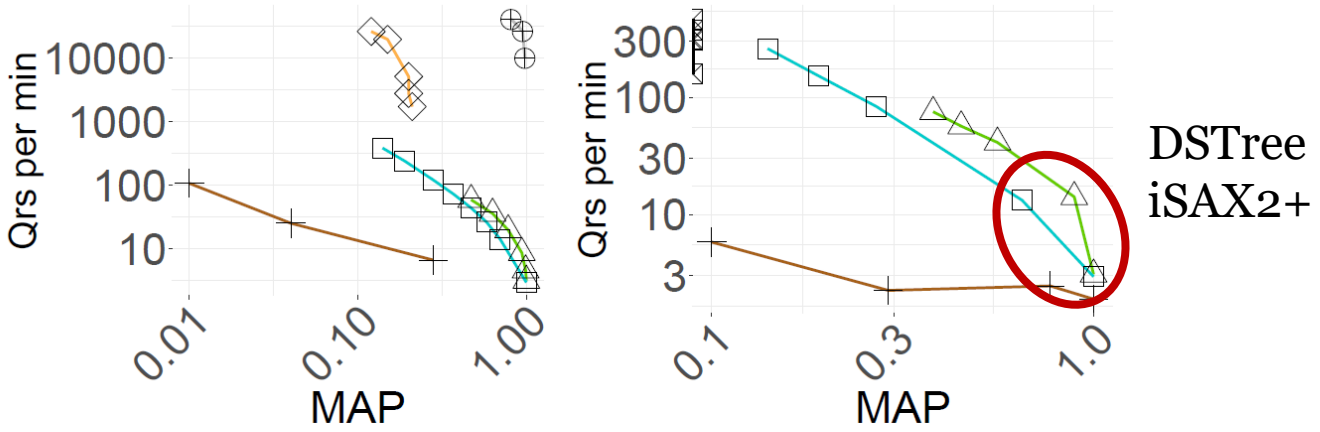
- **data series techniques** are the **overall winners**, even on **general high-d vector** data

Data Series vs. high-d Vectors

Publications

Echihabi-PVLDB'19

- **data series techniques** are the **overall winners**, even on **general high-d vector** data
  - perform the best for approximate queries with probabilistic guarantees (δ-ε-approximate search), in-memory and on-disk



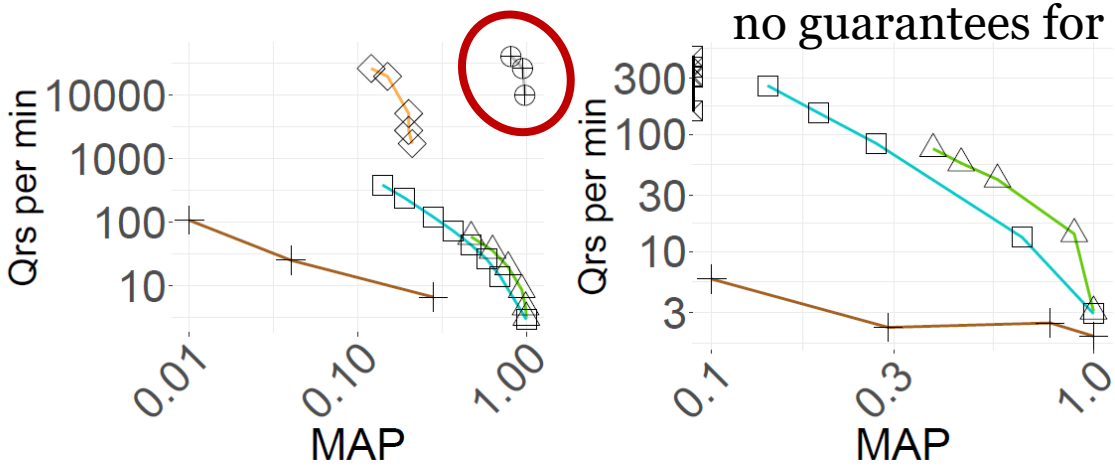
Data Series vs. high-d Vectors

Publications

Echihabi-PVLDB'19

- **data series techniques** are the **overall winners**, even on **general high-d vector** data
  - perform the best for approximate queries with probabilistic guarantees (δ-ε-approximate search), in-memory and on-disk

HNSW: only for in-memory data, with no guarantees for the answers



(s) Deep25GB(ng) (t) Deep25GB(δϵ)

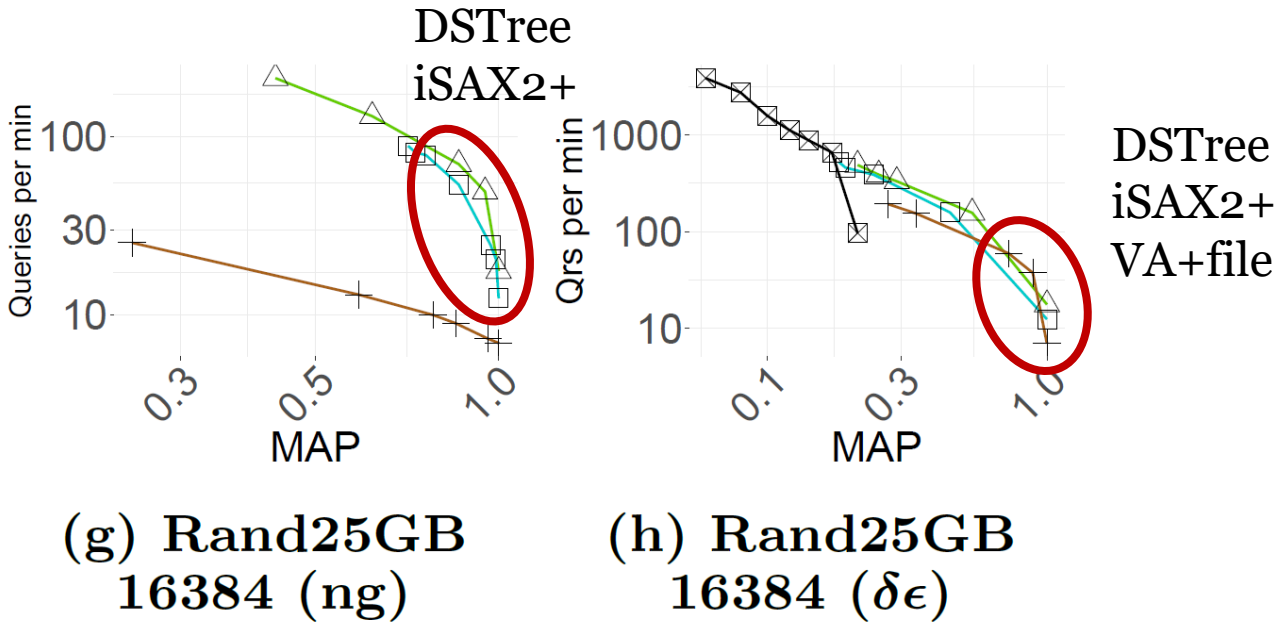
DSTree HNSW IMI iSAX2+ SRS VA+file

Data Series vs. high-d Vectors

Publications

Echihabi-PVLDB'19

- **data series techniques** are the **overall winners**, even on **general high-d vector** data
  - perform the best for approximate queries with probabilistic guarantees (δ-ε-approximate search), in-memory and on-disk
  - perform the best for long vectors, in-memory and on-disk

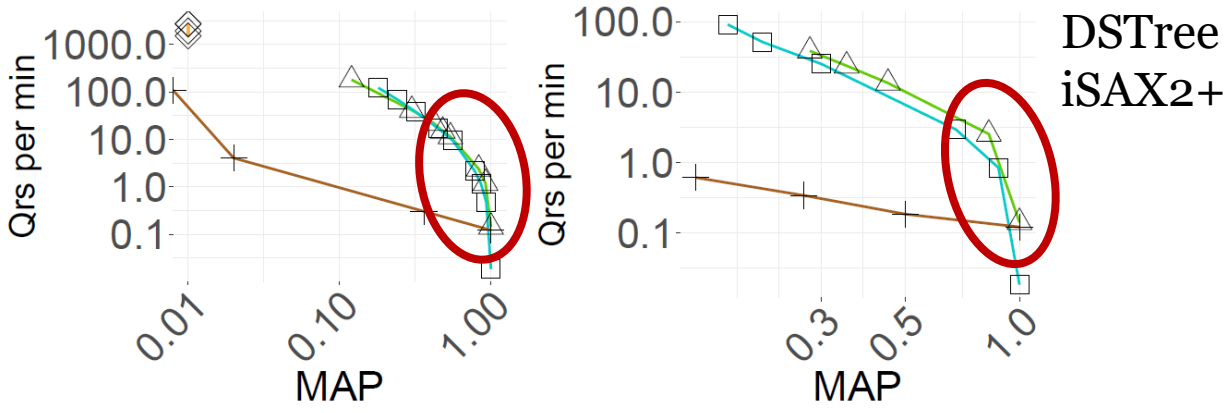


Data Series vs. high-d Vectors

Publications

Echihabi-PVLDB'19

- **data series techniques** are the **overall winners**, even on **general high-d vector** data
  - perform the best for approximate queries with probabilistic guarantees (δ-ε-approximate search), in-memory and on-disk
  - perform the best for long vectors, in-memory and on-disk
  - perform the best for disk-resident vectors



(m) Deep250GB(ng) **(n) Deep250GB(δε)**

DSTree HNSW IMI iSAX2+ SRS VA+file

Data Series vs. high-d Vectors

Publications

Echihabi-PVLDB'19

- **data series techniques** are the **overall winners**, even on **general high-d vector** data

- several new applications (and challenges) for data series similarity search techniques!

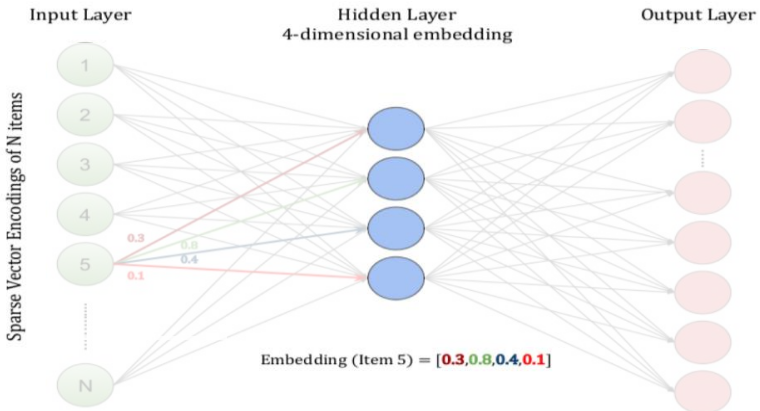
Challenges and Open Problems Outline

- sequence management system
- benchmarking
- interactive analytics
- parallelization and distribution
- general high-dimensional vectors
- deep learning

Connections to Deep Learning

- data series indexing for deep embeddings

sequences
text
images
video
graphs
…



deep embeddings
high-d vectors learned using a DNN

Connections to Deep Learning

- data series indexing for deep embeddings
  - deep embeddings are high-d vectors
  - data series techniques provide effective/scalable similarity search
- deep learning for summarizing data series
  - eg, autoencoders can learn efficient data series summaries
- deep learning for designing index data structures
  - learn an index for similarity search
- deep learning for query optimization
  - search space is vast
  - learn optimization function

Overall Conclusions

- data series is a very common data type
  - across several different domains and applications
- complex data series analytics are challenging
  - have very high complexity
  - efficiency comes from data series management/indexing techniques
- need for Sequence Management System
  - optimize operations based on data/hardware characteristics
  - transparent to user
- several exciting research opportunities

thank you!

google: **Karima Echihabi**
Kostas Zoumpatianos
Themis Palpanas

visit: http://**nestordb.com**

References (chronological order)

- Ramer, U. (1972). An iterative procedure for the polygonal approximation of planar curves. *Computer Graphics and Image Processing*. 1: pp. 244-256.
- Douglas, D. H. & Peucker, T. K.(1973). Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature. *Canadian Cartographer*, Vol. 10, No. 2, December. pp. 112-122.
- Duda, R. O. and Hart, P. E. 1973. Pattern Classification and Scene Analysis. Wiley, New York.
- Pavlidis, T. (1976). Waveform segmentation through functional approximation. *IEEE Transactions on Computers*.
- Ishijima, M., et al. (1983). Scan-Along Polygonal Approximation for Data Compression of Electrocardiograms. *IEEE Transactions on Biomedical Engineering*. BME-30(11):723-729.
- N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: an efficient and robust access method for points and rectangles. In SIGMOD, pages 322–331, 1990.
- C. Faloutsos, M. Ranganathan, & Y. Manolopoulos. Fast Subsequence Matching in Time-Series Databases. In Proc. ACM SIGMOD Int'l Conf. on Management of Data, pp 419–429, 1994.
- McKee, J.J., Evans, N.E., & Owens, F.J. (1994). Efficient implementation of the Fan/SAPA-2 algorithm using fixed point arithmetic. *Automedica*. Vol. 16, pp 109-117.
- Koski, A., Juhola, M. & Meriste, M. (1995). Syntactic Recognition of ECG Signals By Attributed Finite Automata. *Pattern Recognition*, 28 (12), pp. 1927-1940.
- Seshadri P., Livny M. & Ramakrishnan R. (1995): SEQ: A Model for Sequence Databases. ICDE 1995: 232-239
- Shatkay, H. (1995). Approximate Queries and Representations for Large Data Sequences. *Technical Report cs-95-03*, Department of Computer Science, Brown University.
- Shatkay, H., & Zdonik, S. (1996). Approximate queries and representations for large data sequences. *Proceedings of the 12th IEEE International Conference on Data Engineering*. pp 546-553.
- Vullings, H.J.L.M., Verhaegen, M.H.G. & Verbruggen H.B. (1997). ECG Segmentation Using Time-Warping. *Proceedings of the 2nd International Symposium on Intelligent Data Analysis*.

References (chronological order)

- Keogh, E., & Smyth, P. (1997). A probabilistic approach to fast pattern matching in time series databases. *Proceedings of the 3rd International Conference of Knowledge Discovery and Data Mining*. pp 24-20.
- P. Ciaccia, M. Patella, and P. Zezula. M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. Proceedings of VLDB'97, pp 426–435.
- Heckbert, P. S. & Garland, M. (1997). Survey of polygonal surface simplification algorithms, Multiresolution Surface Modeling Course. *Proceedings of the 24th International Conference on Computer Graphics and Interactive Techniques*.
- Piotr Indyk, Rajeev Motwani. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. STOC 1998.
- Qu, Y., Wang, C. & Wang, S. (1998). Supporting fast search in time series for movement patterns in multiples scales. *Proceedings of the 7th International Conference on Information and Knowledge Management*.
- Keogh, E., & Pazzani, M. (1998). An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. *Proceedings of the 4th International Conference of Knowledge Discovery and Data Mining*. pp 239-241, AAAI Press.
- Hunter, J. & McIntosh, N. (1999). Knowledge-based event detection in complex time series data. *Artificial Intelligence in Medicine*. pp. 271-280. Springer.
- Keogh, E. & Pazzani, M. (1999). Relevance feedback retrieval of time series data. *Proceedings of the 22th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval.*
- P. Ciaccia and M. Patella. PAC Nearest Neighbor Queries: Approximate and Controlled Search in HighDimensional and Metric Spaces. In ICDE, pages 244– 255, 2000.
- H. Ferhatosmanoglu, E. Tuncel, D. Agrawal, and A. El Abbadi. Vector Approximation Based Indexing for Non-uniform High Dimensional Data Sets. In CIKM, pp 202–209, 2000.

References (chronological order)

- Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D., & Allan, J. (2000). Mining of Concurent Text and Time Series. *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining*. pp. 37-44.
- Wang, C. & Wang, S. (2000). Supporting content-based searches on time Series via approximation. *Proceedings of the 12th International Conference on Scientific and Statistical Database Management*.
- Keogh, E., Chu, S., Hart, D. & Pazzani, M. (2001). An Online Algorithm for Segmenting Time Series. In *Proceedings of IEEE International Conference on Data Mining*. pp 289-296.
- Ge, X. & Smyth P. (2001). Segmental Semi-Markov Models for Endpoint Detection in Plasma Etching. To appear in *IEEE Transactions on Semiconductor Engineering*.
- Eamonn J. Keogh, Shruti Kasetty: On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. Data Min. Knowl. Discov. 7(4): 349-371 (2003)
- T. Palpanas, M. Vlachos, E. Keogh, D. Gunopulos, W. Truppel (2004). Online Amnesic Approximation of Streaming Time Series. In *ICDE* . Boston, MA, USA, March 2004.
- E. Keogh. Tutorial on Data Mining and Machine Learning in Time Series Databases. KDD 2004.
- Richard Cole, Dennis E. Shasha, Xiaojian Zhao: Fast window correlations over uncooperative time series. KDD 2005: 743-749
- Jessica Lin, Eamonn J. Keogh, Li Wei, Stefano Lonardi: Experiencing SAX: a novel symbolic representation of time series. Data Min. Knowl. Discov. 15(2): 107-144 (2007)
- Jin Shieh, Eamonn J. Keogh: iSAX: indexing and mining terabyte sized time series. KDD 2008: 623-631
- Themis Palpanas, Michail Vlachos, Eamonn J. Keogh, Dimitrios Gunopulos: Streaming Time Series Summarization Using User-Defined Amnesic Functions. IEEE Trans. Knowl. Data Eng. 20(7): 992-1006 (2008)
- Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, Eamonn J. Keogh: Querying and mining of time series data: experimental comparison of representations and distance measures. Proc. VLDB Endow. 1(2): 1542-1552 (2008)

References (chronological order)

- M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In VISAPP International Conference on Computer Vision Theory and Applications, pages 331–340, 2009
- Alessandro Camerra, Themis Palpanas, Jin Shieh, Eamonn J. Keogh: iSAX 2.0: Indexing and Mining One Billion Time Series. ICDM 2010: 58-67
- S. Kashyap and P. Karras. Scalable kNN search on vertically stored time series. In KDD, pages 1334–1342 (2011)
- P. Schafer and M. Hogvist. Sfa: A symbolic fourier approximation and index for similarity search in high dimensional datasets. EDBT Conference 2012: 516–527
- T. Rakthanmanon, B. J. L. Campana, A. Mueen, G. E. A. P. A. Batista, M. B. Westover, Q. Zhu, J. Zakaria, and E. J. Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In KDD, pages 262–270. ACM, 2012.
- Y. Wang, P. Wang, J. Pei, W. Wang, and S. Huang. A data-adaptive and dynamic segmentation index for whole matching on time series. PVLDB, 6(10):793–804, 2013.
- M. Norouzi and D. J. Fleet. Cartesian K-Means. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, CVPR ’13, pages 3017–3024, 2013
- Alessandro Camerra, Jin Shieh, Themis Palpanas, Thanawin Rakthanmanon, Eamonn J. Keogh: Beyond one billion time series: indexing and mining very large time series collections with iSAX2+. Knowl. Inf. Syst. 39(1): 123-151 (2014)
- Y. Sun, W. Wang, J. Qin, Y. Zhang, and X. Lin. SRS: Solving c-approximate Nearest Neighbor Queries in High Dimensional Euclidean Space with a Tiny Index. PVLDB, 8(1):1–12, 2014
- Kostas Zoumpatianos, Stratos Idreos, Themis Palpanas: Indexing for interactive exploration of big data series. SIGMOD Conference 2014: 1555-1566

References (chronological order)

- Y. Malkov, A. Ponomarenko, A. Logvinov, and V. Krylov. Approximate nearest neighbor algorithm based on navigable small world graphs. Information Systems (IS), 45:61 – 68, 2014.
- T. Ge, K. He, Q. Ke, and J. Sun. Optimized Product Quantization. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI), 36(4):744–755, Apr. 2014
- A. Babenko and V. Lempitsky. The Inverted MultiIndex. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 37(6):1247–1260, June 2015.
- Kostas Zoumpatianos, Stratos Idreos, Themis Palpanas: RINSE: Interactive Data Series Exploration with ADS+. Proc. VLDB Endow. 8(12): 1912-1915 (2015)
- Kostas Zoumpatianos, Yin Lou, Themis Palpanas, Johannes Gehrke: Query Workloads for Data Series Indexes. KDD 2015: 1603-1612
- Q. Huang, J. Feng, Y. Zhang, Q. Fang, and W. Ng. Query-aware Locality-sensitive Hashing for Approximate Nearest Neighbor Search. PVLDB, 9(1):1–12, 2015
- Themis Palpanas: Big Sequence Management: A glimpse of the Past, the Present, and the Future. SOFSEM 2016: 63-80
- Kostas Zoumpatianos, Stratos Idreos, Themis Palpanas: ADS: the adaptive data series index. VLDB J. 25(6): 843-866 (2016)
- Y. A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. CoRR, abs/1603.09320, 2016
- Djamel Edine Yagoubi, Reza Akbarinia, Florent Masseglia, Themis Palpanas: DPiSAX: Massively Distributed Partitioned iSAX. ICDM 2017: 1135-1140
- A. Mueen, Y. Zhu, M. Yeh, K. Kamgar, K. Viswanathan, C. Gupta, and E. Keogh. The Fastest Similarity Search Algorithm for Time Series Subsequences under Euclidean Distance, August 2017. http://www.cs.unm.edu/~mueen/ FastestSimilaritySearch.html.

References (chronological order)

- Katsiaryna Mirylenka, Michele Dallachiesa, Themis Palpanas: Correlation-Aware Distance Measures for Data Series. EDBT 2017: 502-505
- Katsiaryna Mirylenka, Michele Dallachiesa, Themis Palpanas: Data Series Similarity Using Correlation-Aware Measures. SSDBM 2017: 11:1-11:12
- Kostas Zoumpatianos, Themis Palpanas: Data Series Management: Fulfilling the Need for Big Sequence Analytics. ICDE 2018: 1677-1678
- A. Arora, S. Sinha, P. Kumar, and A. Bhattacharya. HD-index: Pushing the Scalability-accuracy Boundary for Approximate kNN Search in High-dimensional Spaces. PVLDB, 11(8):906–919, 2018
- Michele Linardi, Themis Palpanas: ULISSE: ULtra Compact Index for Variable-Length Similarity Search in Data Series. ICDE 2018: 1356-1359
- J. Wang, T. Zhang, j. song, N. Sebe, and H. T. Shen. A survey on learning to hash. TPAMI, 40(4): 769-790 (2018).
- Kostas Zoumpatianos, Yin Lou, Ioana Ileana, Themis Palpanas, Johannes Gehrke: Generating data series query workloads. VLDB J. 27(6): 823-846 (2018)
- Haridimos Kondylakis, Niv Dayan, Kostas Zoumpatianos, Themis Palpanas: Coconut: A Scalable Bottom-Up Approach for Building Data Series Indexes. Proc. VLDB Endow. 11(6): 677-690 (2018)
- Cagatay Turkay, Nicola Pezzotti, Carsten Binnig, Hendrik Strobelt, Barbara Hammer, Daniel A. Keim, Jean-Daniel Fekete, Themis Palpanas, Yunhai Wang, Florin Rusu: Progressive Data Science: Potential and Challenges. CoRR abs/1812.08032 (2018)
- Michele Linardi, Themis Palpanas: Scalable, Variable-Length Similarity Search in Data Series: The ULISSE Approach. Proc. VLDB Endow. 11(13): 2236-2248 (2018)
- Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, Houda Benbrahim: The Lernaean Hydra of Data Series Similarity Search: An Experimental Evaluation of the State of the Art. Proc. VLDB Endow. 12(2): 112-127 (2018)
- Botao Peng, Panagiota Fatourou, Themis Palpanas: ParIS: The Next Destination for Fast Data Series Indexing and Query Answering. BigData 2018: 791-800

References (chronological order)

- D.E. Yagoubi, R. Akbarinia, B. Kolev, O. Levchenko, F. Masseglia, P. Valduriez, D. Shasha. ParCorr: efficient parallel methods to identify similar time series pairs across sliding windows. Data Mining and Knowledge Discovery (DMKD), 2018
- Haridimos Kondylakis, Niv Dayan, Kostas Zoumpatianos, Themis Palpanas: Coconut Palm: Static and Streaming Data Series Exploration Now in your Palm. SIGMOD Conference 2019: 1941-1944
- Themis Palpanas, Volker Beckmann: Report on the First and Second Interdisciplinary Time Series Analysis Workshop (ITISA). SIGMOD Rec. 48(3): 36-40 (2019)
- Oleksandra Levchenko, Boyan Kolev, Djamel Edine Yagoubi, Dennis E. Shasha, Themis Palpanas, Patrick Valduriez, Reza Akbarinia, Florent Masseglia: Distributed Algorithms to Find Similar Time Series. ECML/PKDD (3) 2019: 781-785
- Haridimos Kondylakis, Niv Dayan, Kostas Zoumpatianos, Themis Palpanas: Coconut: sortable summarizations for scalable indexes over static and streaming data series. VLDB J. 28(6): 847-869 (2019)
- Danila Piatov, Sven Helmer, Anton Dignös, Johann Gamper: Interactive and space-efficient multi-dimensional time series subsequence matching. Inf. Syst. 82: 121-135 (2019)
- Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, Houda Benbrahim: Return of the Lernaean Hydra: Experimental Evaluation of Data Series Approximate Similarity Search. Proc. VLDB Endow. 13(3): 403-420 (2019)
- C. Fu, C. Xiang, C. Wang, and D. Cai. Fast Approximate Nearest Neighbor Search with the Navigating Spreading-out Graph. PVLDB, 12(5):461–474, 2019.
- Anna Gogolou, Theophanis Tsandilas, Themis Palpanas, Anastasia Bezerianos: Comparing Similarity Perception in Time Series Visualizations. IEEE Trans. Vis. Comput. Graph. 25(1): 523-533 (2019)
- John Paparrizos, Michael J. Franklin: GRAIL: Efficient Time-Series Representation Learning. Proc. VLDB Endow. 12(11): 1762-1777 (2019)
- Jiaye Wu, Peng Wang, Ningting Pan, Chen Wang, Wei Wang, Jianmin Wang: KV-Match: A Subsequence Matching Approach Supporting Normalization and Time Warping. ICDE 2019: 866-877

References (chronological order)

- Liang Zhang, Noura Alghamdi, Mohamed Y. Eltabakh, Elke A. Rundensteiner: TARDIS: Distributed Indexing Framework for Big Time Series Data. ICDE 2019: 1202-1213
- Georgios Chatzigeorgakidis, Dimitrios Skoutas, Kostas Patroumpas, Themis Palpanas, Spiros Athanasiou, Spiros Skiadopoulos: Local Pair and Bundle Discovery over Co-Evolving Time Series. SSTD 2019
- Georgios Chatzigeorgakidis, Dimitrios Skoutas, Kostas Patroumpas, Themis Palpanas, Spiros Athanasiou, Spiros Skiadopoulos: Local Similarity Search on Geolocated Time Series Using Hybrid Indexing. SIGSPATIAL/GIS 2019
- Anna Gogolou, Theophanis Tsandilas, Karima Echihabi, Anastasia Bezerianos, Themis Palpanas: Data Series Progressive Similarity Search with Probabilistic Quality Guarantees. SIGMOD Conference 2020: 1857-1873
- Themis Palpanas. Evolution of a Data Series Index - The iSAX Family of Data Series Indexes. CCIS, 1197 (2020)
- Djamel Edine Yagoubi, Reza Akbarinia, Florent Masseglia, Themis Palpanas: Massively Distributed Time Series Indexing and Querying. IEEE Trans. Knowl. Data Eng. 32(1): 108-120 (2020)
- Botao Peng, Panagiota Fatourou, Themis Palpanas: MESSI: In-Memory Data Series Indexing. ICDE 2020: 337-348
- Kefeng Feng, Peng Wang, Jiaye Wu, Wei Wang: L-Match: A Lightweight and Effective Subsequence Matching Approach. IEEE Access 8: 71572-71583 (2020)
- Chen Wang, Xiangdong Huang, Jialin Qiao, Tian Jiang, Lei Rui, Jinrui Zhang, Rong Kang, Julian Feinauer, Kevin Mcgrail, Peng Wang, Diaohan Luo, Jun Yuan, Jianmin Wang, Jiaguang Sun: Apache IoTDB: Time-series database for Internet of Things. Proc. VLDB Endow. 13(12): 2901-2904 (2020)
- Botao Peng, Panagiota Fatourou, Themis Palpanas. Paris+: Data series indexing on multi-core architectures. TKDE, 2020
- Michele Linardi, Themis Palpanas. Scalable Data Series Subsequence Matching with ULISSE. VLDBJ 2020
- John Paparrizos, Chunwei Liu, Aaron J. Elmore, Michael J. Franklin: Debunking Four Long-Standing Misconceptions of Time-Series Distance Measures. SIGMOD Conference 2020
- Karima Echihabi, Kostas Zoumpatianos, and Themis Palpanas. Scalable Machine Learning on High-Dimensional Vectors: From Data Series to Deep Network Embeddings. In WIMS, 2020

References (chronological order)

- Oleksandra Levchenko, Boyan Kolev, Djamel-Edine Yagoubi, Reza Akbarinia, Florent Masseflia, Themis Palpanas, Dennis Shasha, Patrick Valduriez. BestNeighbor: Efficient Evaluation of kNN Queries on Large Time Series Databases. Knowledge and Information Systems (KAIS), 2020
- Botao Peng, Panagiota Fatourou, Themis Palpanas. SING: Sequence Indexing Using GPUs. ICDE, 2021
- Georgios Chatzigeorgakidis, Dimitrios Skoutas, Kostas Patroumpas, Themis Palpanas, Spiros Athanasiou, Spiros Skiadopoulos: Twin Subsequence Search in Time Series. EDBT 2021
- Botao Peng, Panagiota Fatourou, Themis Palpanas. Fast Data Series Indexing for In-Memory Data. VLDBJ 2021

References (time series management systems)

- InfluxDB: https://www.influxdata.com/
- Timescale: https://www.timescale.com
- Beringei: https://github.com/facebookarchive/beringei
- Druid: https://druid.apache.org
- Prometheus: https://Prometheus.io
- CrateDB: https://crate.io
- IoTDb: https://iotdb.apache.org
- OpenTSDB: http://opentsdb.net/
- QuasarDB: https://www.quasardb.net/
- Timestream: https://aws.amazon.com/timestream/
- Apache IoTDB: https://iotdb.apache.org/
- nestor: http://nestordb.com/