Scalable Analytics on Large Sequence Collections

Karima Echihabi

Mohammed VI Polytechnic University

Themis Palpanas

Université Paris Cité & French University Institute (IUF)

International Conference on Mobile Data Management (MDM), June 2022

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 indices for similarity search?
- can such indices help with geolocated data series analysis?

- how can these indices parallelize/distribute their operations?
- can these indexes be used for general high-d vector similarity search?

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

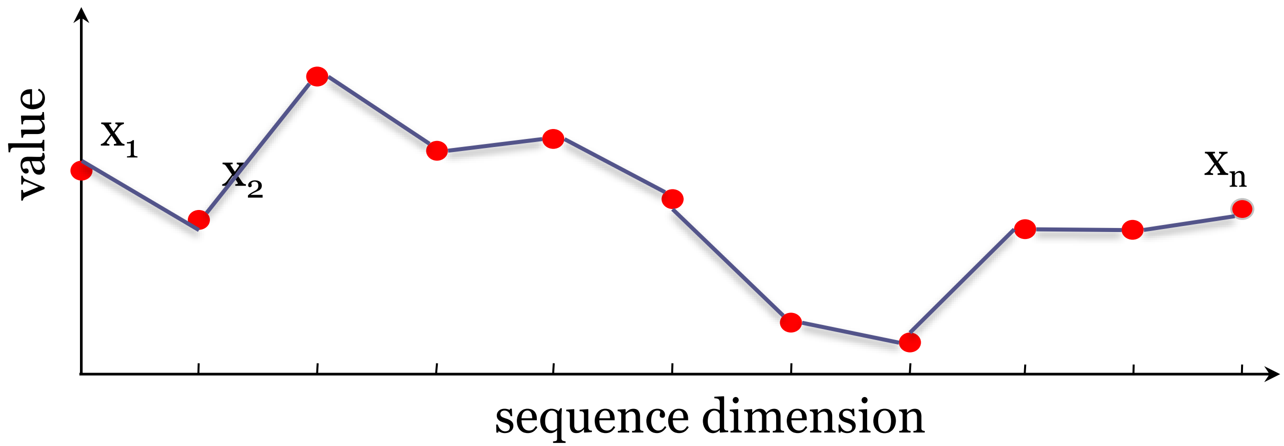
Acknowledgements

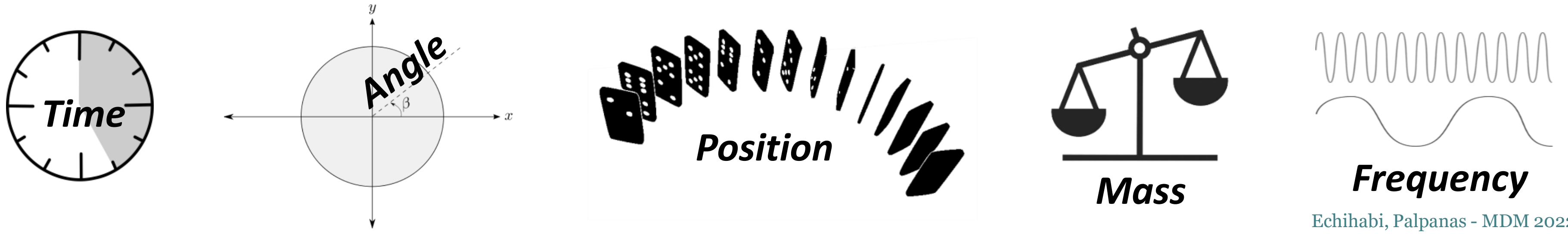
- thanks for slides to
  - Michail Vlachos
  - Panagiotis Papapetrou
  - George Kollios
  - Dimitrios Gunopulos
  - Christos Faloutsos
  - Panos Karras
  - Peng Wang
  - Liang Zhang
  - Reza Akbarinia
  - Georgios Chatzigeorgakidis

Introduction, Motivation

Data series

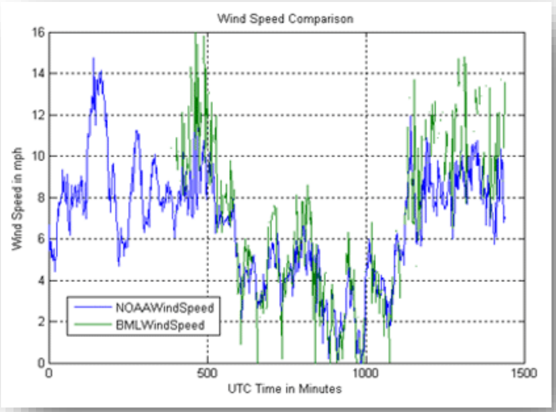
- Sequence of points ordered along some dimension





Scientific Monitoring

- meteorology, oceanography, astronomy, finance, sociology, ...

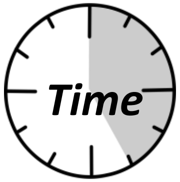


Wind speed

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

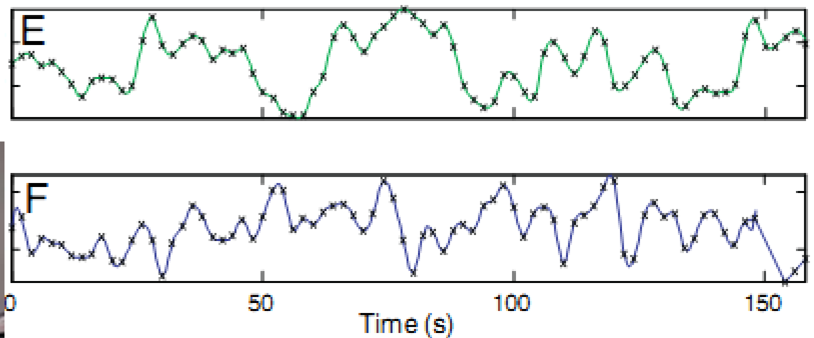
Historical stock quotes

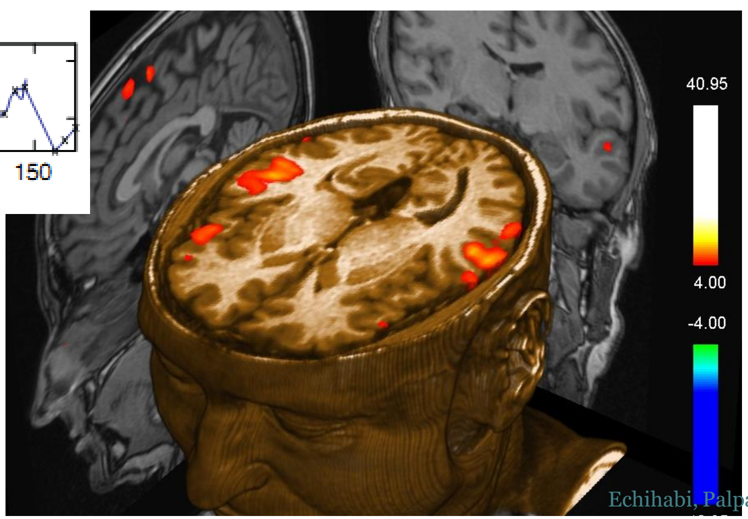
http://money.cnn.com/2012/04/23/markets/walmart_stock/index.htm

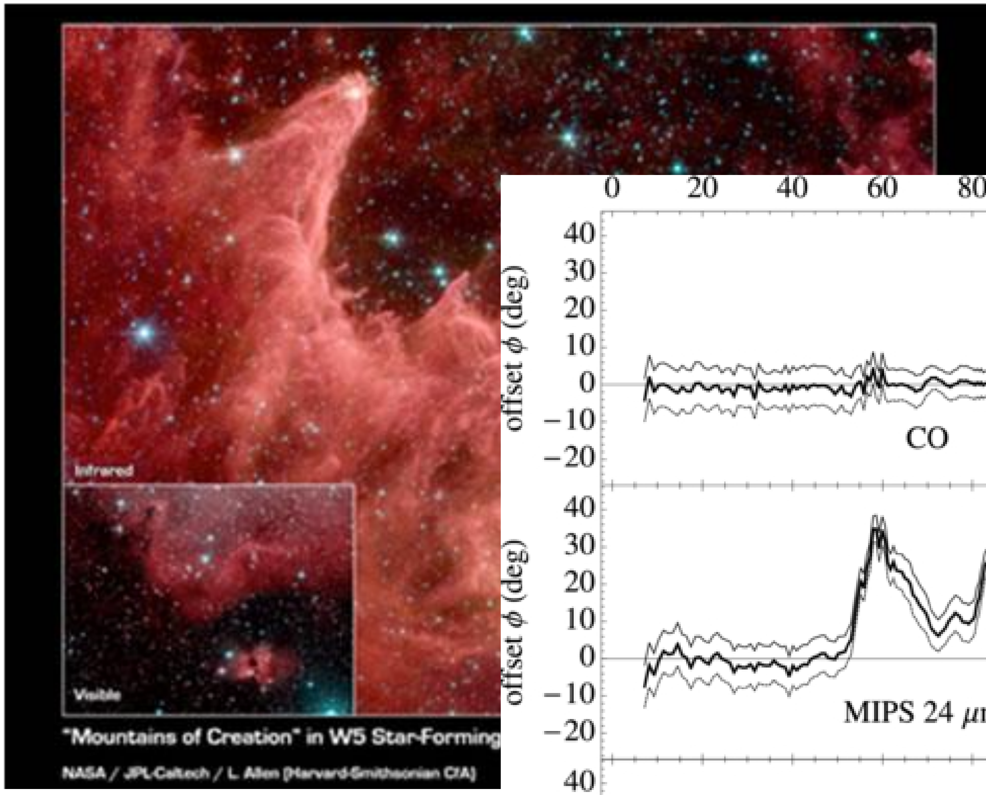


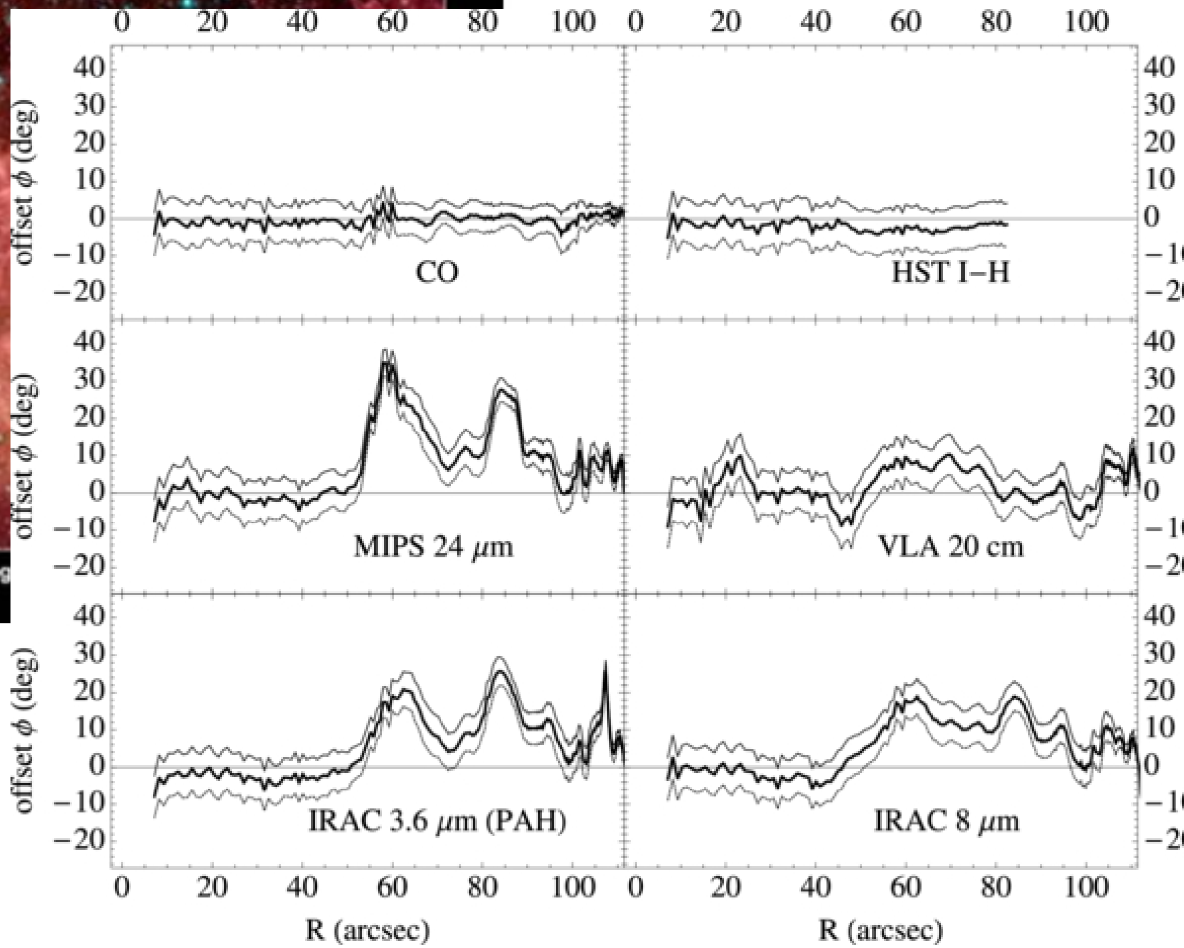
Neuroscience

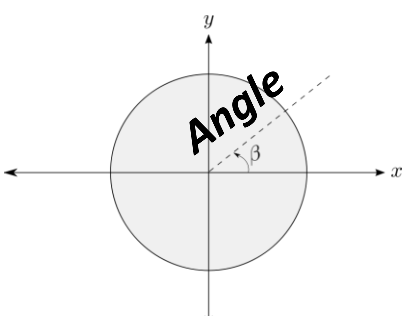
- functional Magnetic Resonance Imaging (fMRI) data
  - primary experimental tool of neuroscientists
  - reveal how different parts of brain respond to stimuli





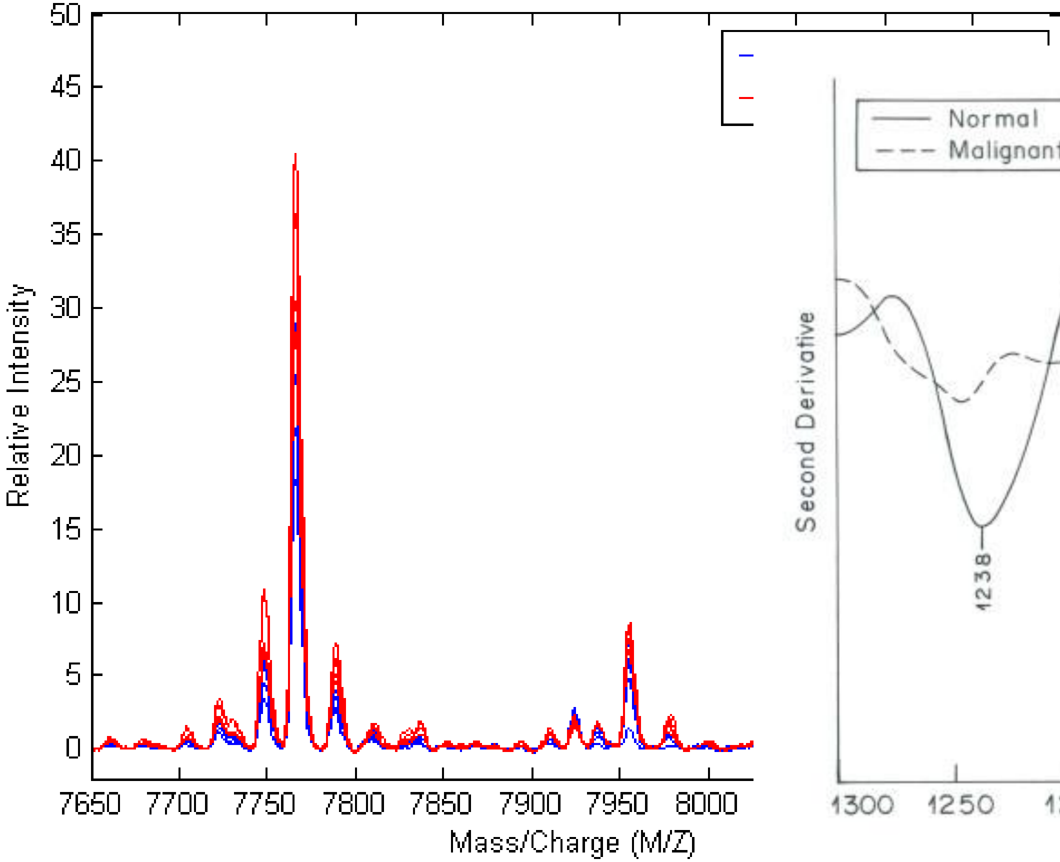


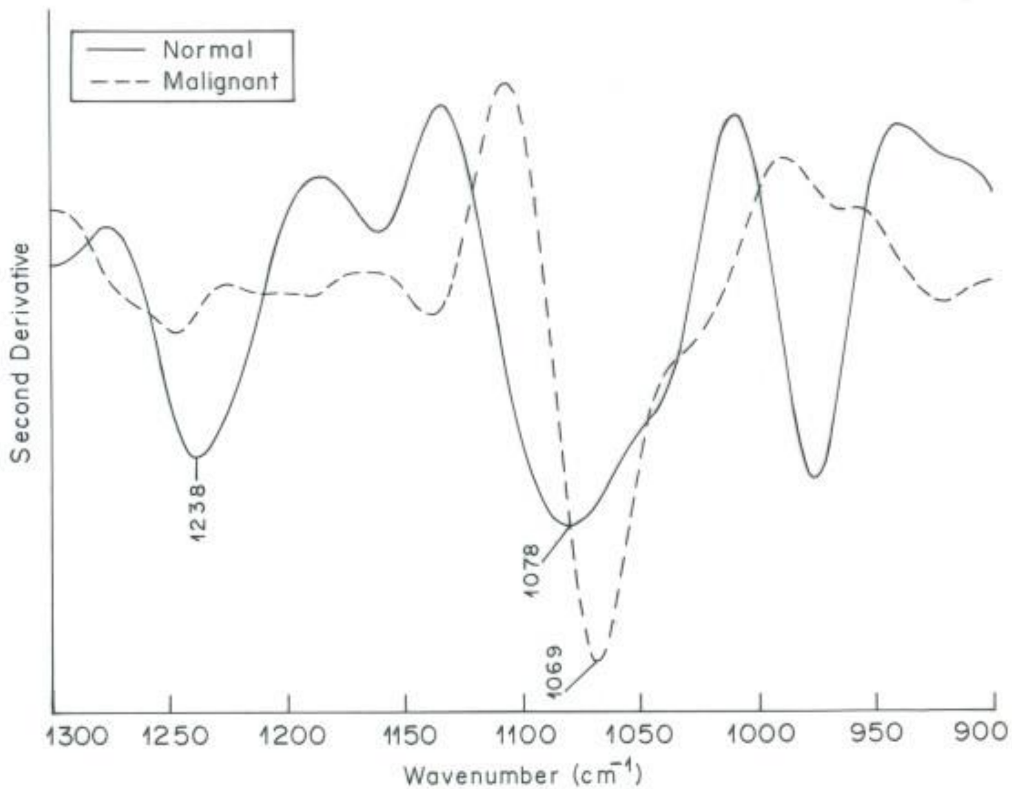




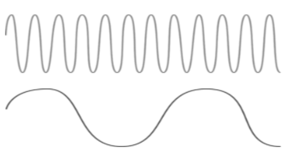
Schinnerer et al.

Medicine





Mass



Frequency

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

Analysis Tasks: Simple Query Answering

select values in time interval

select values in some range

select some data series

combinations of those

Analysis Tasks: Complex Analytics

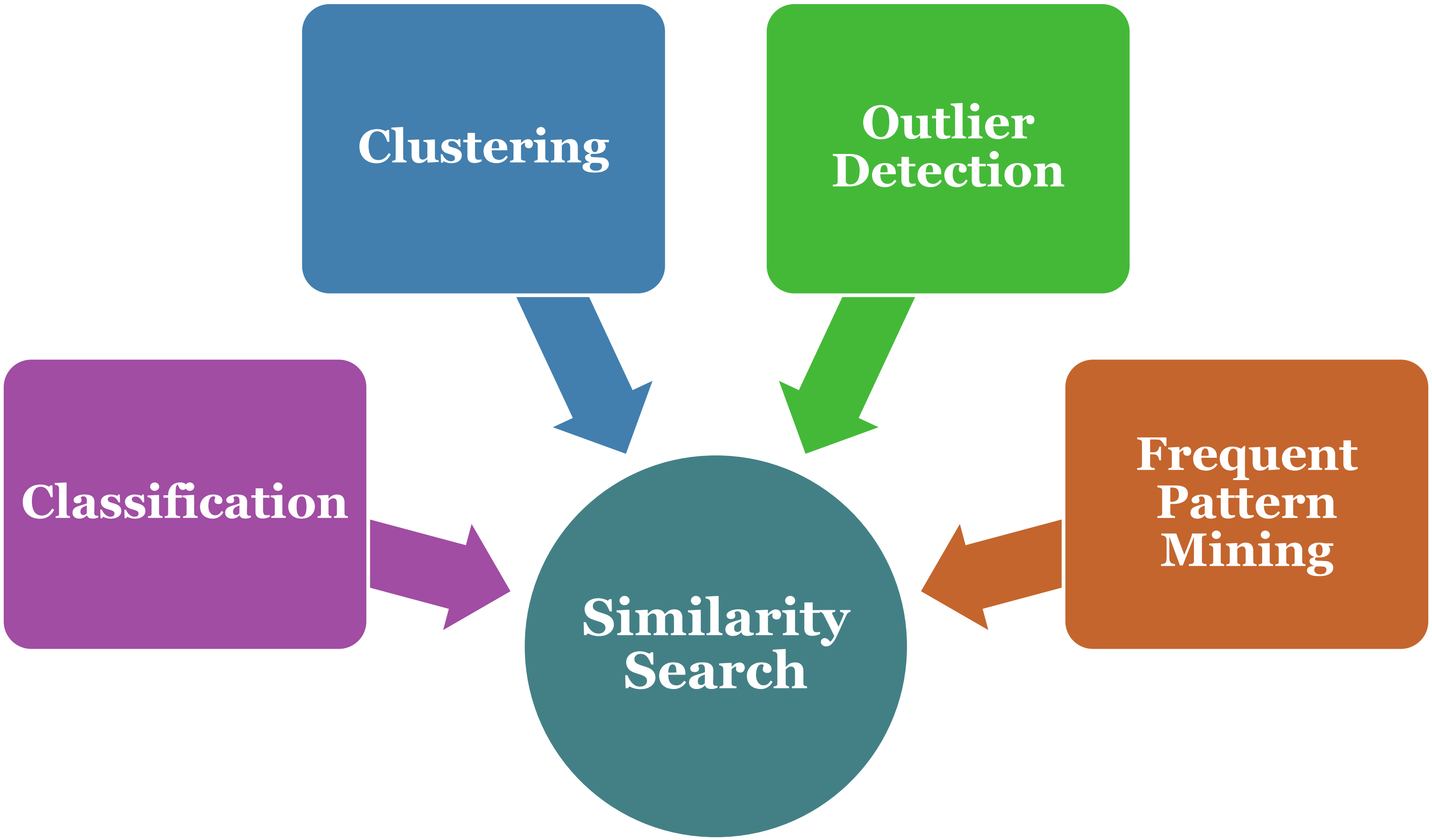
Clustering

Outlier Detection

Classification

Frequent Pattern Mining

Analysis Tasks: Complex Analytics



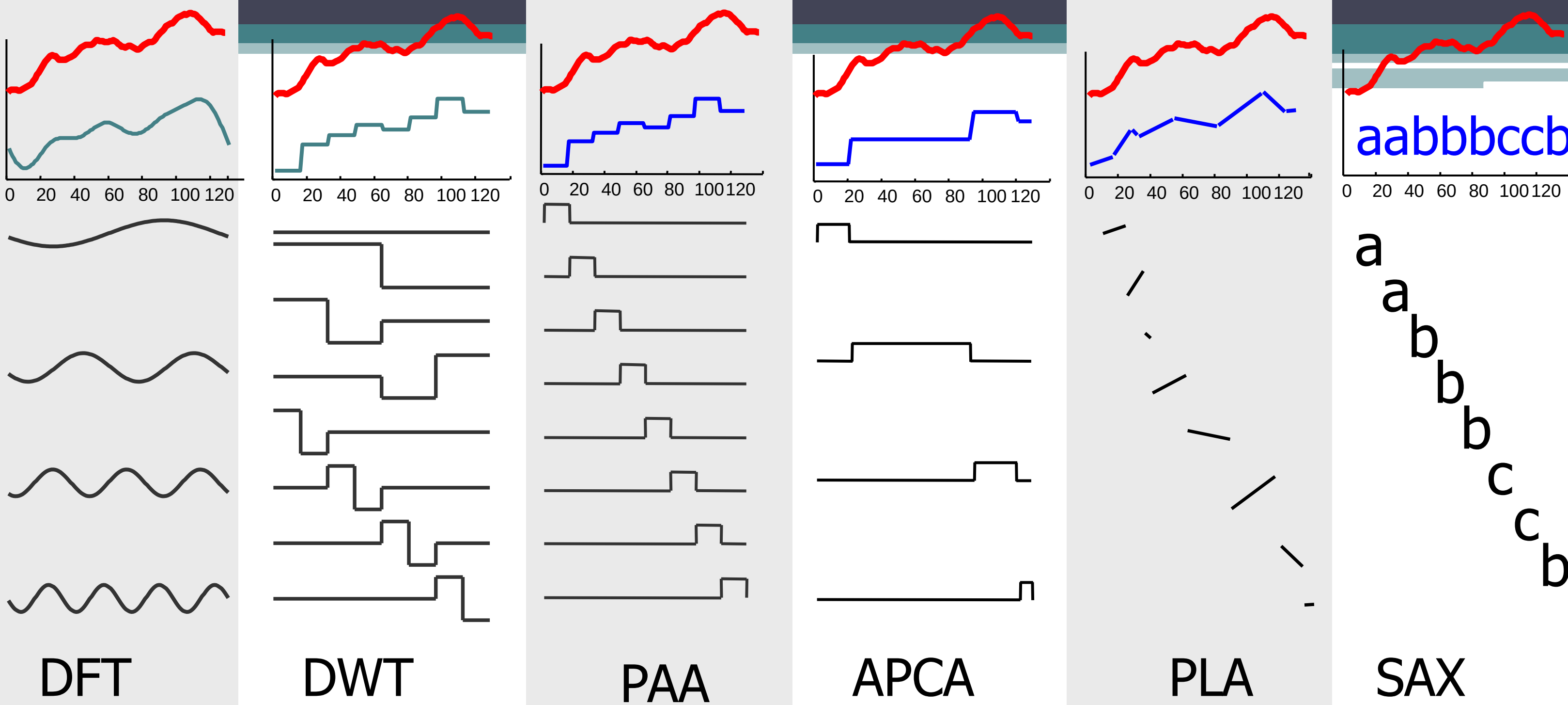
Analysis Tasks: Complex Analytics

Clustering

Outlier Detection

HARD, because of very high dimensionality: each data series has 100s-1000s of points!

even HARDER, because of very large size: millions to billions of data series (multi-TBs)!



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

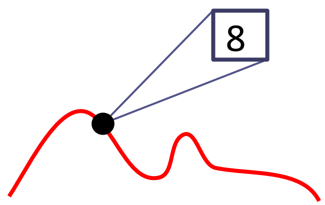
Comparison of Representations

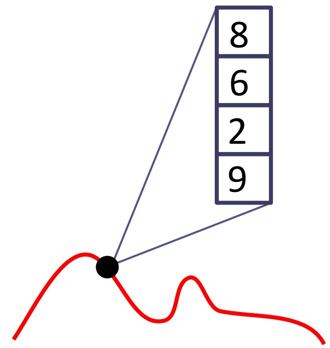
- which representation is the best?
- depends on data characteristics
  - periodic, smooth, spiky, ...
- overall (averaged over many diverse datasets, using same memory budget), when measuring reconstruction error (RMSE)
  - no big differences among methods
  - DFT, PAA, DWT (Haar), iSAX slightly better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

Data Series Similarity Problem Variations

Problem Variations

Series





<u>Univariate</u>

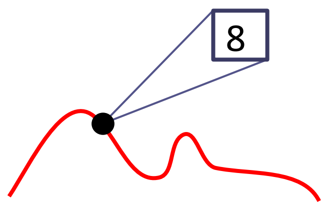
each point represents one value (e.g., temperature)

<u>Multivariate</u>

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

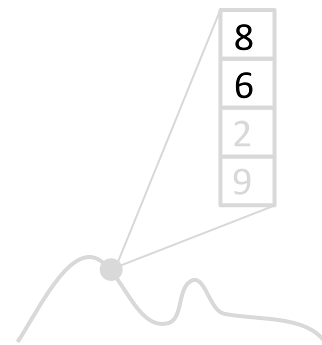
Problem Variations

Series



Univariate

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



Multivariate

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

Problem Variations

Distance Measures

Publications

Ding-PVLDB'08

Paparrizos-SIGMOD'20

- 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

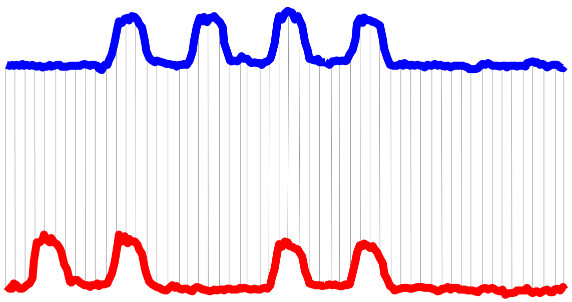
Problem Variations

Distance Measures

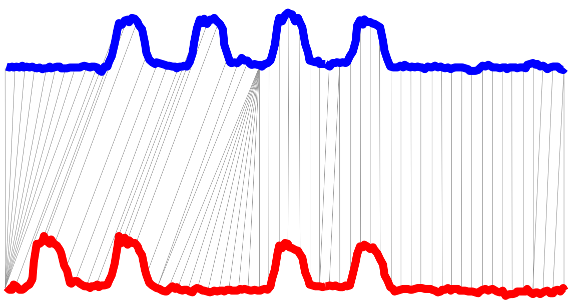
- similarity search is based on measuring distance between sequences
- dozens of distance measures have been proposed
  - lock-step
    - Minkowski, Manhattan, Euclidean, Maximum, DISSIM, ...
  - sliding
    - Normalized Cross-Correlation, SBD, ...
  - elastic
    - DTW, LCSS, MSM, EDR, ERP, Swale, ...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

Distance Measures: LCSS against Euclidean, DTW

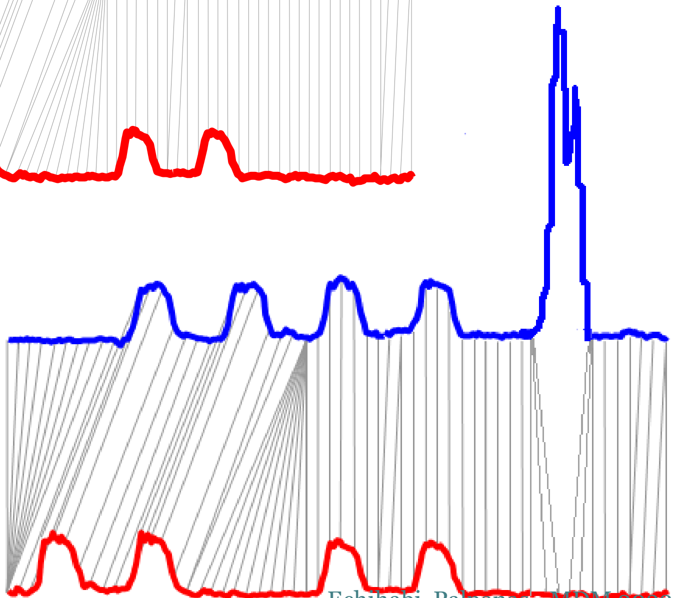
- Euclidean
  - rigid



- Dynamic Time Warping (DTW)
  - allows local scaling

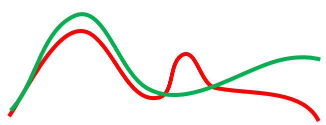


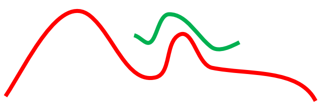
- Longest Common SubSequence (LCSS)
  - allows local scaling
  - ignores outliers



Problem Variations

Queries





Whole matching

Entire query

Entire candidate

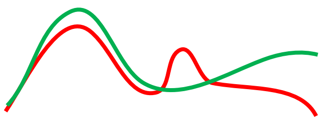
Subsequence matching

Entire query

A subsequence of a candidate

Problem Variations

Queries



<u>Whole matching</u>

Entire query

Entire candidate

<u>Subsequence matching</u>

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
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

Problem Variations

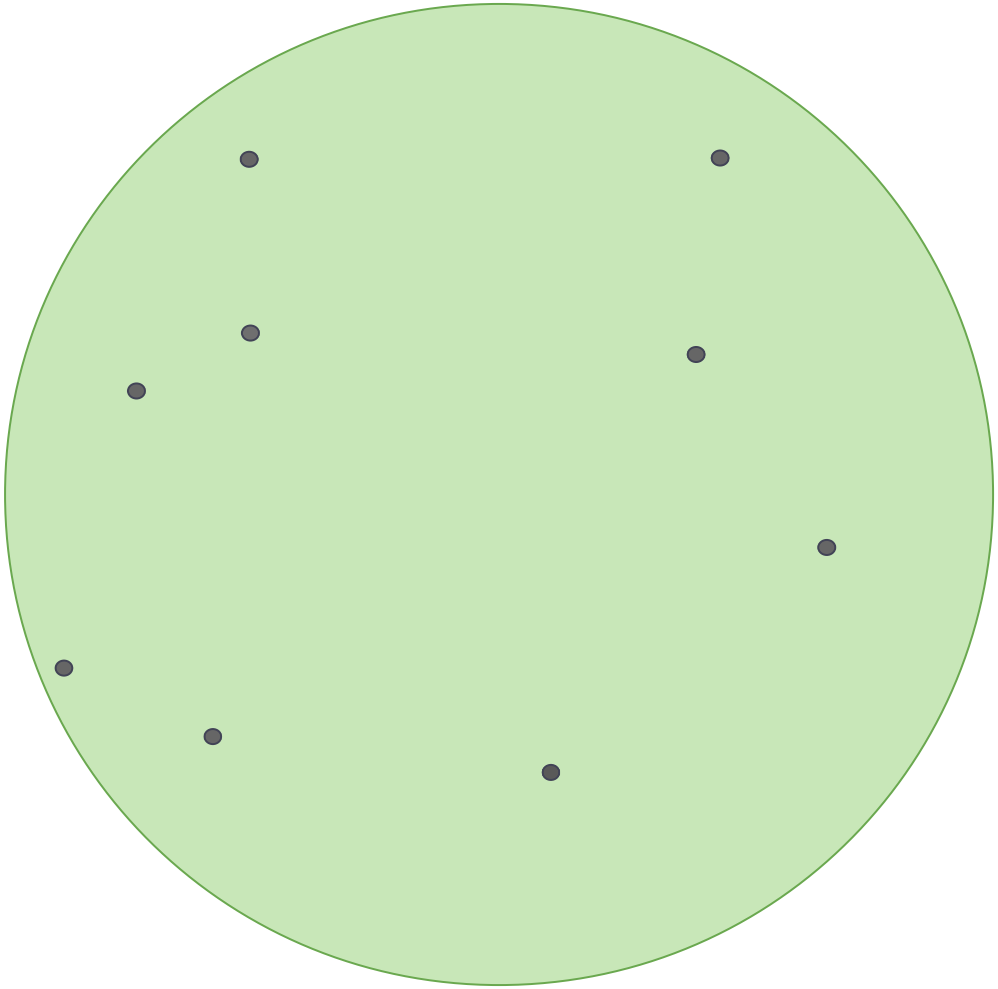
Queries

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

Nearest Neighbor (NN) Queries…

Publications

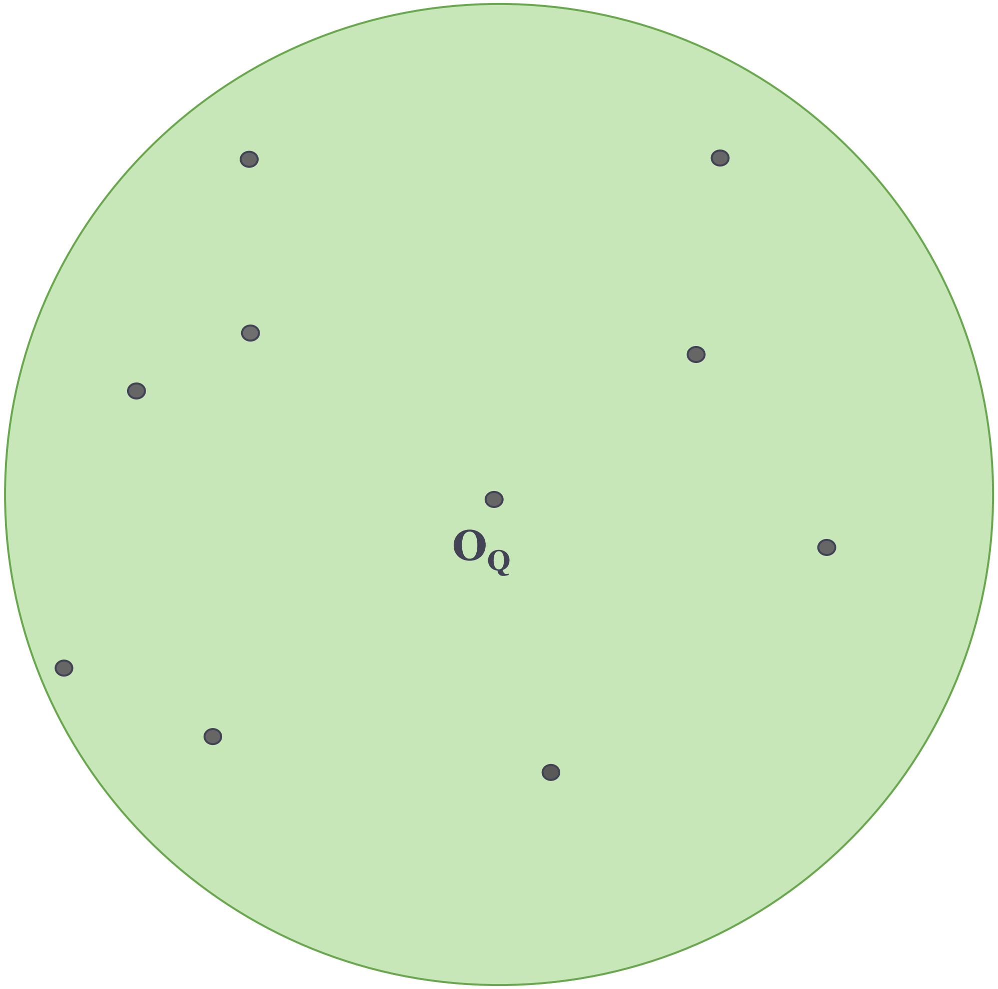
Echihabi et al. PVLDB‘19



Nearest Neighbor (NN) Queries…

Publications

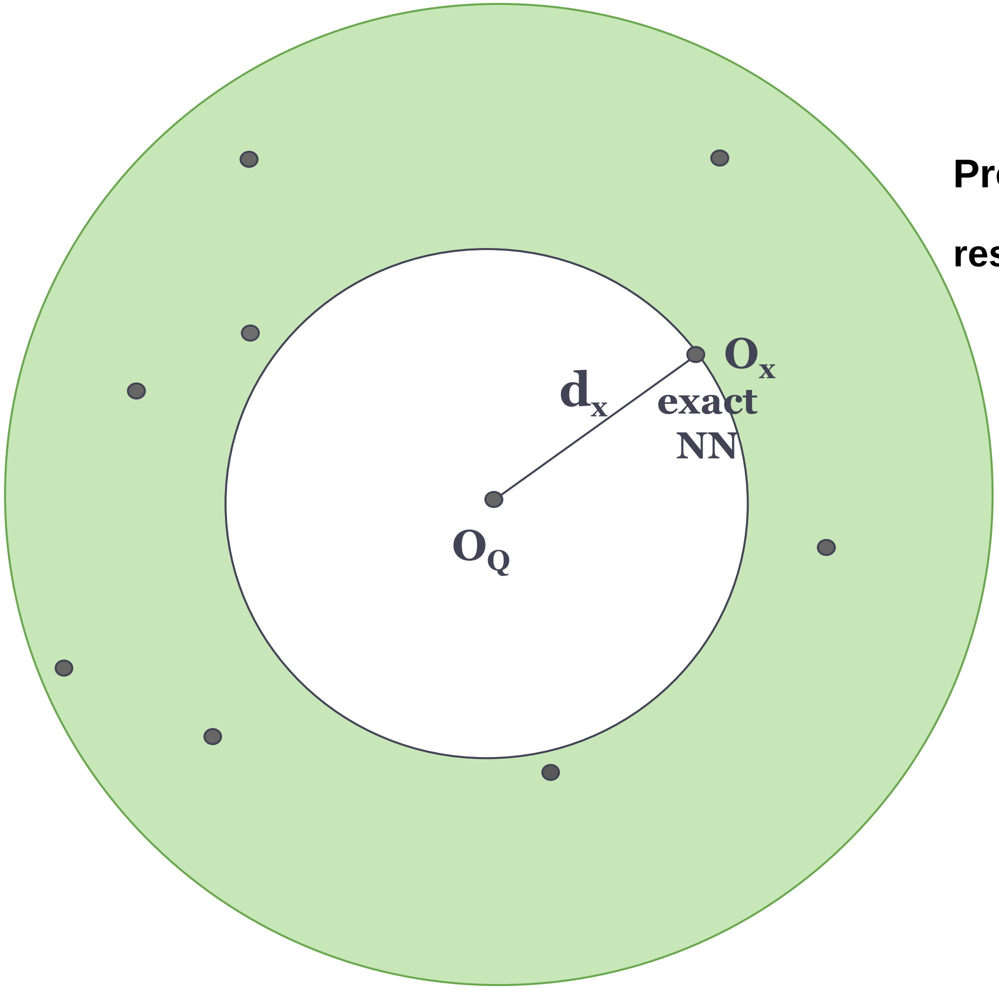
Echihabi et al. PVLDB'19



Nearest Neighbor (NN) Queries…

Publications

Echihabi et al. PVLDB‘19



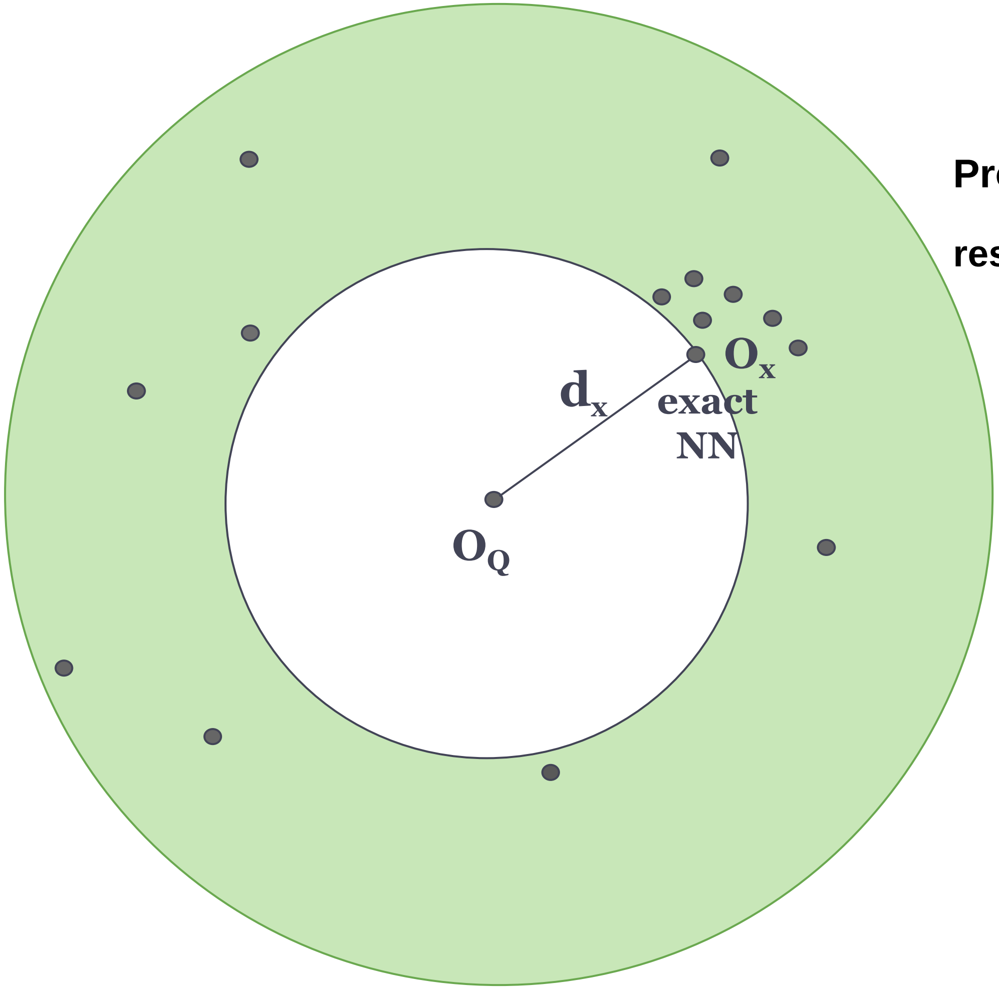
Prob(d_x = min{d_i}) = 1

result is exact NN

Nearest Neighbor (NN) Queries…

Publications

Echihabi et al. PVLDB'19



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

result is exact NN

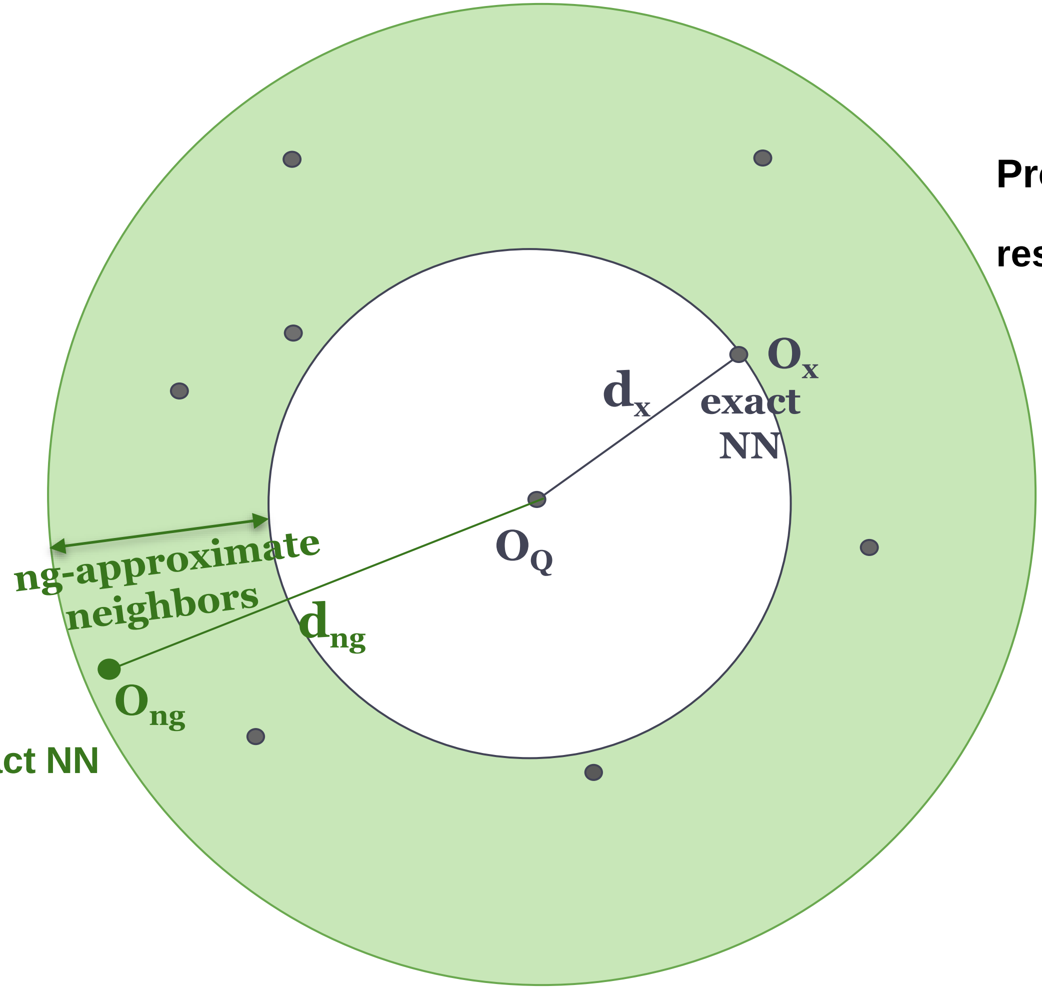
Nearest Neighbor (NN) Queries…

Publications

Echihabi et al. PVLDB'19

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

result is exact NN



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

result within ? of exact NN

Nearest Neighbor (NN) Queries…

Publications

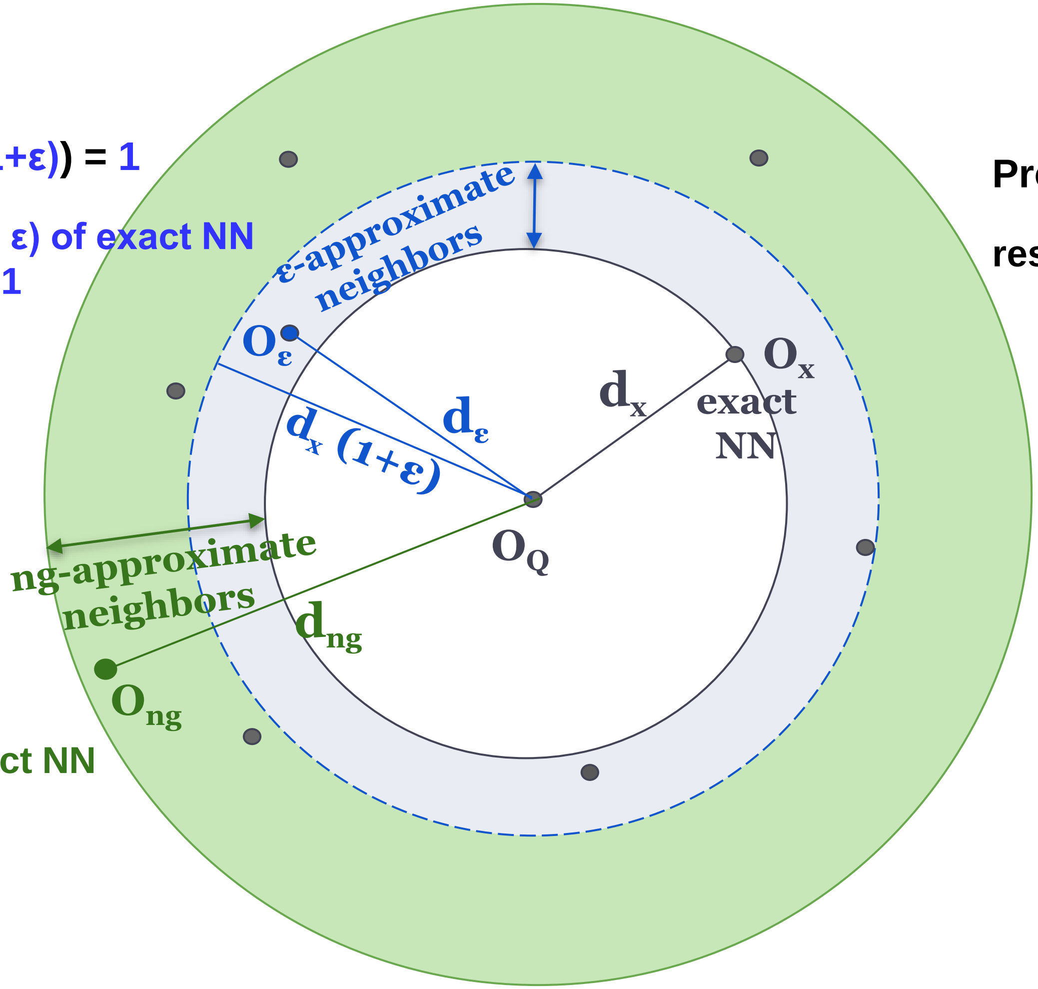
Echihabi et al. PVLDB‘19

Prob(d_ε <= d_x (1+ε)) = 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

Nearest Neighbor (NN) Queries…

Publications

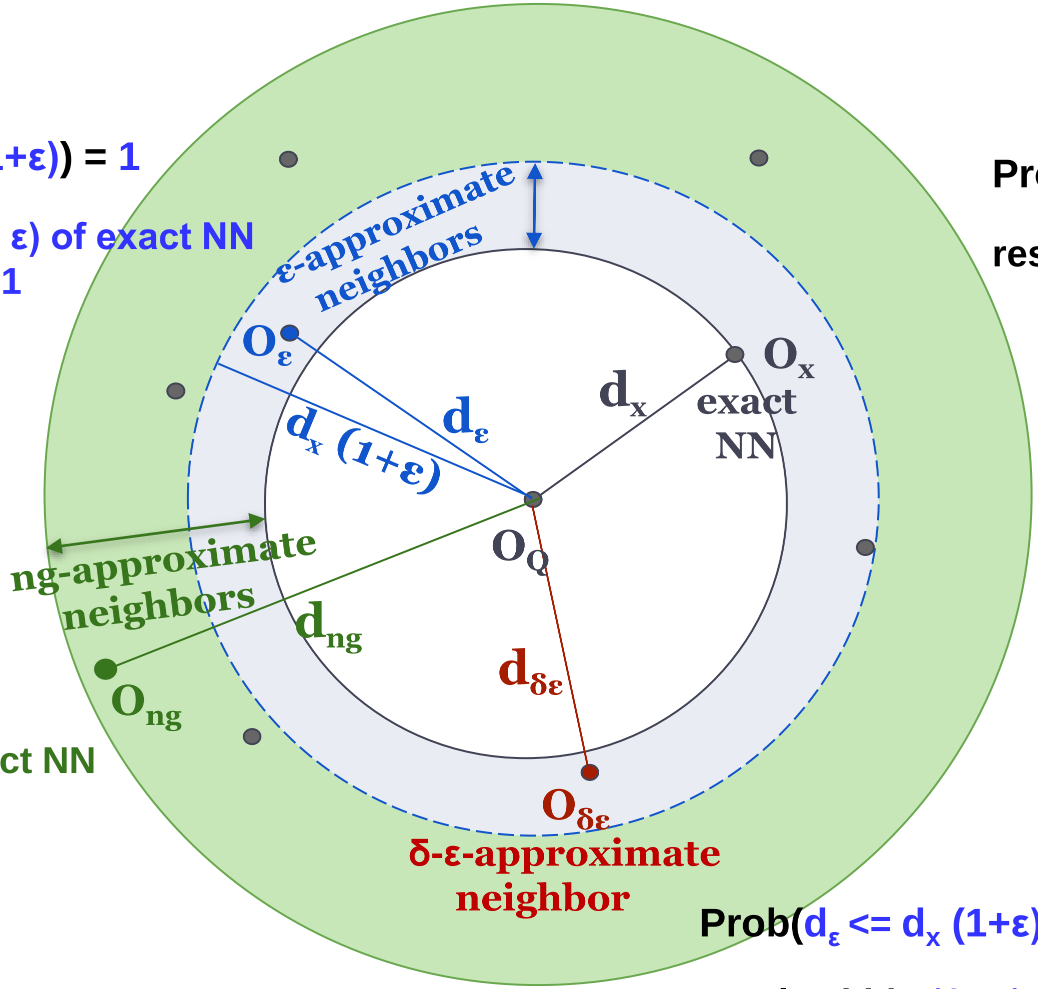
Echihabi et al. PVLDB'19

Prob(d_ε <= d_x (1+ε)) = 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_ε <= d_x (1+ε)) >= δ

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

Nearest Neighbor (NN) Queries…

Publications

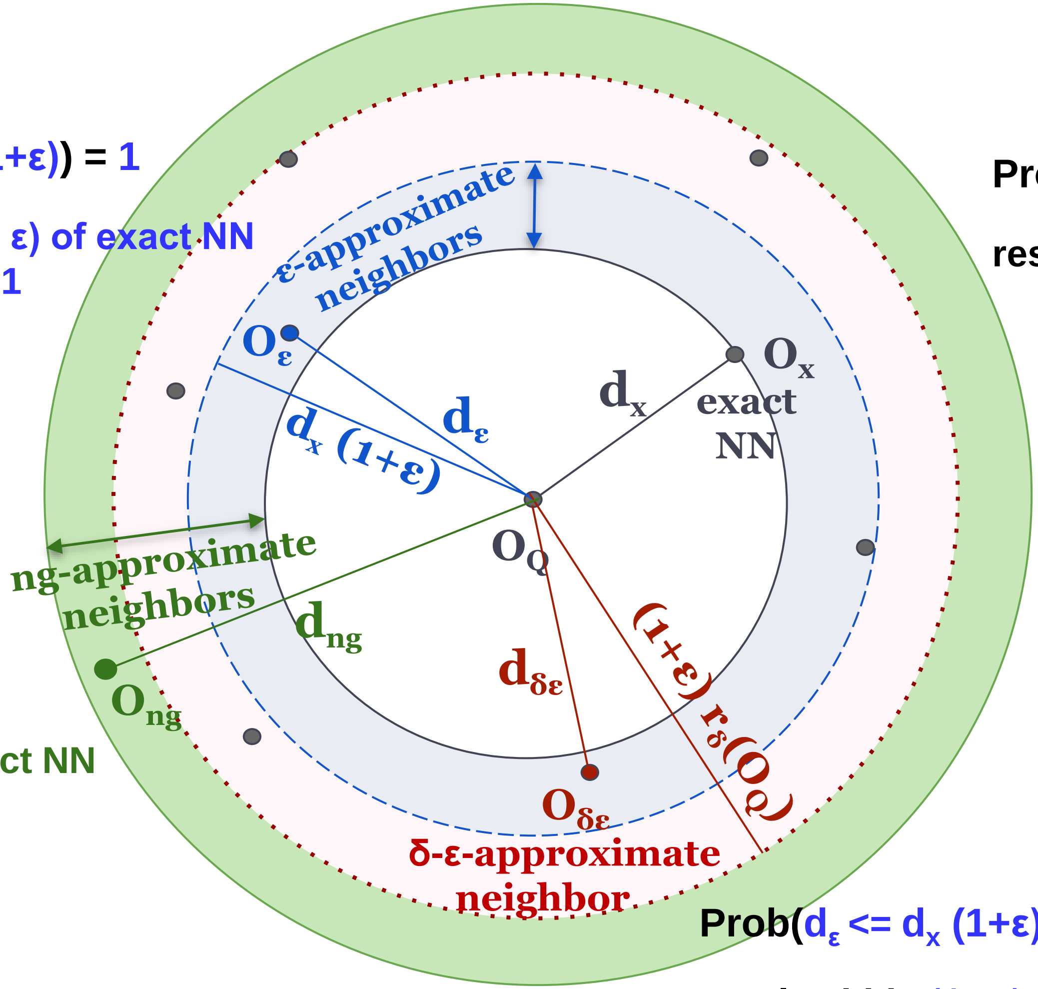
Echihabi et al. PVLDB'19

Prob(d_ε <= d_x (1+ε)) = 1

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

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

result is exact NN



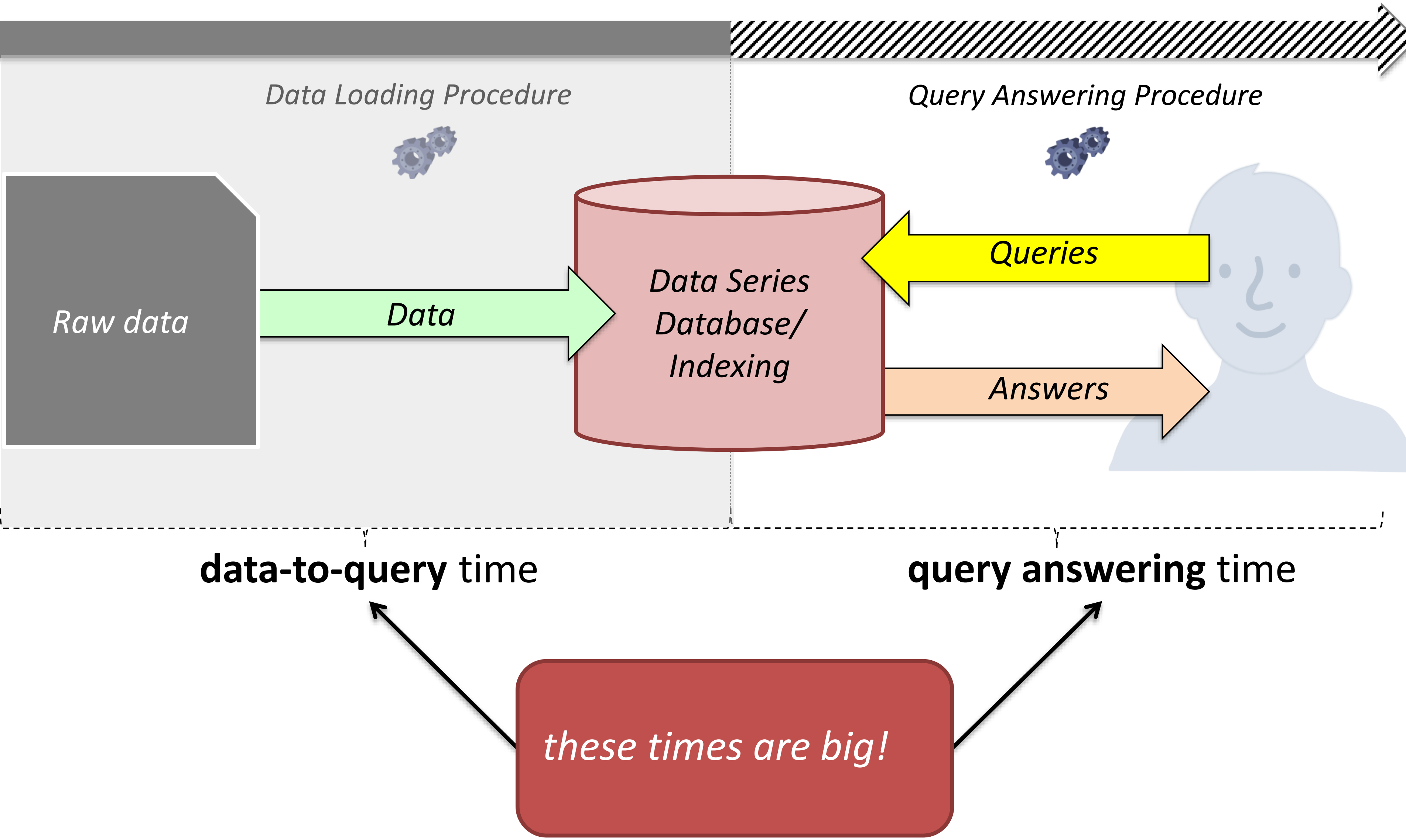
Prob(d_{ng} <>= ?) = ?

result within ? of exact NN

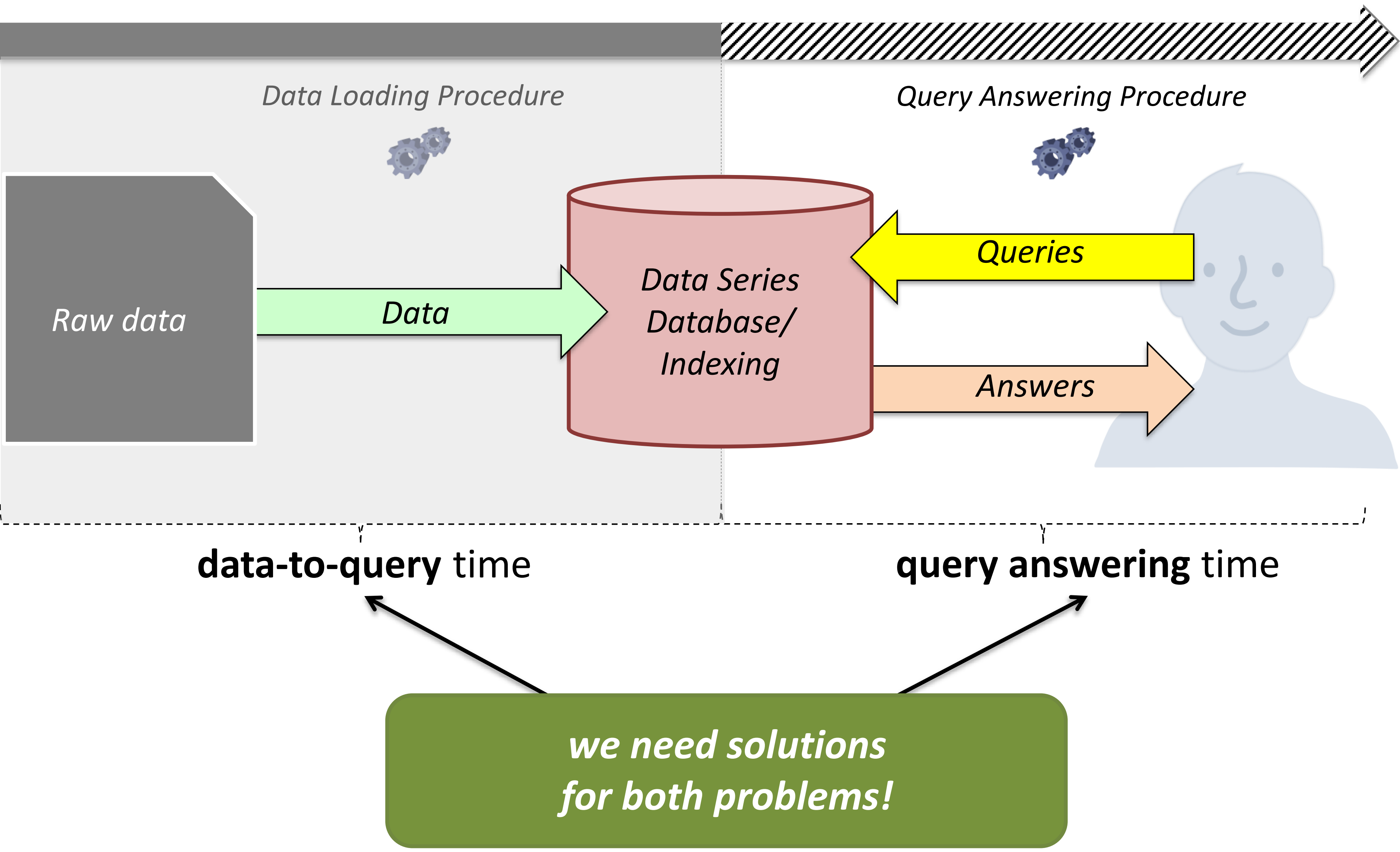
Prob(d_ε <= d_x (1+ε)) >= δ

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

Query answering process



Query answering process



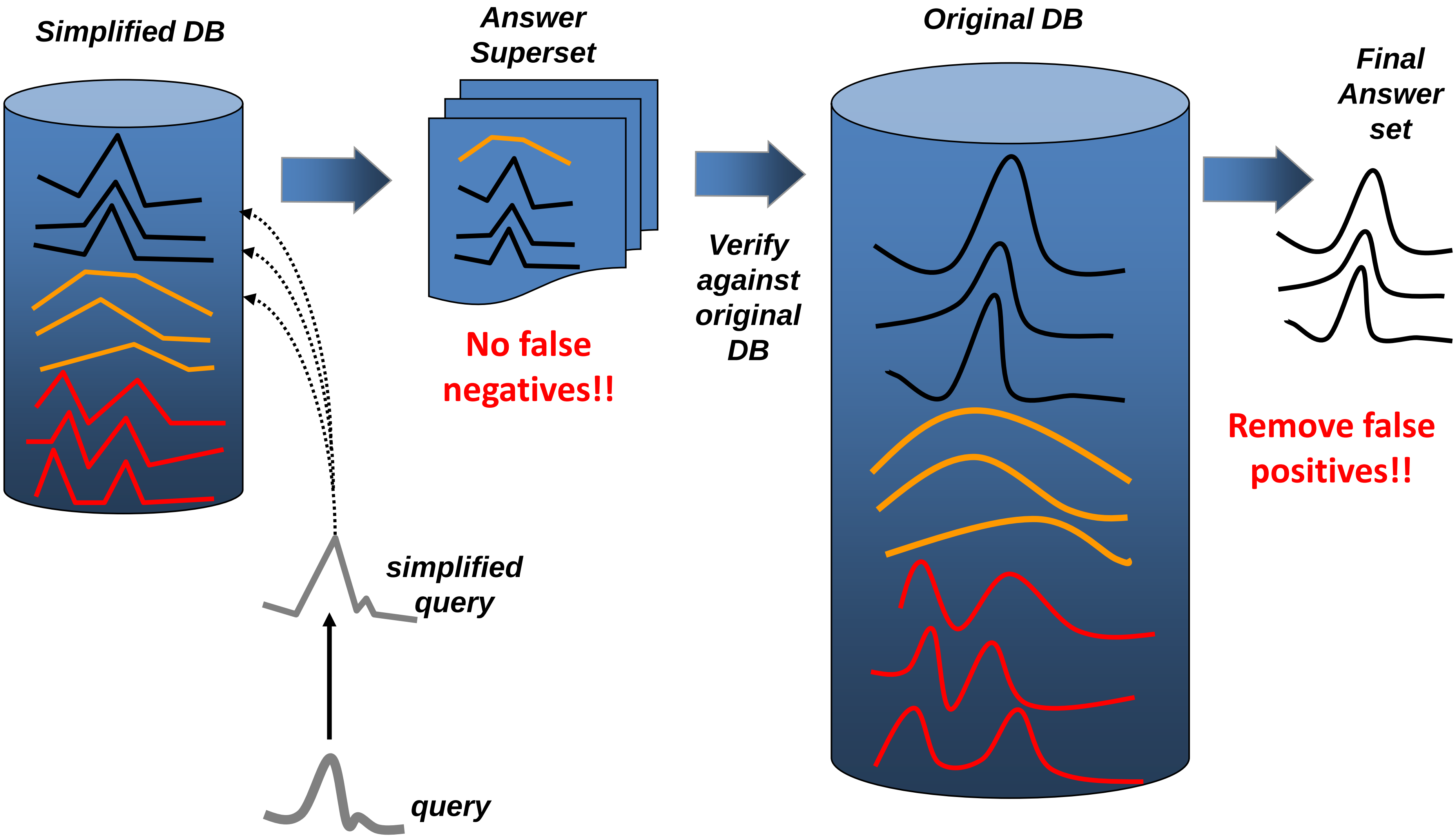
GEMINI Framework

Publications

Faloutsos-SIGMOD'94

- 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

Generic Search using Lower Bounding



GEMINI: contractiveness

Publications

Faloutsos-SIGMOD'94

- GEMINI works when:

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

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

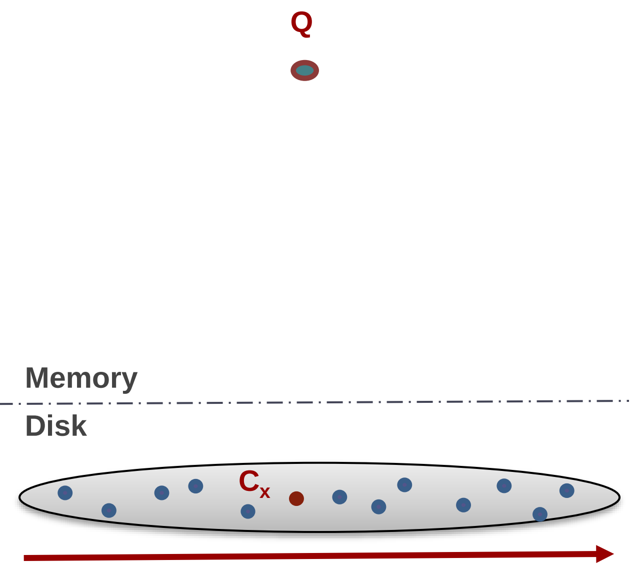
Questions?

Similarity Search Classes of Methods

Similarity Search
Classes of Methods

Exact Search

Similarity Matching
Serial Scan



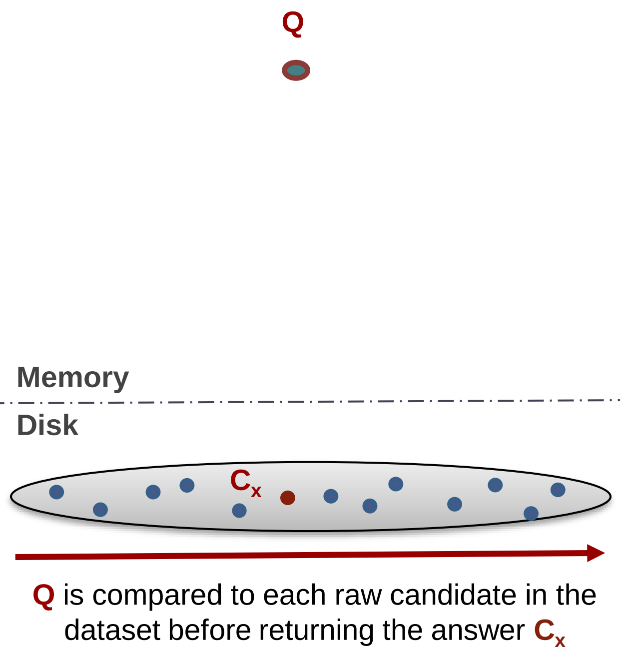
Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching
Serial Scan

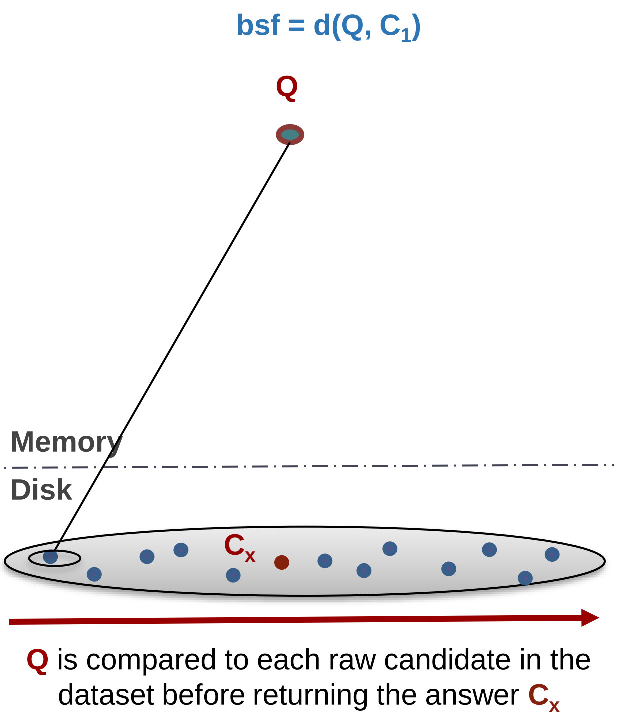
bsf = +∞



(a) Serial scan

Answering a similarity search query using different access paths

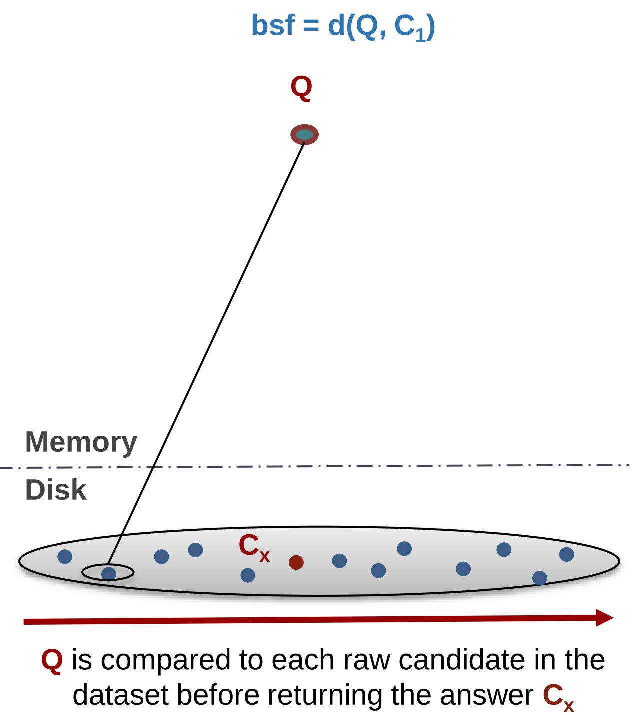
Similarity Matching
Serial Scan



(a) Serial scan

<u>**Answering a similarity search query using different access paths**</u>

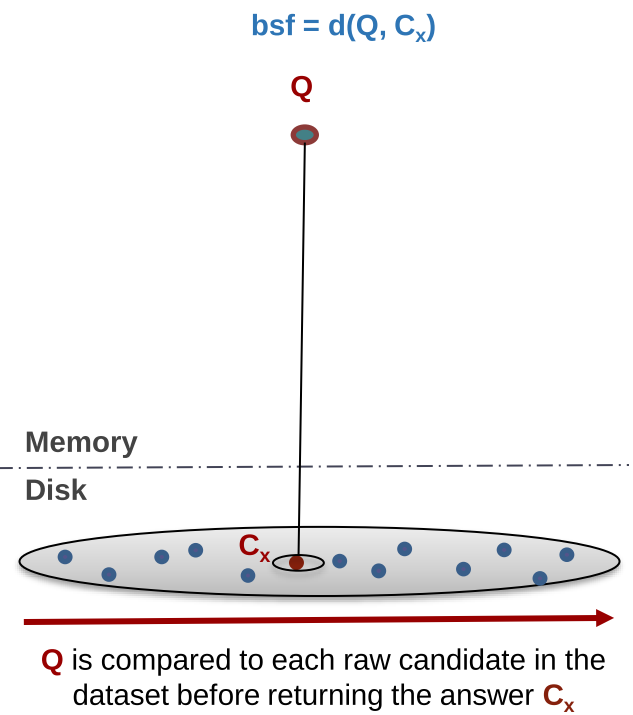
Similarity Matching
Serial Scan



(a) Serial scan

Answering a similarity search query using different access paths

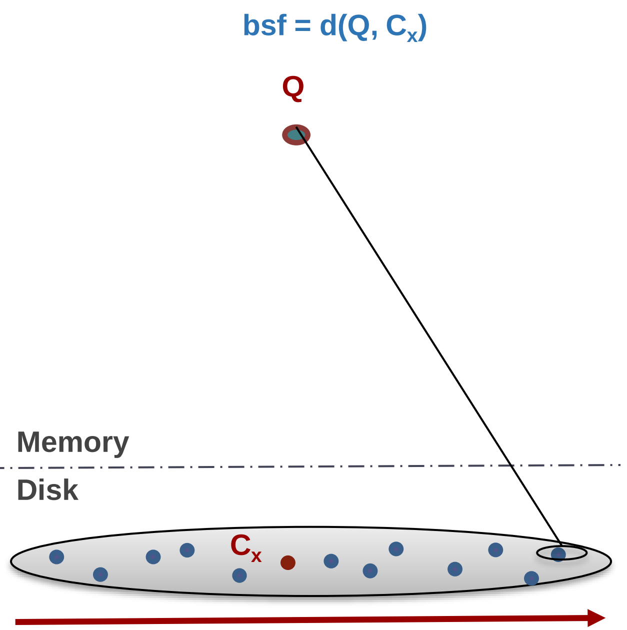
Similarity Matching Serial Scan



(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching
Serial Scan

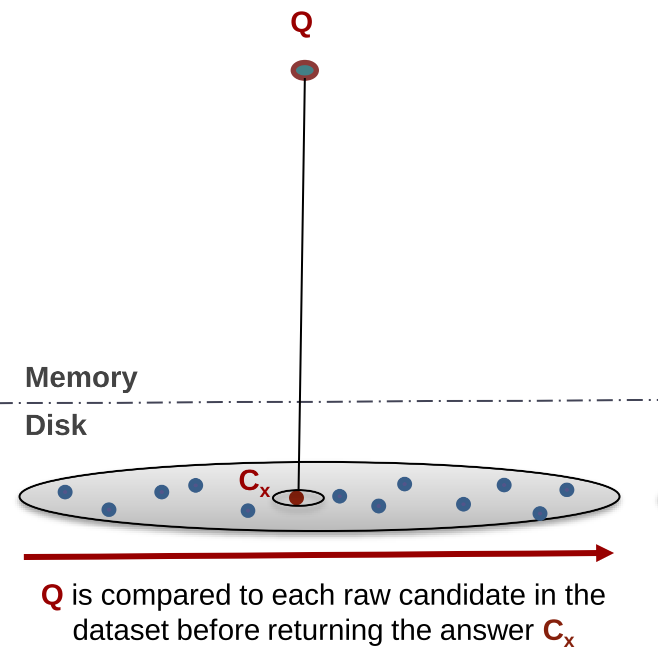


Q is compared to each raw candidate in the dataset before returning the answer **C_x**

(a) Serial scan

Answering a similarity search query using different access paths

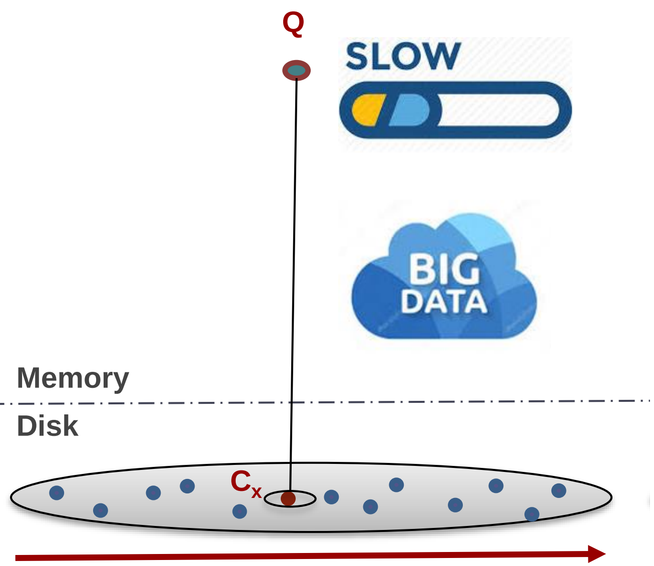
Similarity Matching
Serial Scan



(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching
Serial Scan

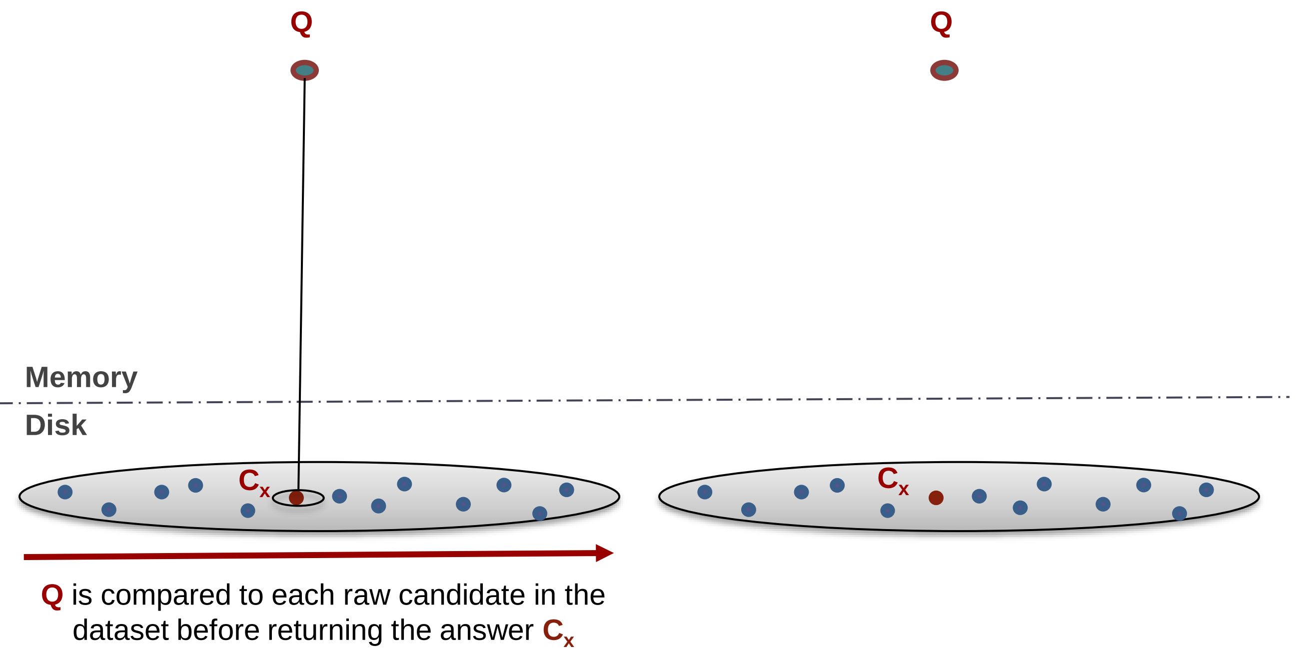


Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

Answering a similarity search query using different access paths

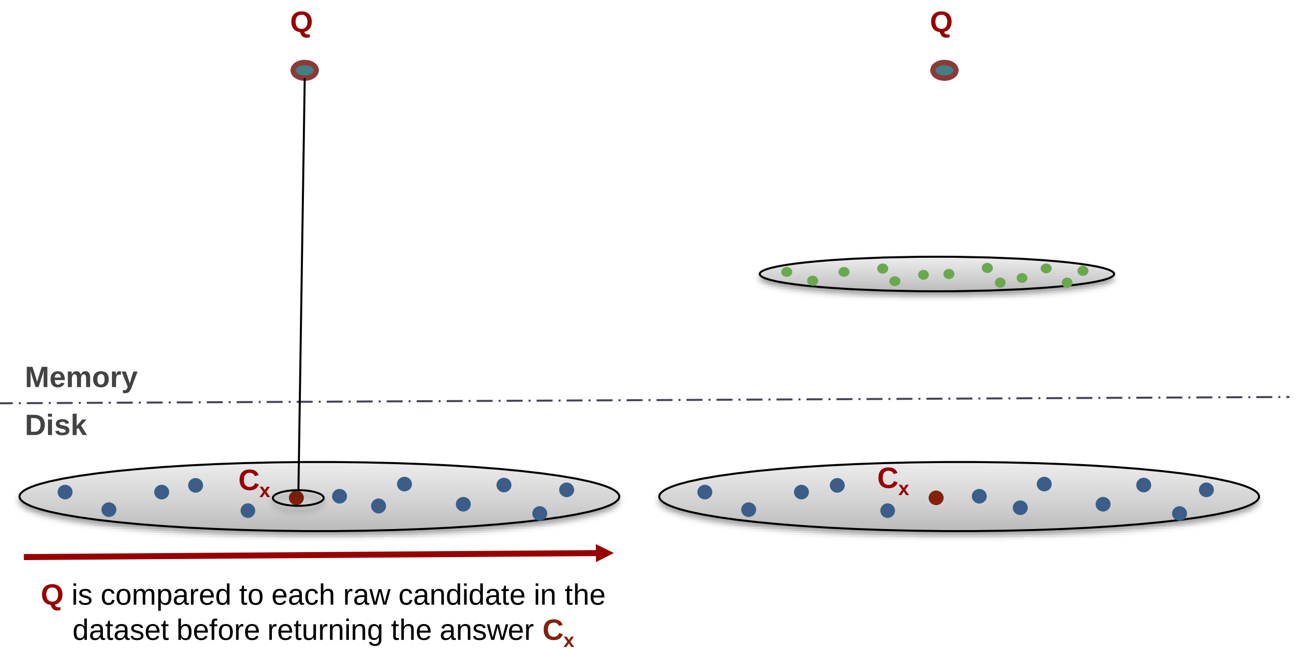
Indexes vs. Scans



(a) Serial scan **(b) Skip-sequential scan**

Answering a similarity search query using different access paths

Indexes vs. Scans

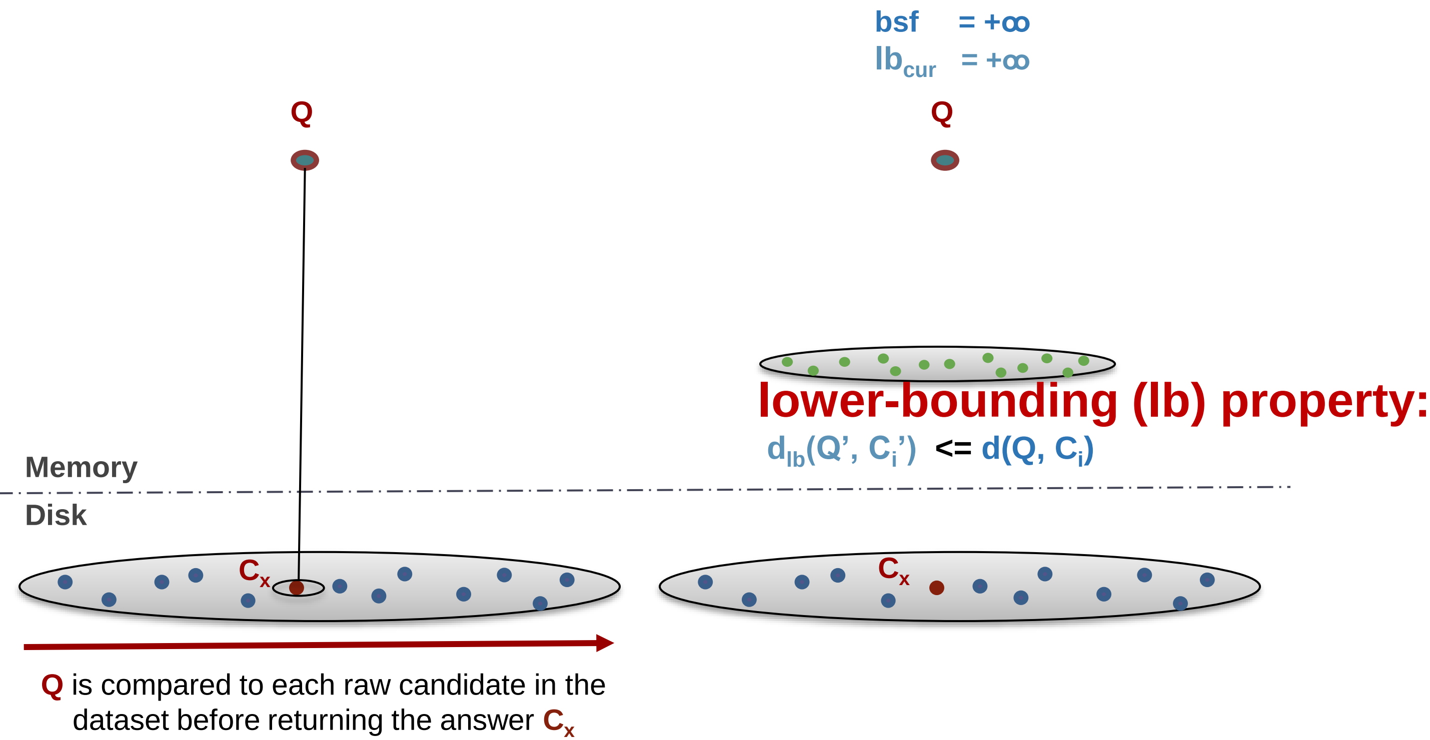


(a) Serial scan

(b) Skip-sequential scan

Answering a similarity search query using different access paths

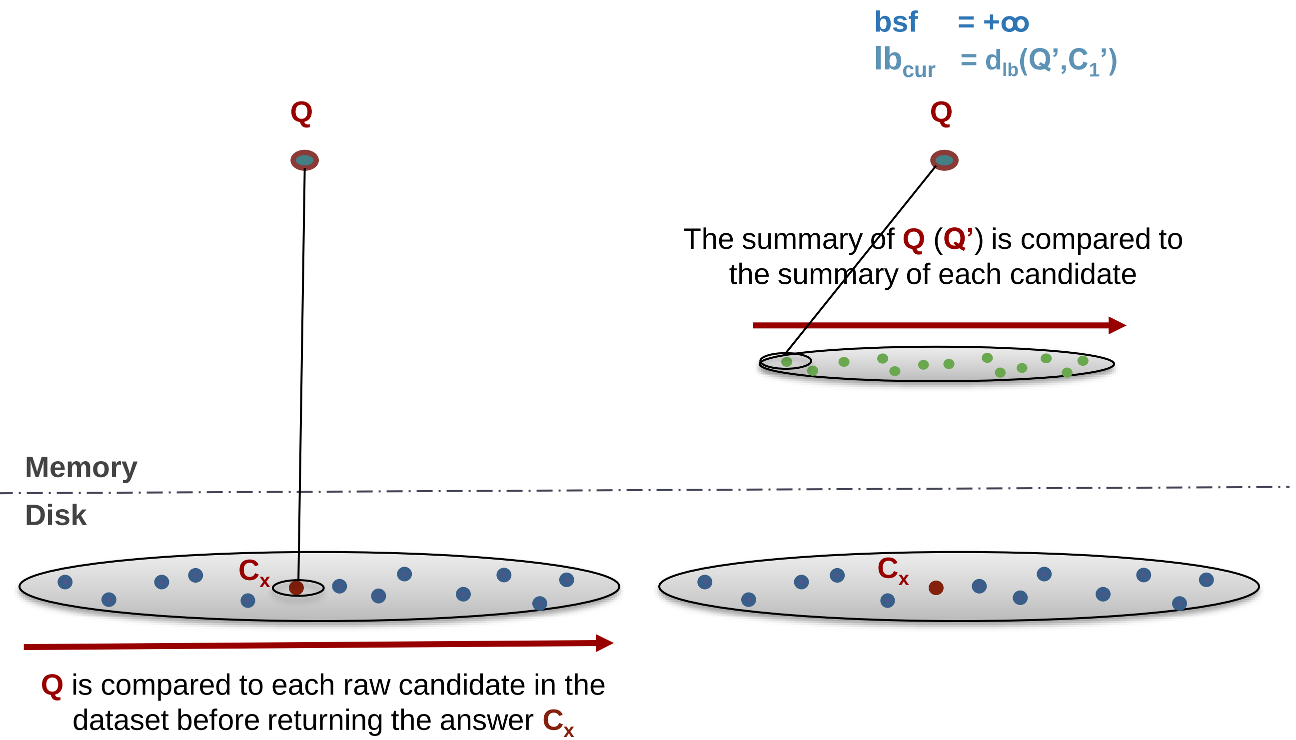
Indexes vs. Scans



(a) Serial scan (b) Skip-sequential scan

Answering a similarity search query using different access paths

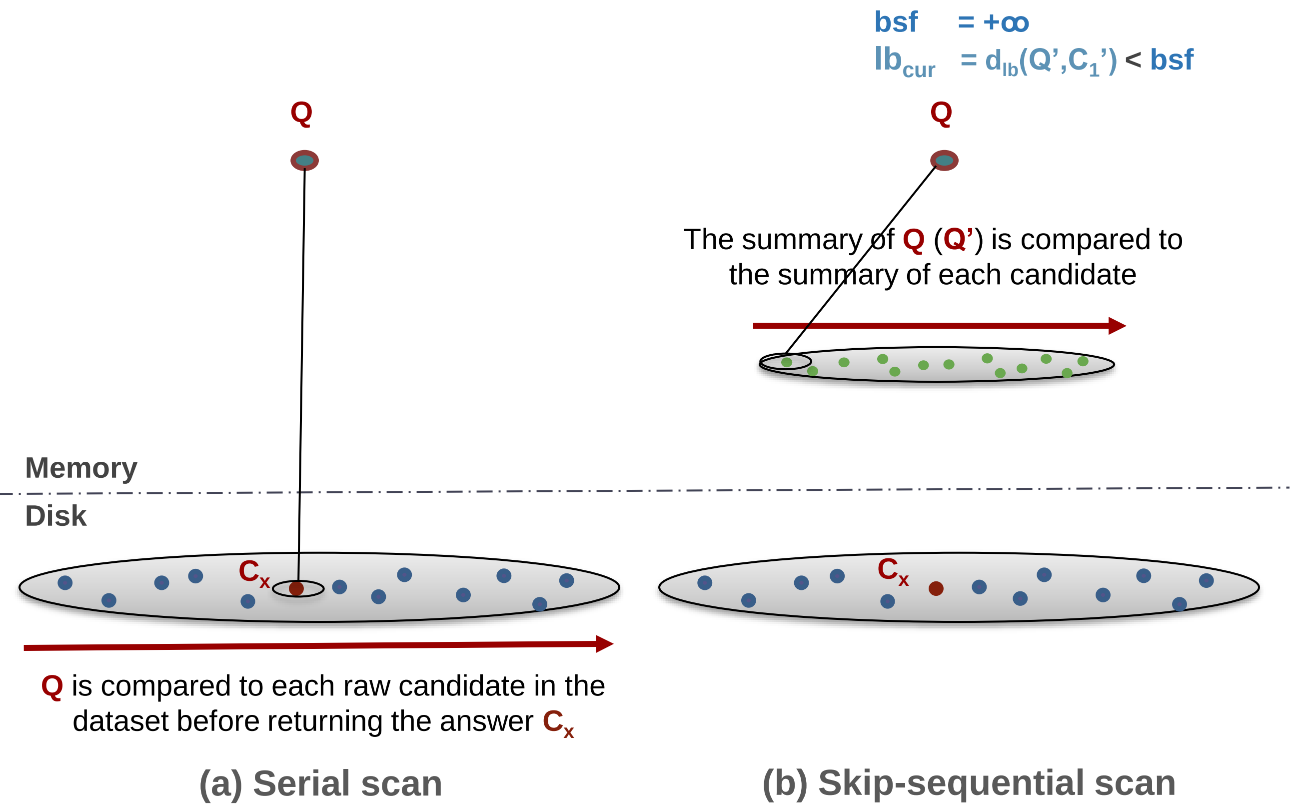
Indexes vs. Scans



(a) Serial scan **(b) Skip-sequential scan**

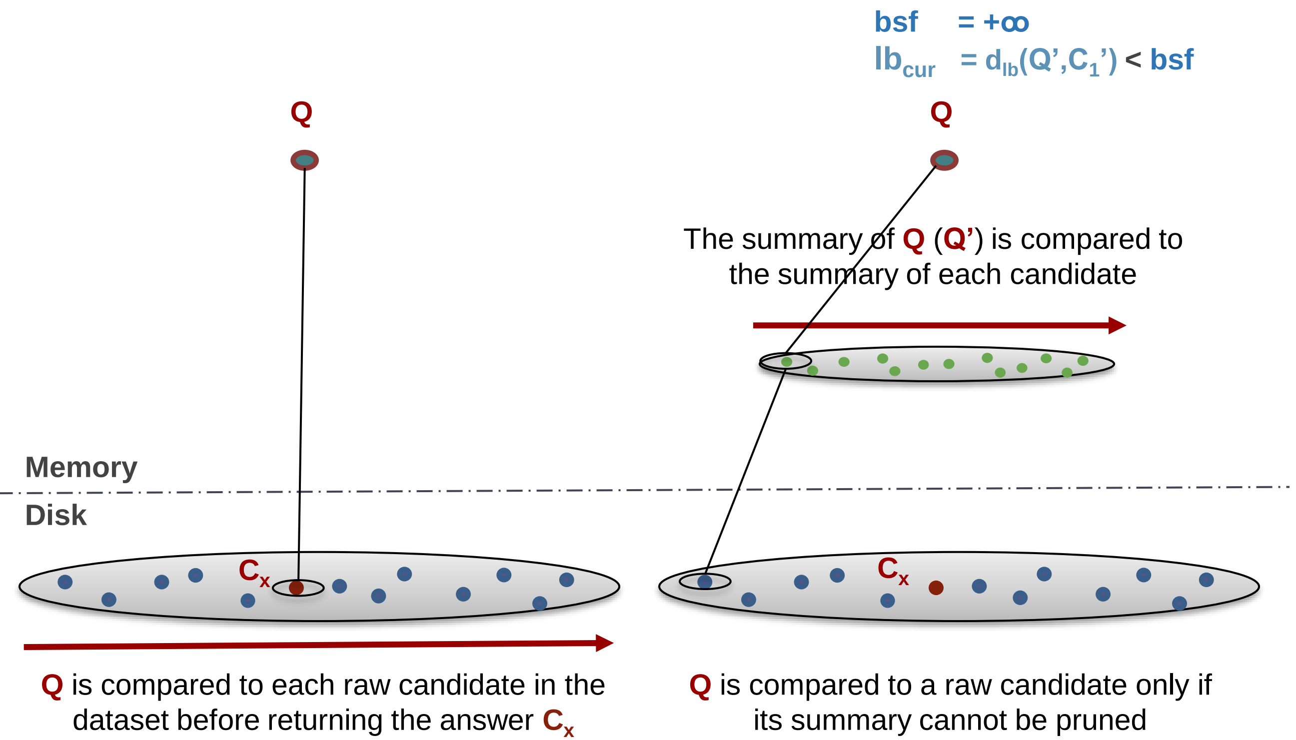
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

Indexes vs. Scans

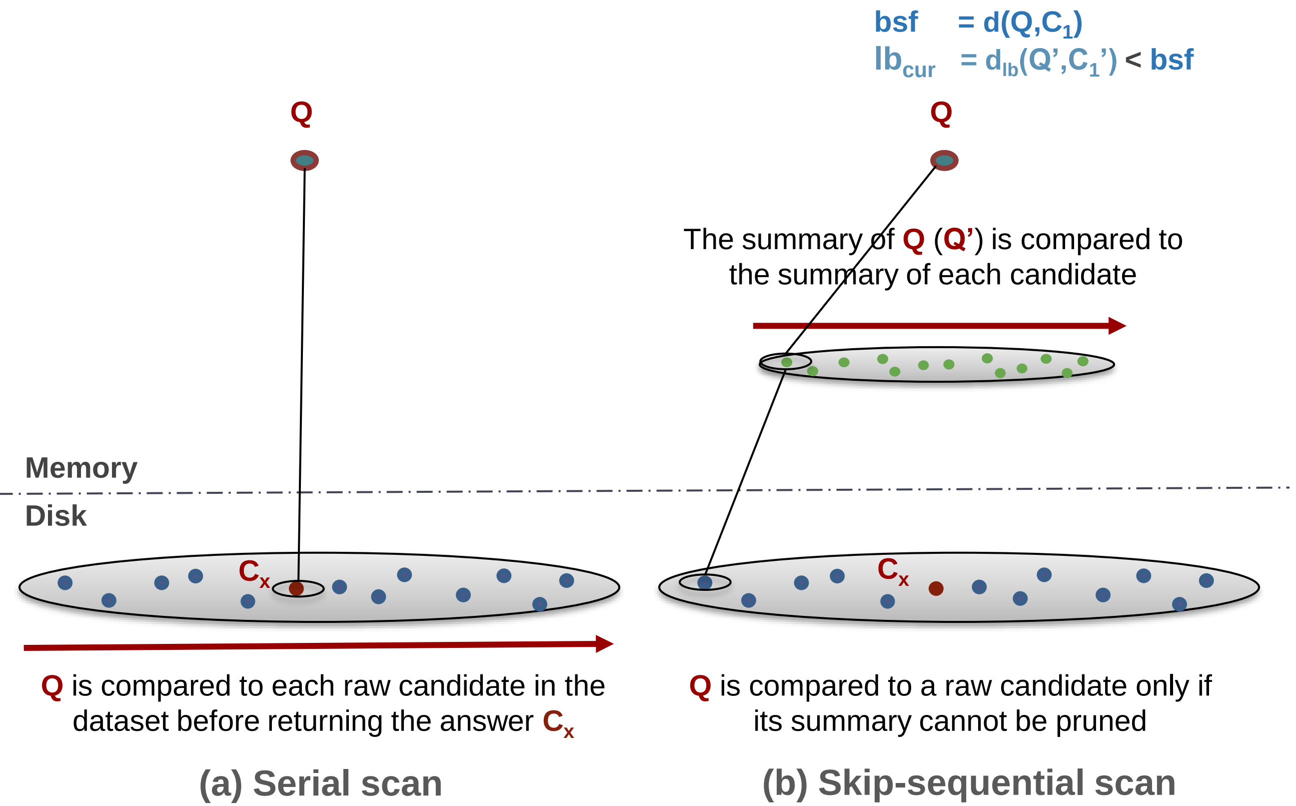


(a) Serial scan

(b) Skip-sequential scan

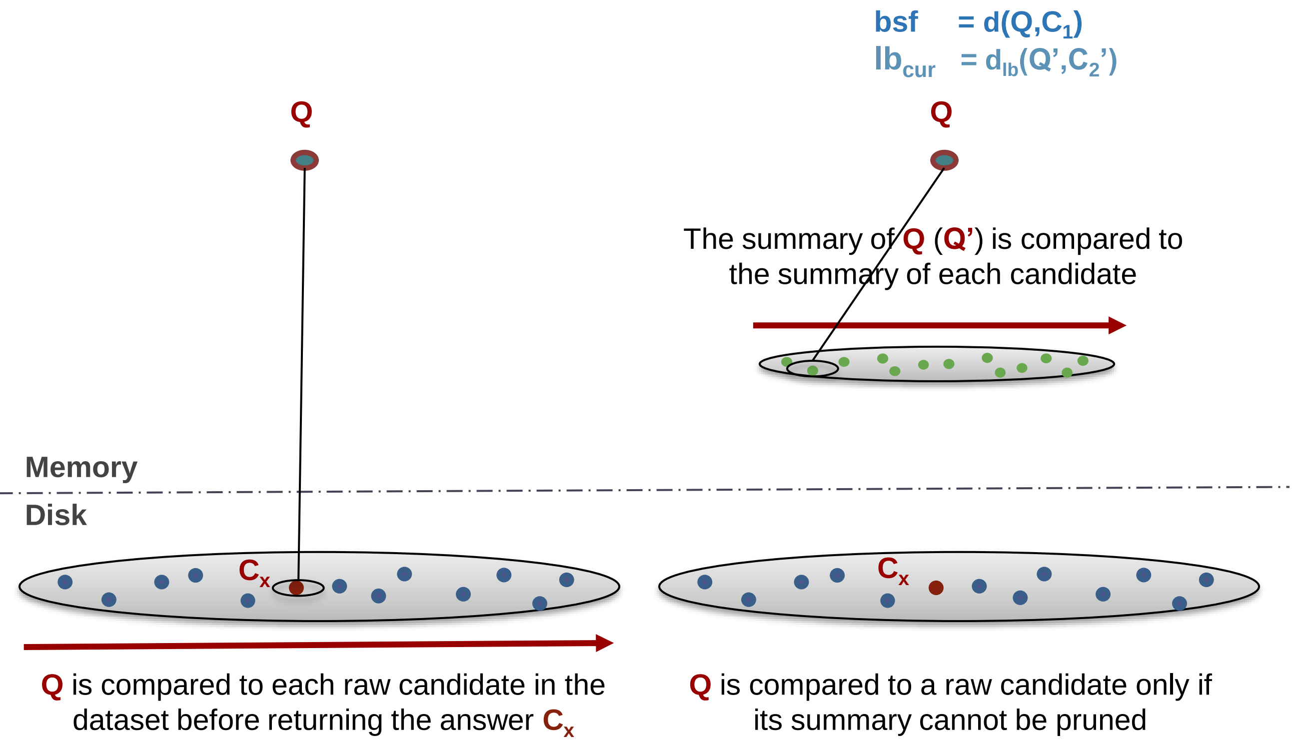
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

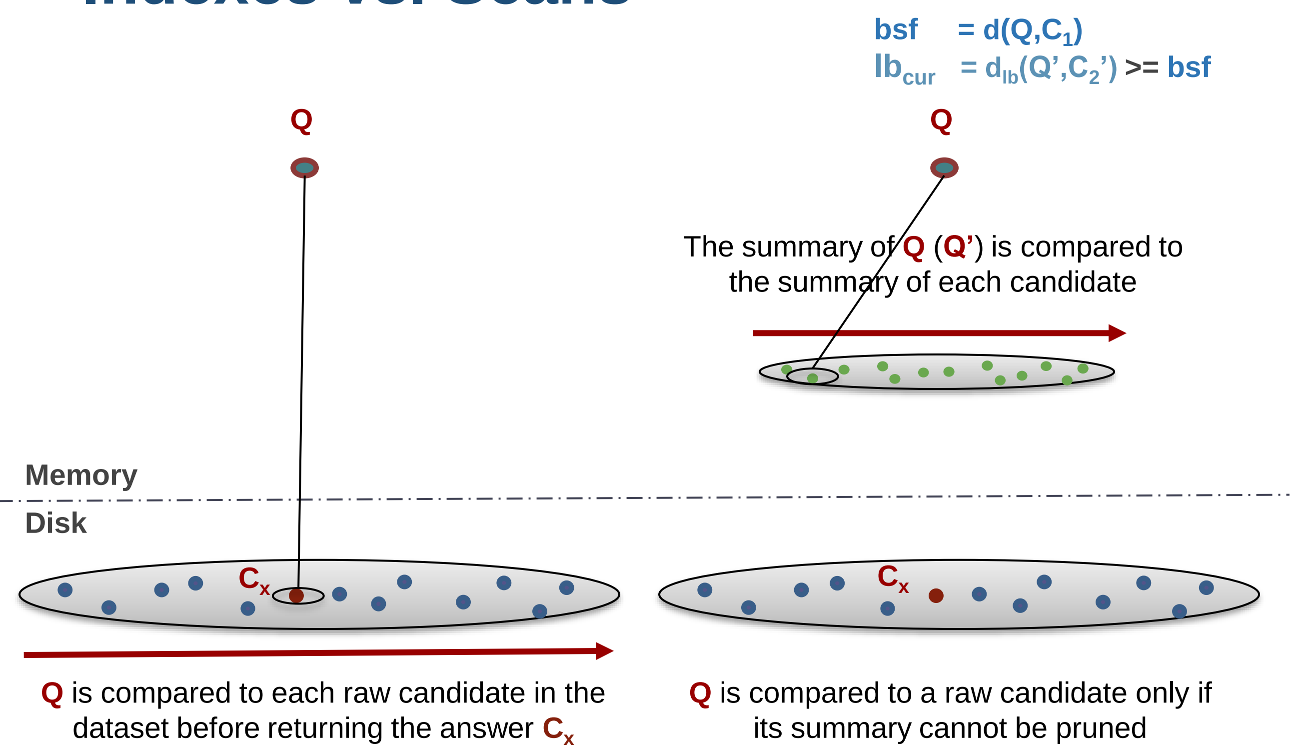
Indexes vs. Scans



(a) Serial scan **(b) Skip-sequential scan**

Answering a similarity search query using different access paths

Indexes vs. Scans



(a) Serial scan (b) Skip-sequential scan

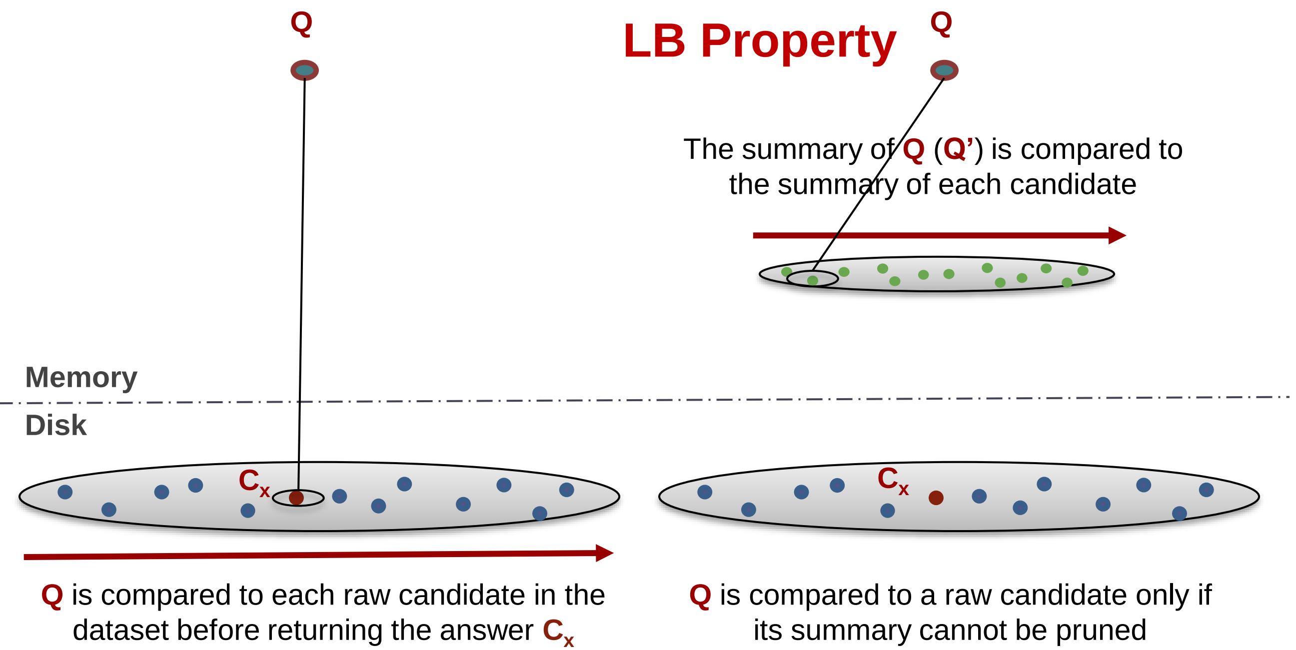
<u>Answering a similarity search query using different access paths</u>

Indexes vs. Scans

bsf = $d(Q,C_1)$

$d(Q,C_2)$ >= lb_{cur} = $d_{lb}(Q',C_2')$ >= bsf

LB Property

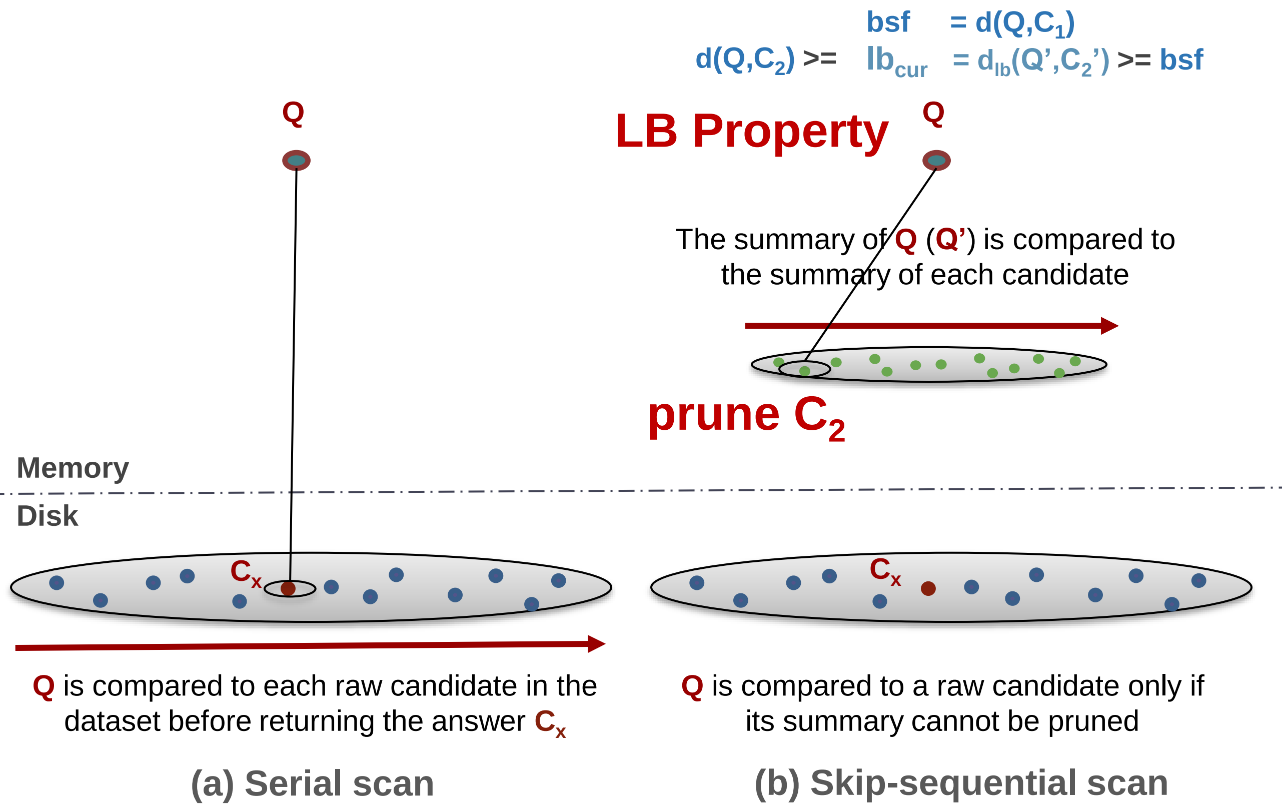


(a) Serial scan

(b) Skip-sequential scan

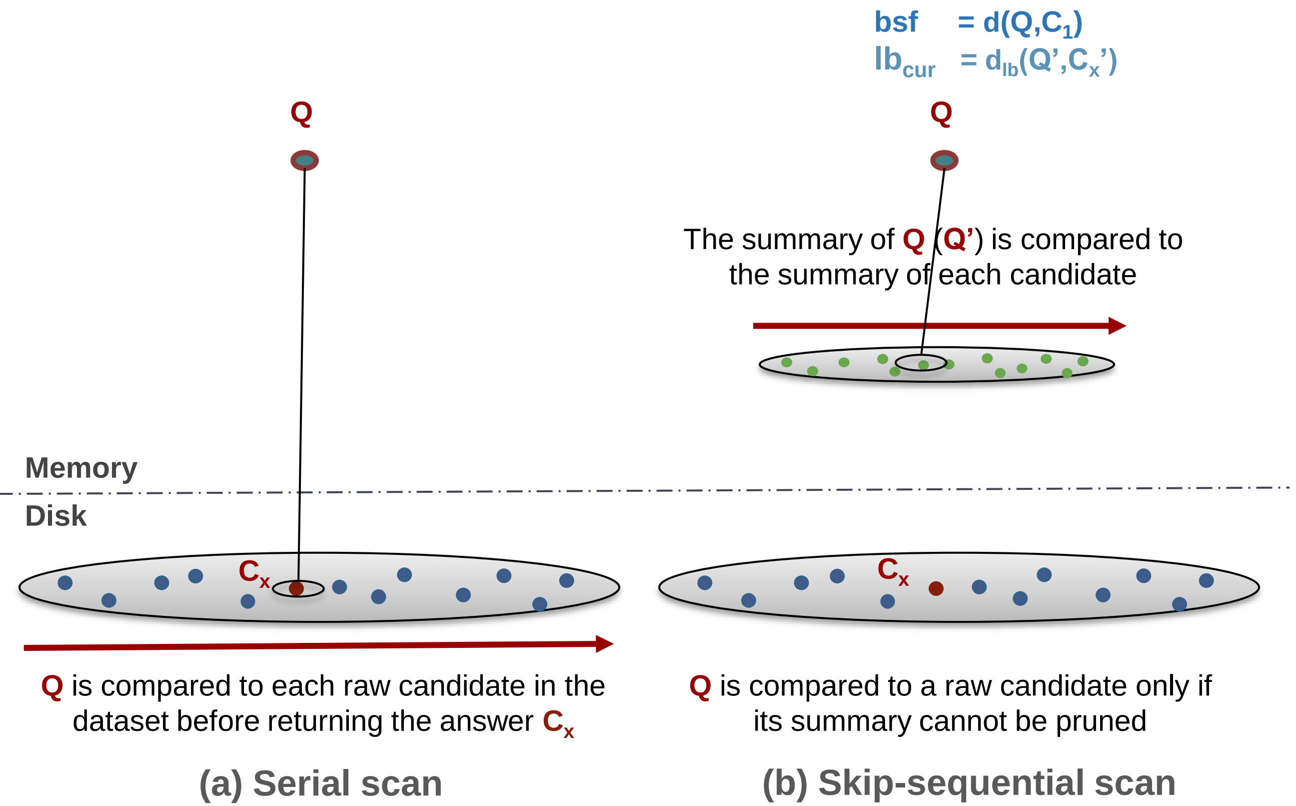
Answering a similarity search query using different access paths

Indexes vs. Scans



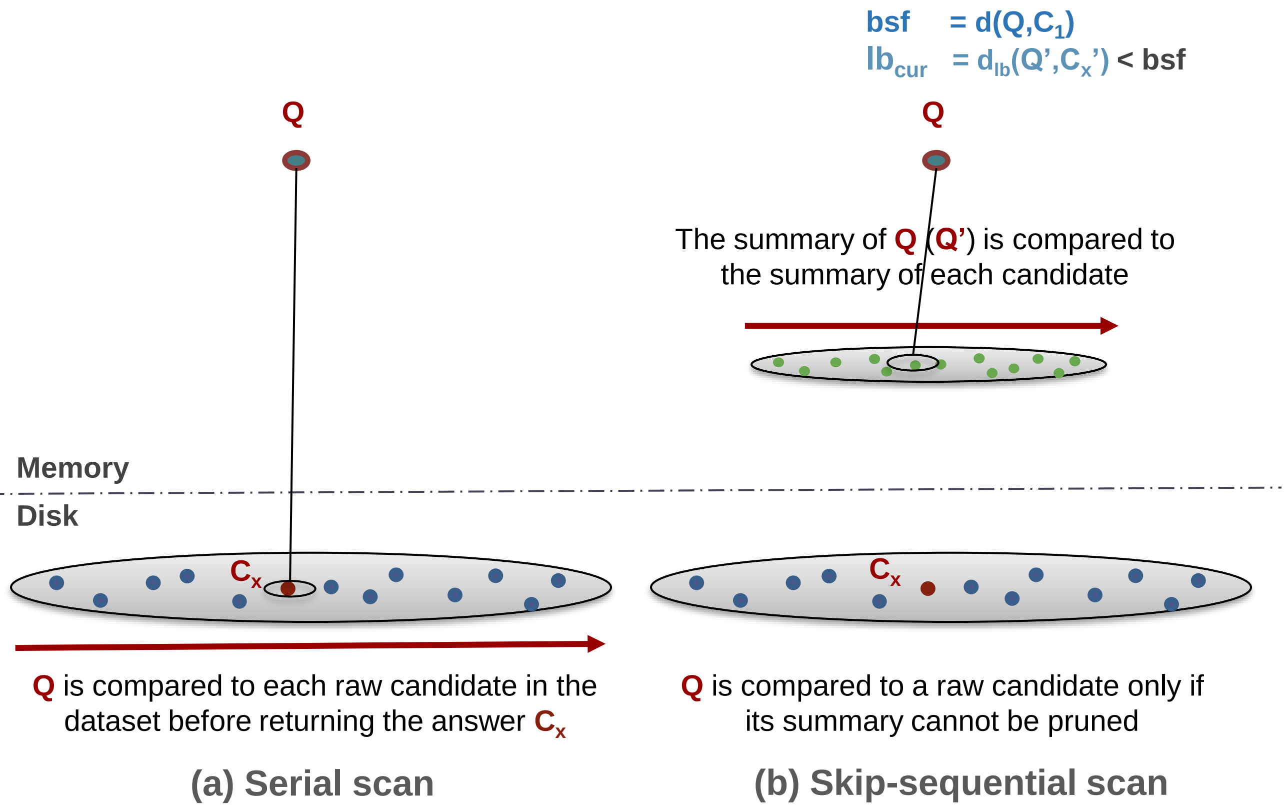
Answering a similarity search query using different access paths

Indexes vs. Scans



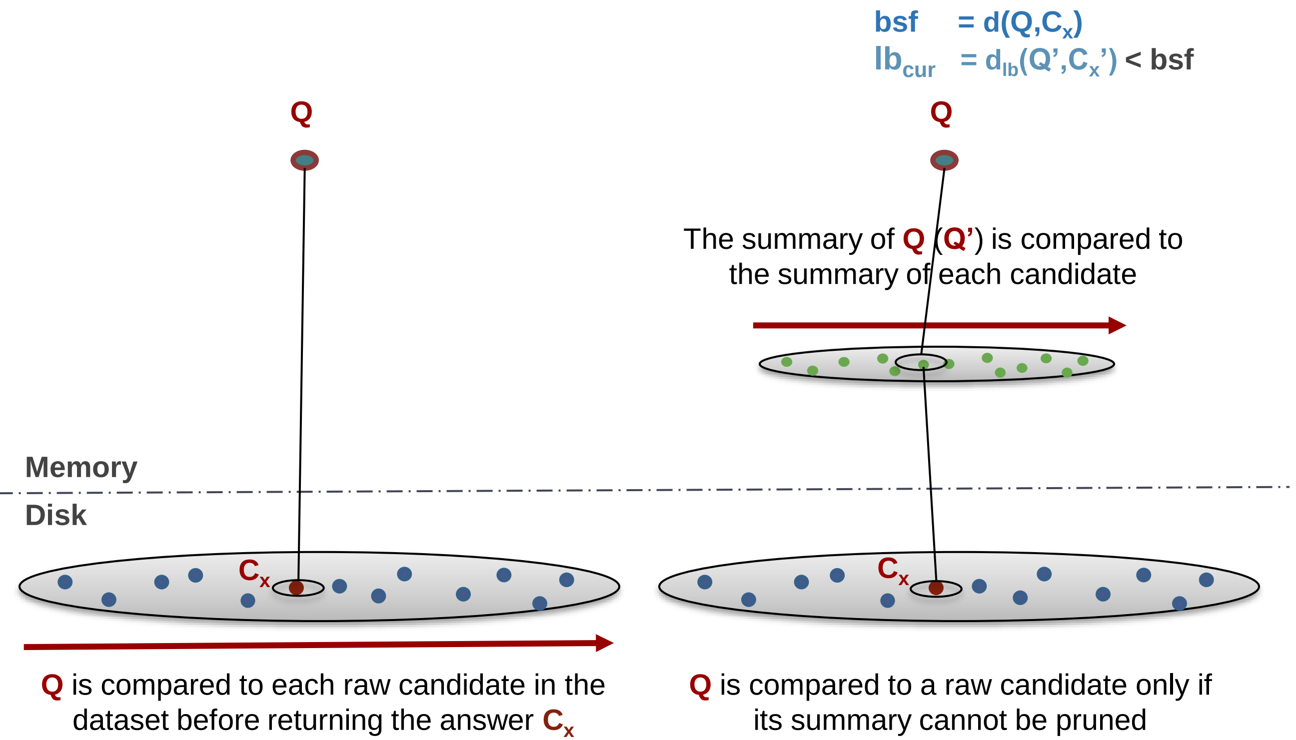
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

Indexes vs. Scans

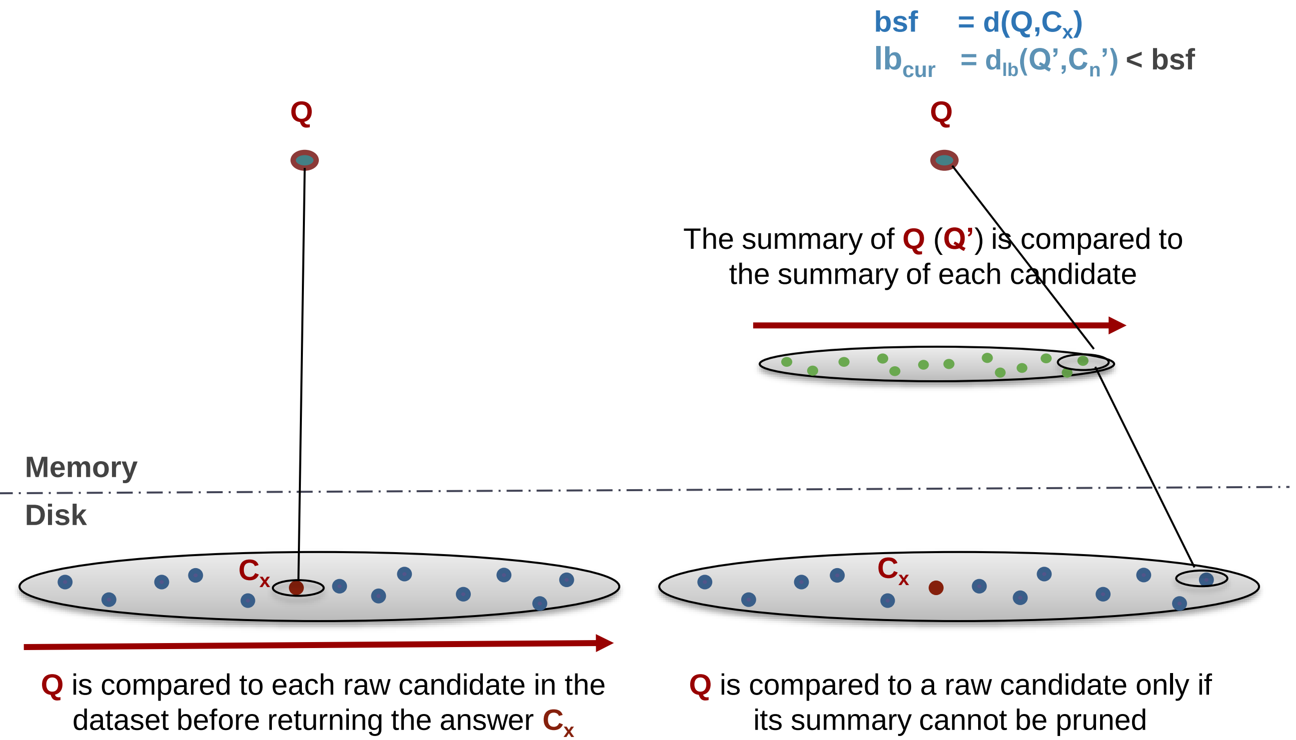


(a) Serial scan

(b) Skip-sequential scan

Answering a similarity search query using different access paths

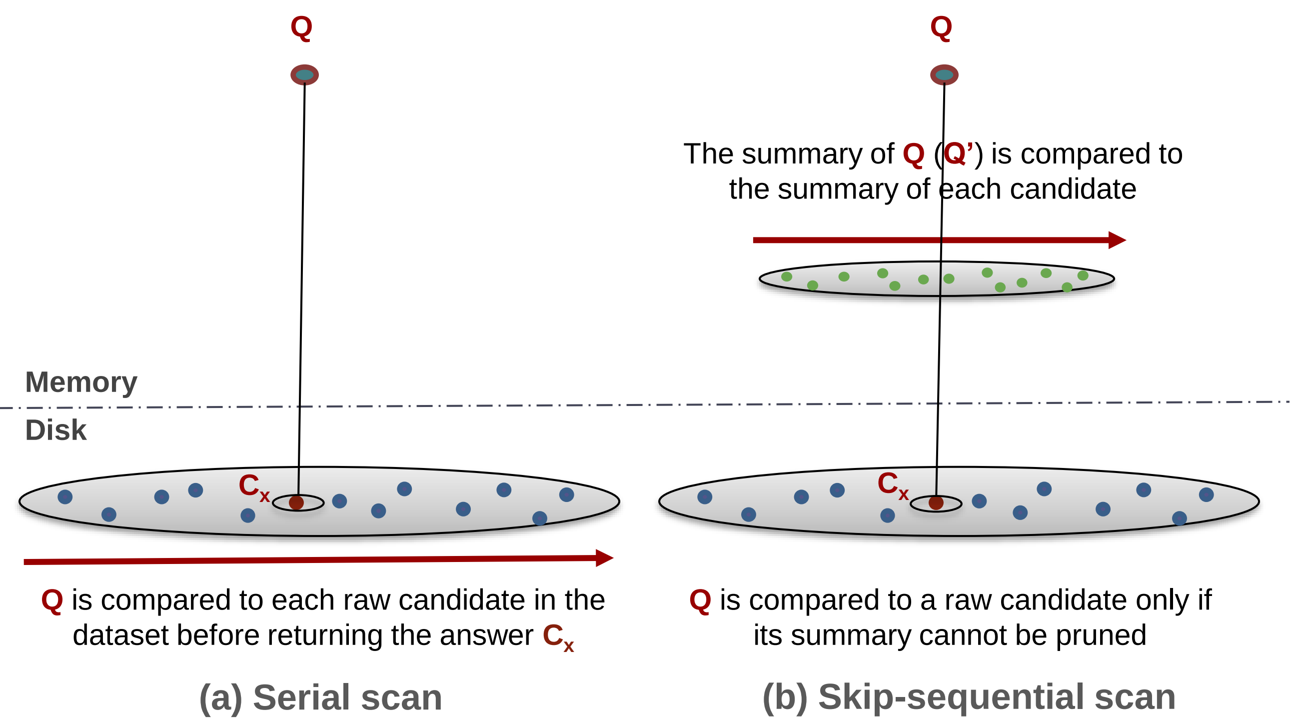
Indexes vs. Scans



(a) Serial scan (b) Skip-sequential scan

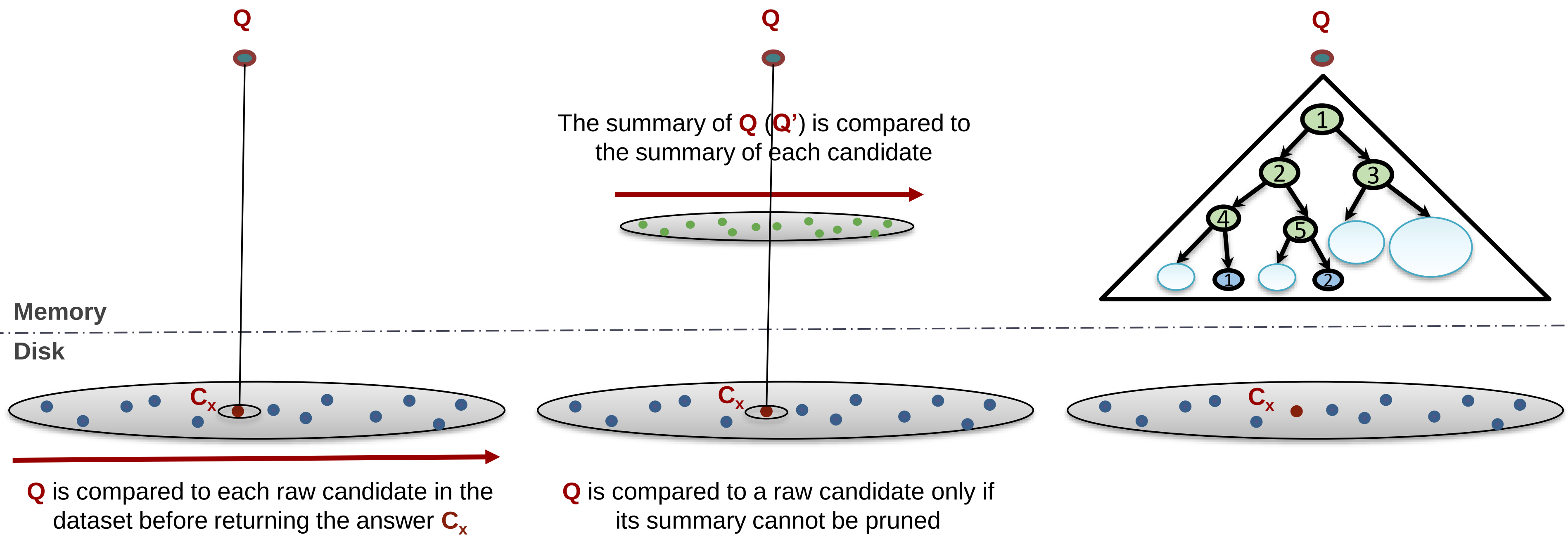
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

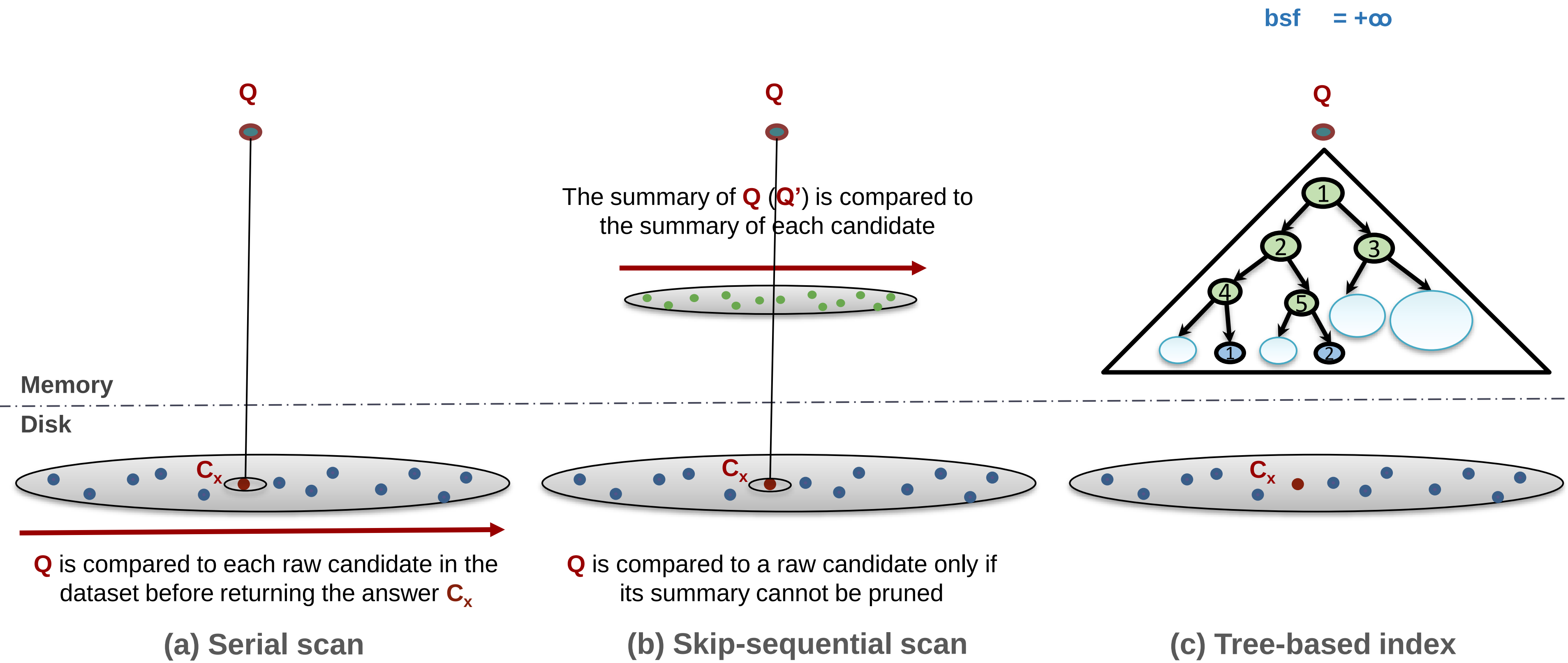
Indexes vs. Scans



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

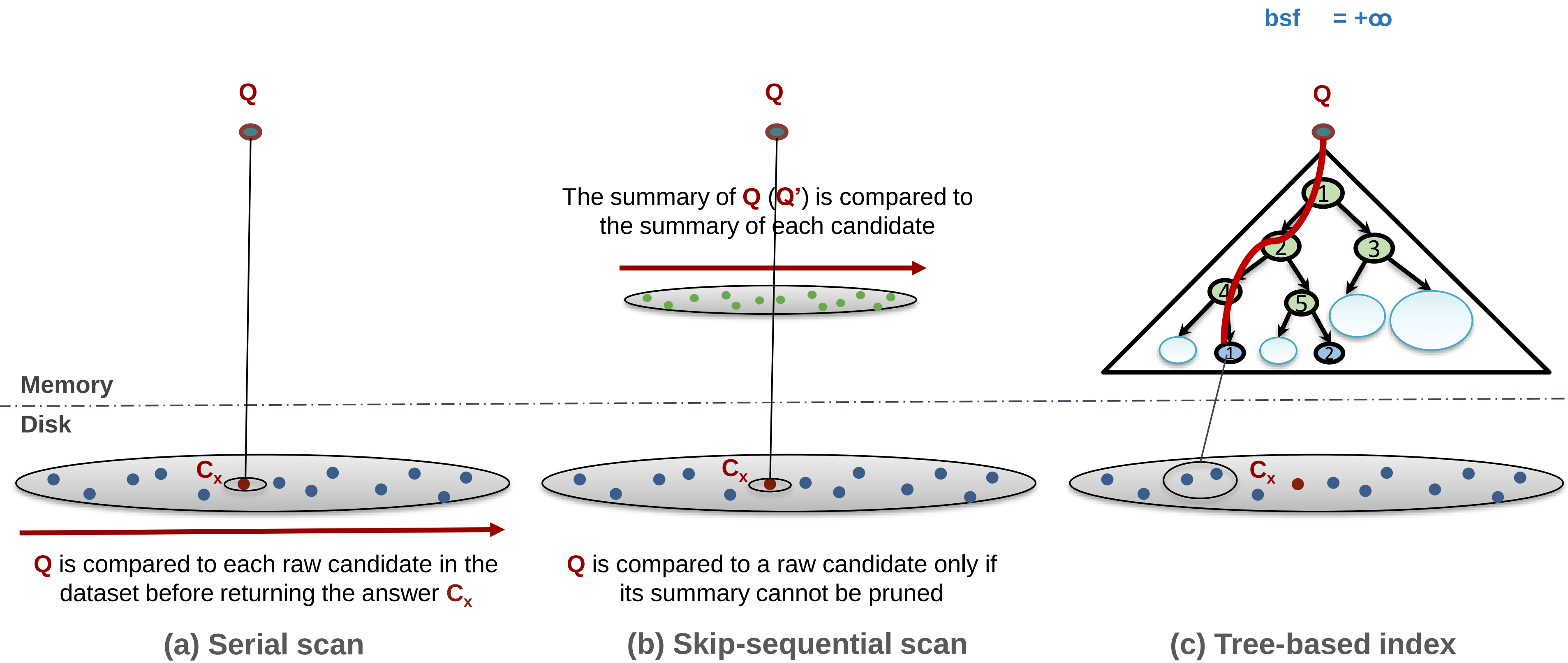
Answering a similarity search query using different access paths

Indexes vs. Scans



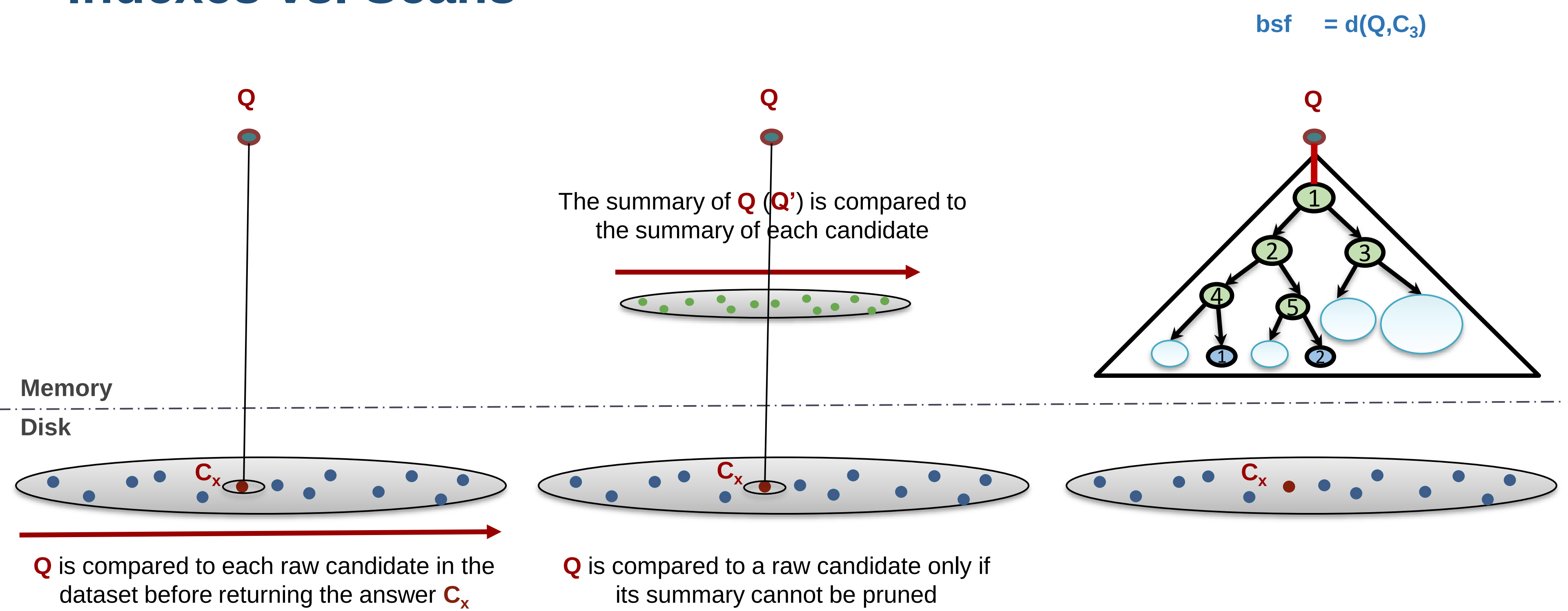
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

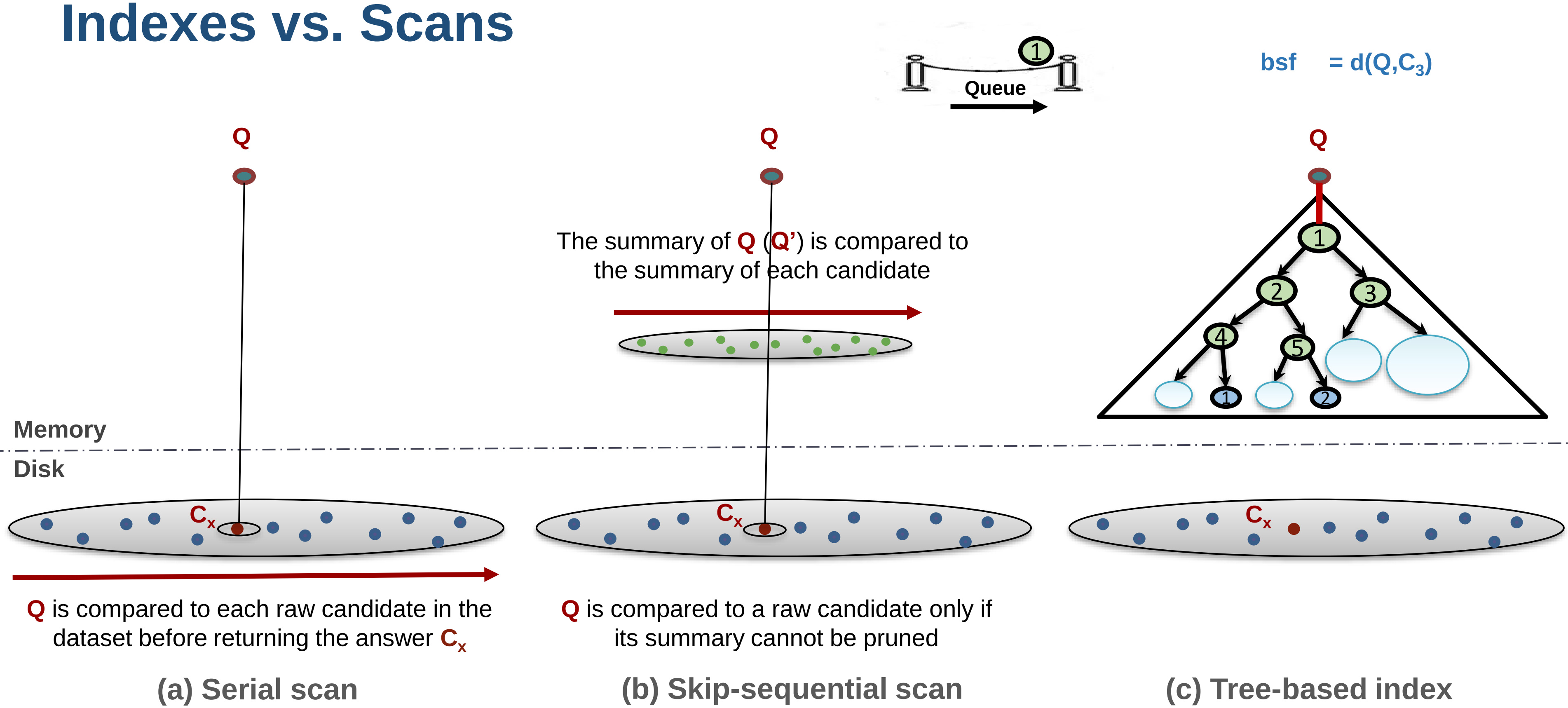
Indexes vs. Scans



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

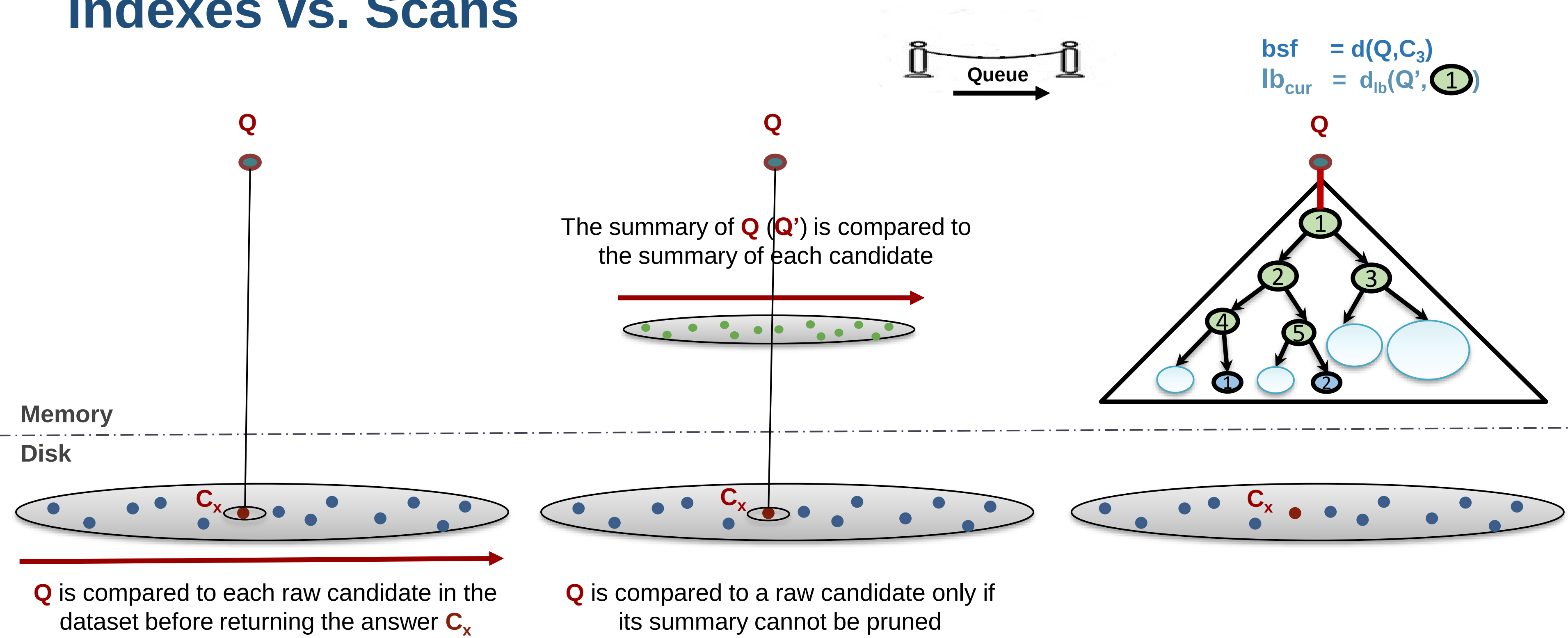
<u>Answering a similarity search query using different access paths</u>

Indexes vs. Scans



Answering a similarity search query using different access paths

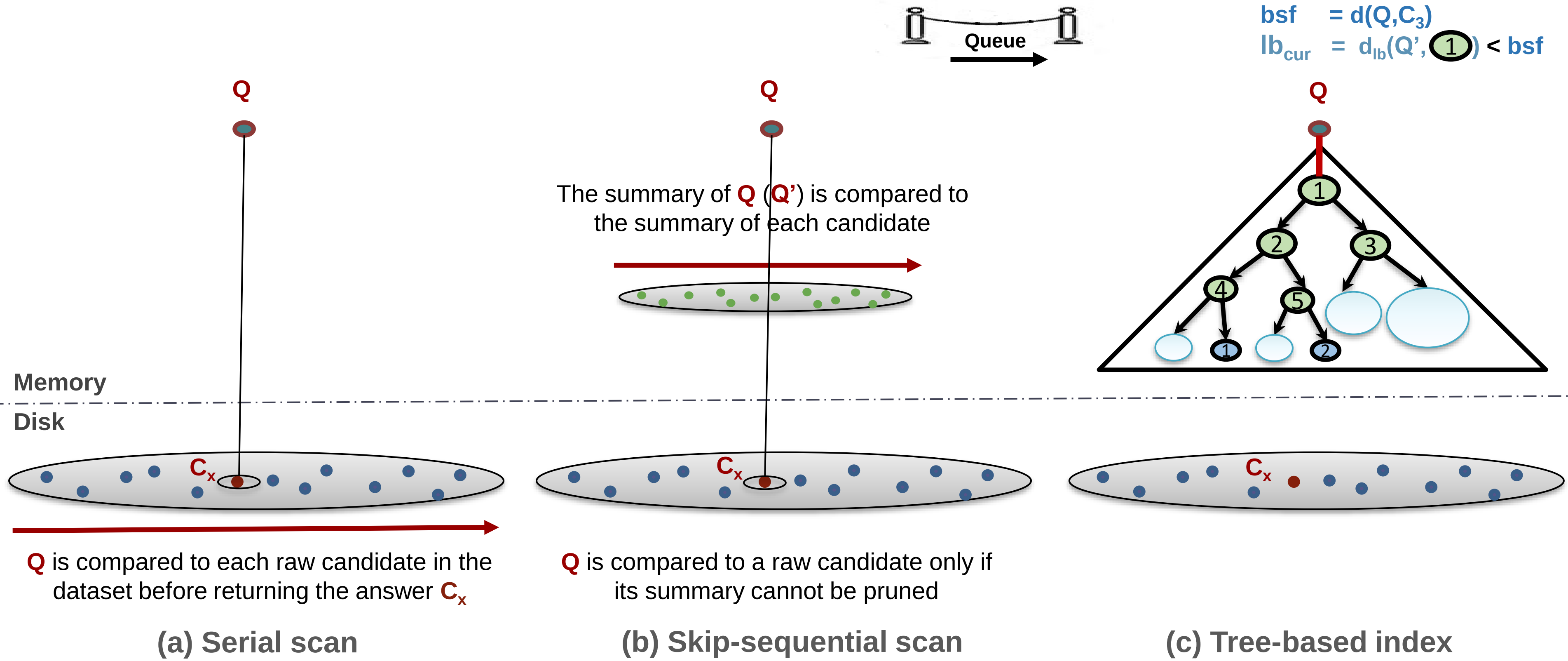
Indexes vs. Scans



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

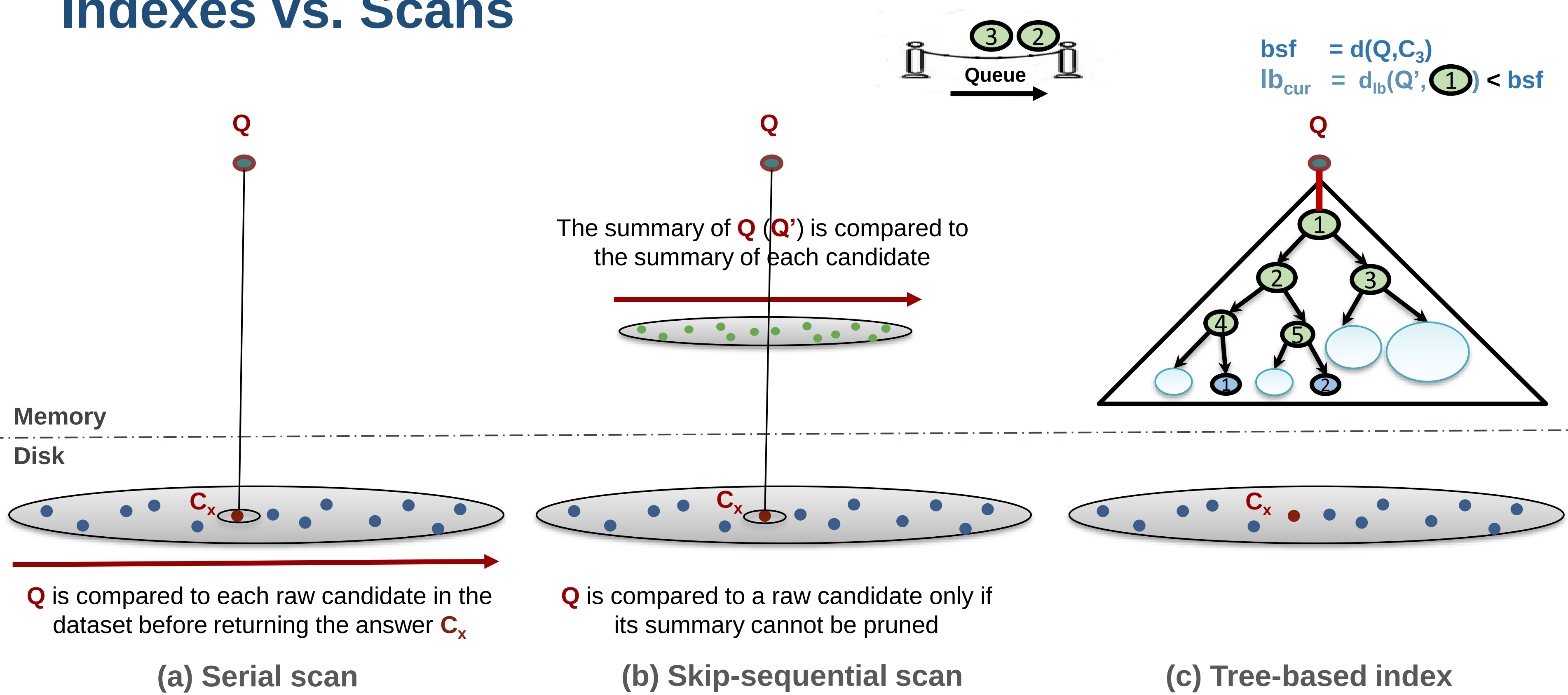
Answering a similarity search query using different access paths

Indexes vs. Scans



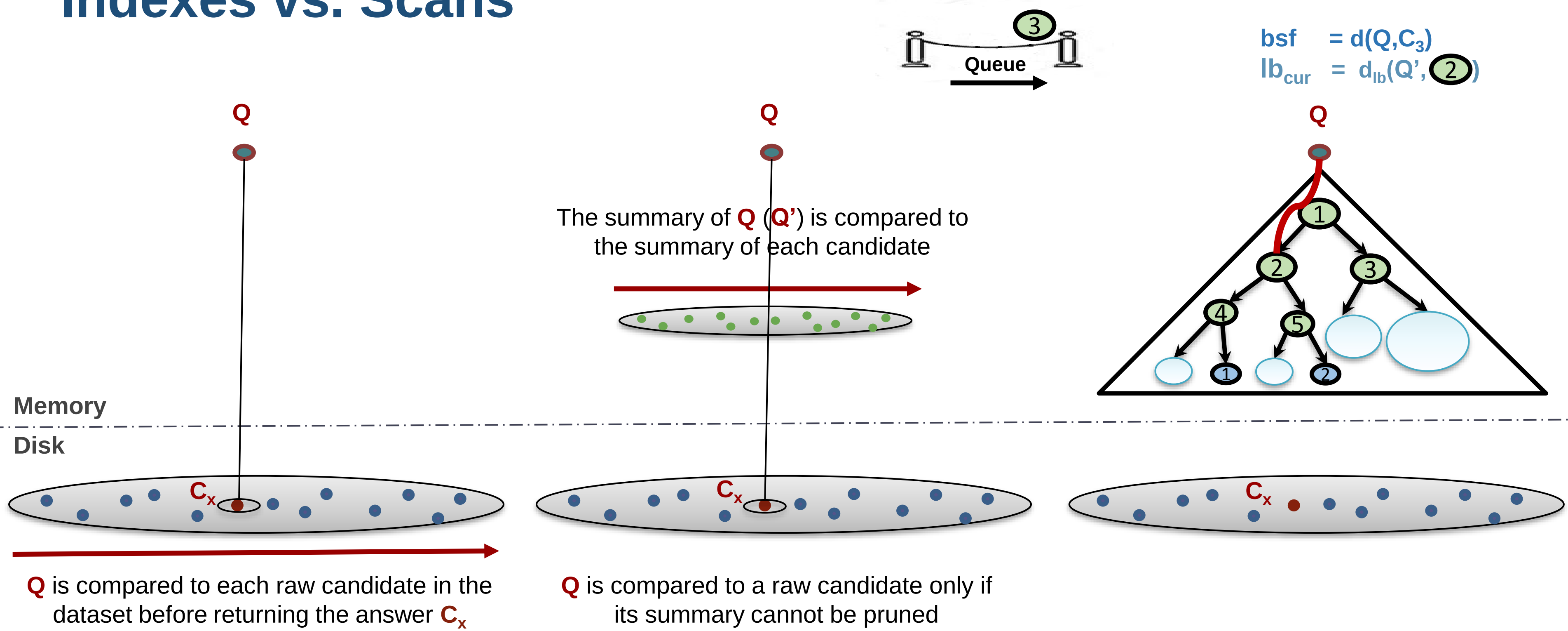
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

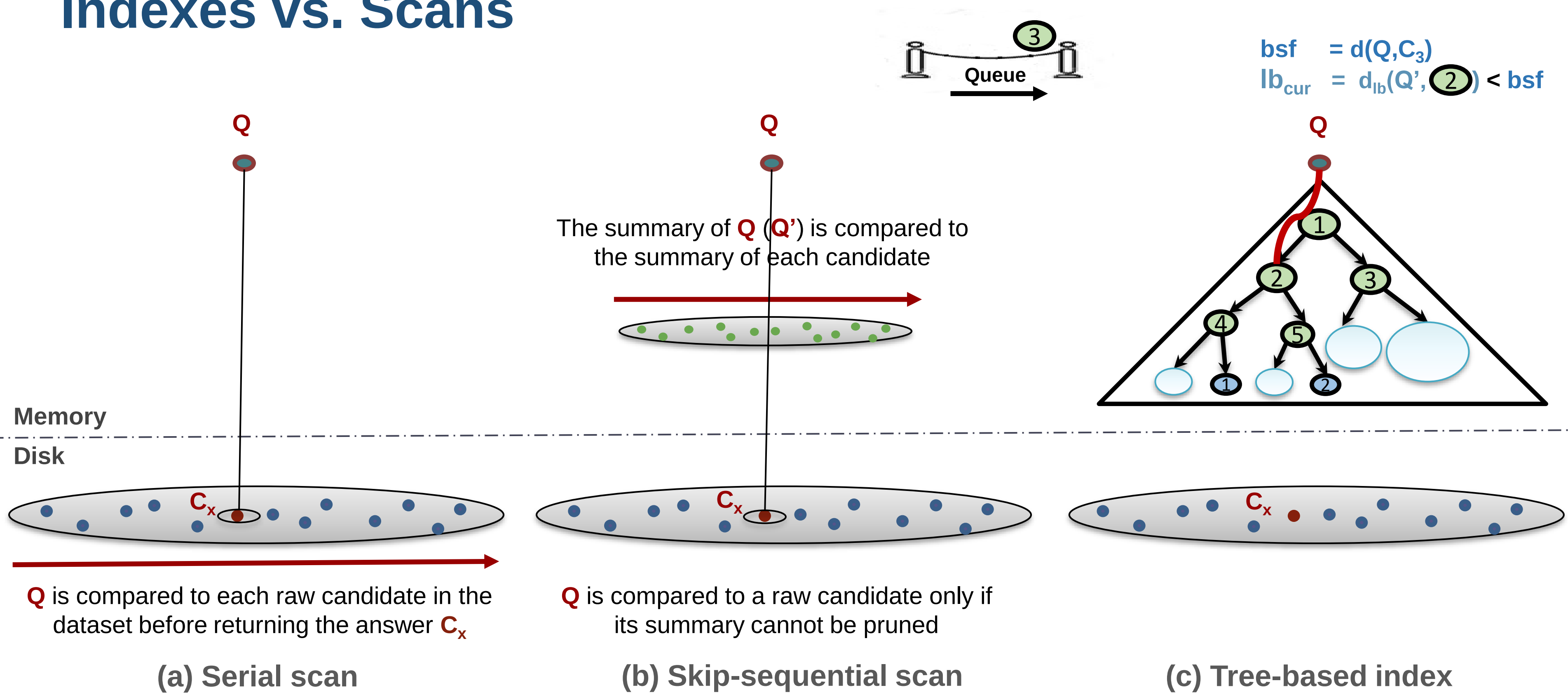
Indexes vs. Scans



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

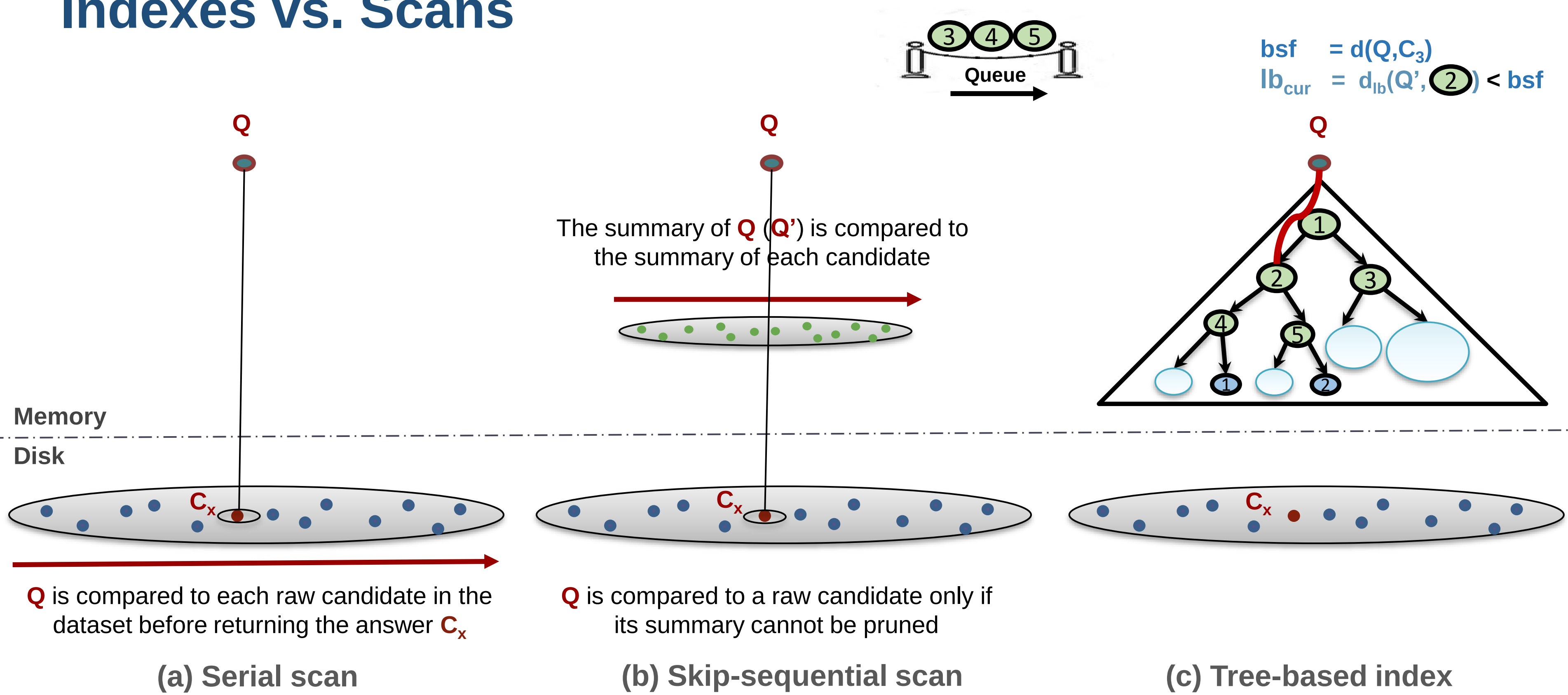
<u>Answering a similarity search query using different access paths</u>

Indexes vs. Scans



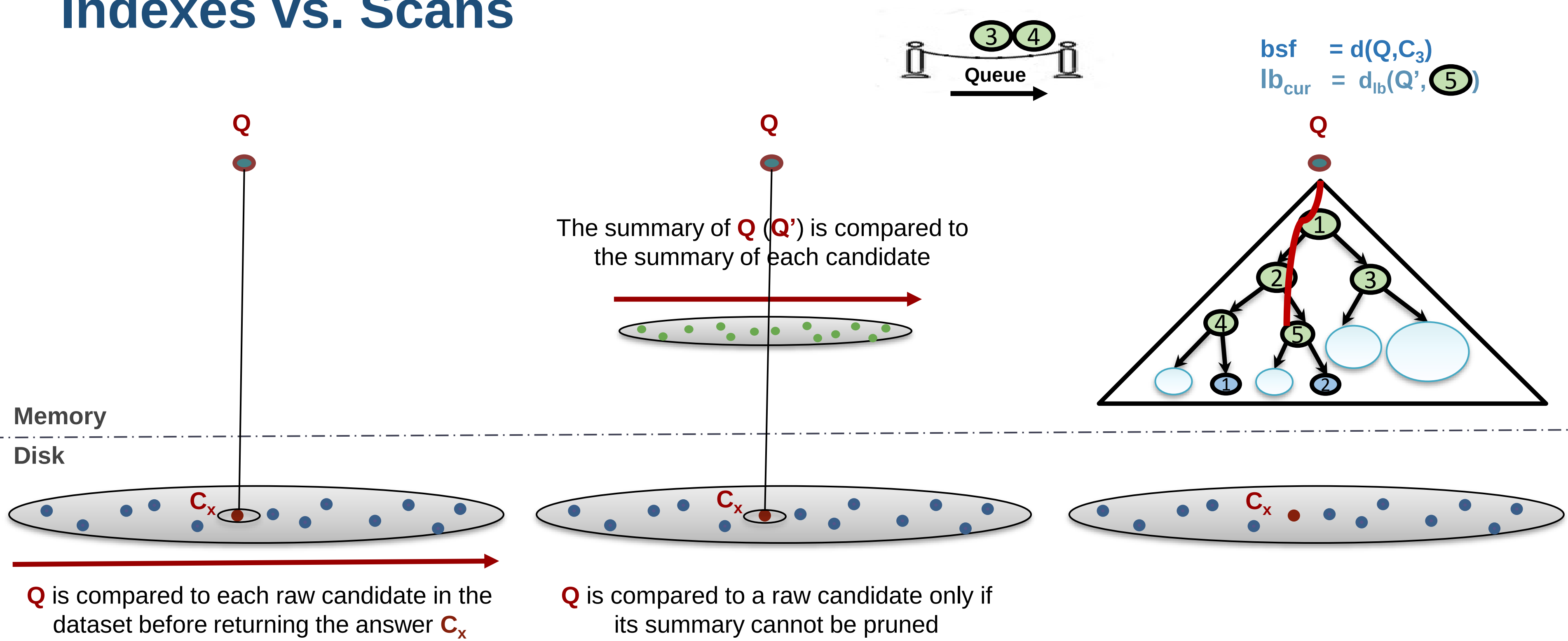
<u>Answering a similarity search query using different access paths</u>

Indexes vs. Scans



<u>Answering a similarity search query using different access paths</u>

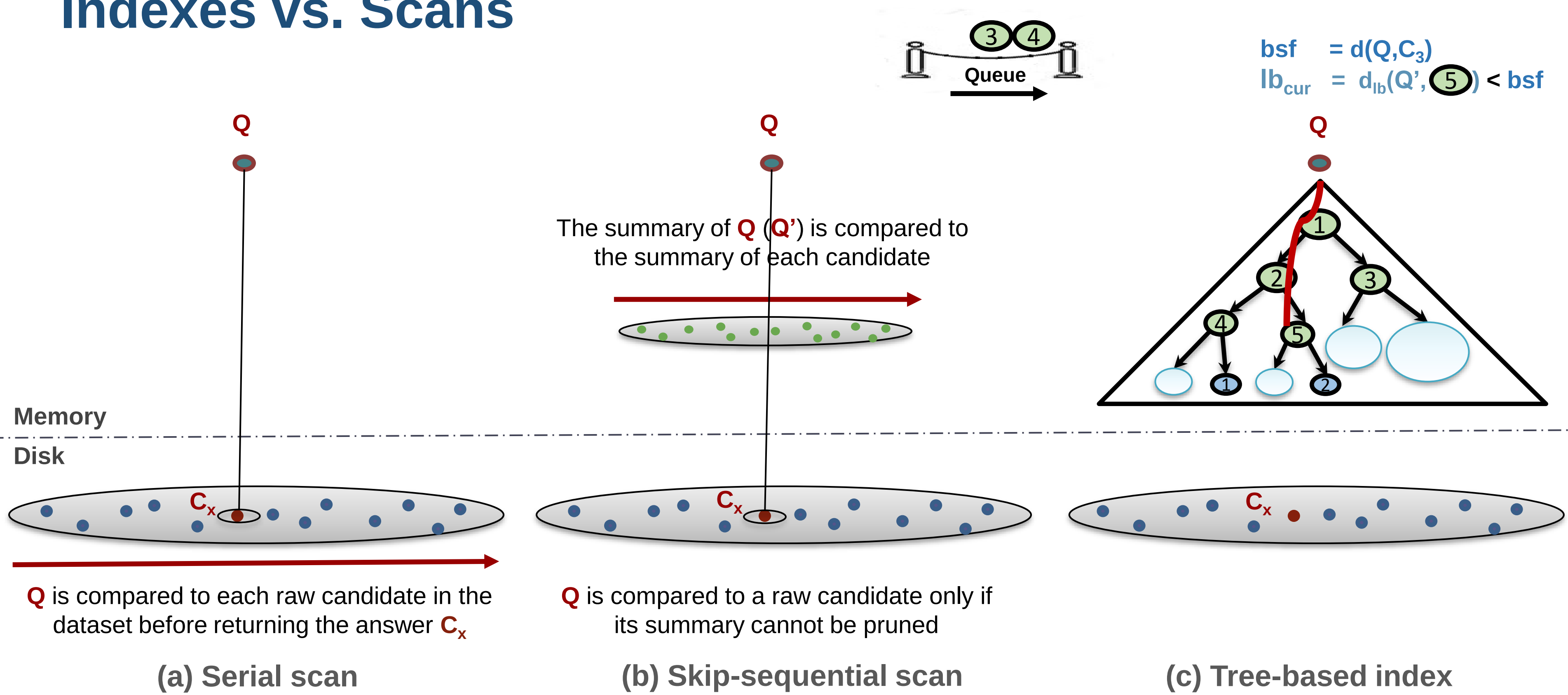
Indexes vs. Scans



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

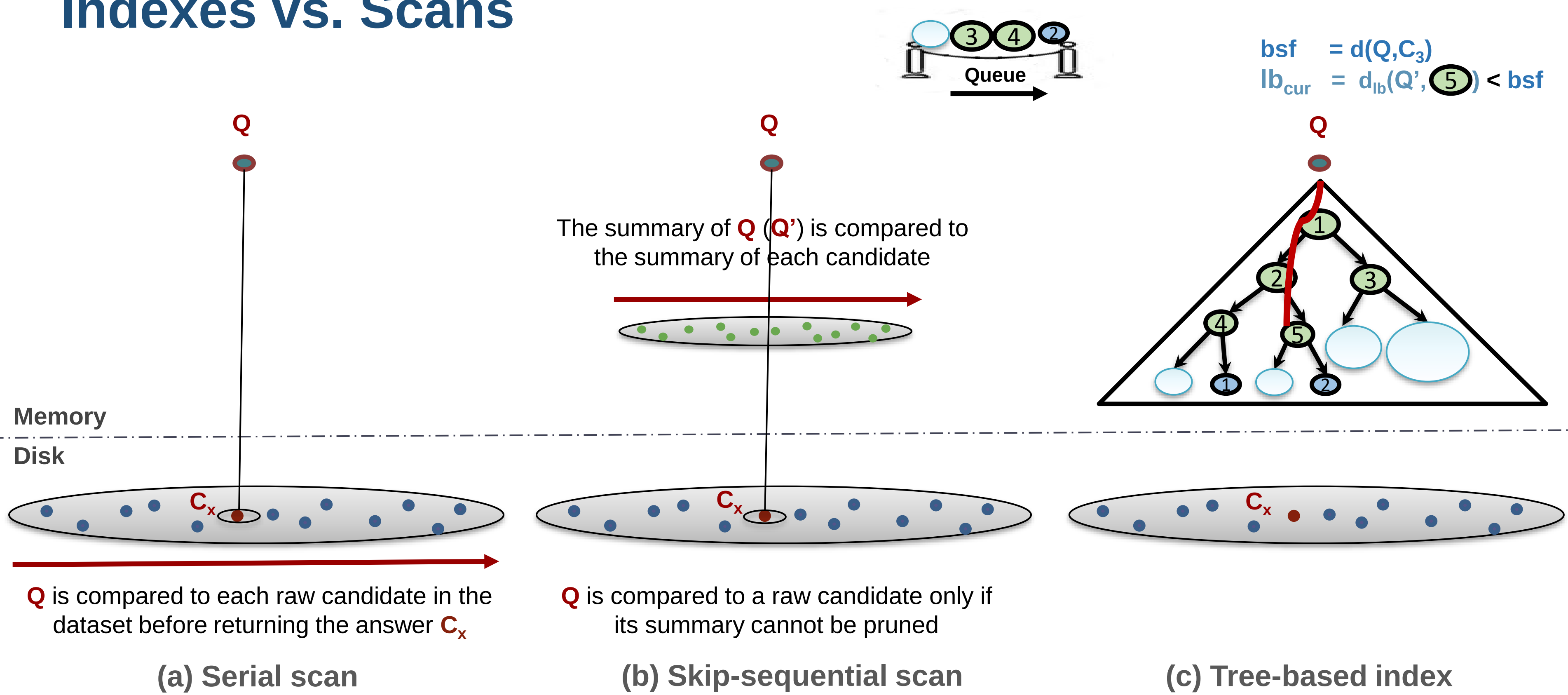
Answering a similarity search query using different access paths

Indexes vs. Scans



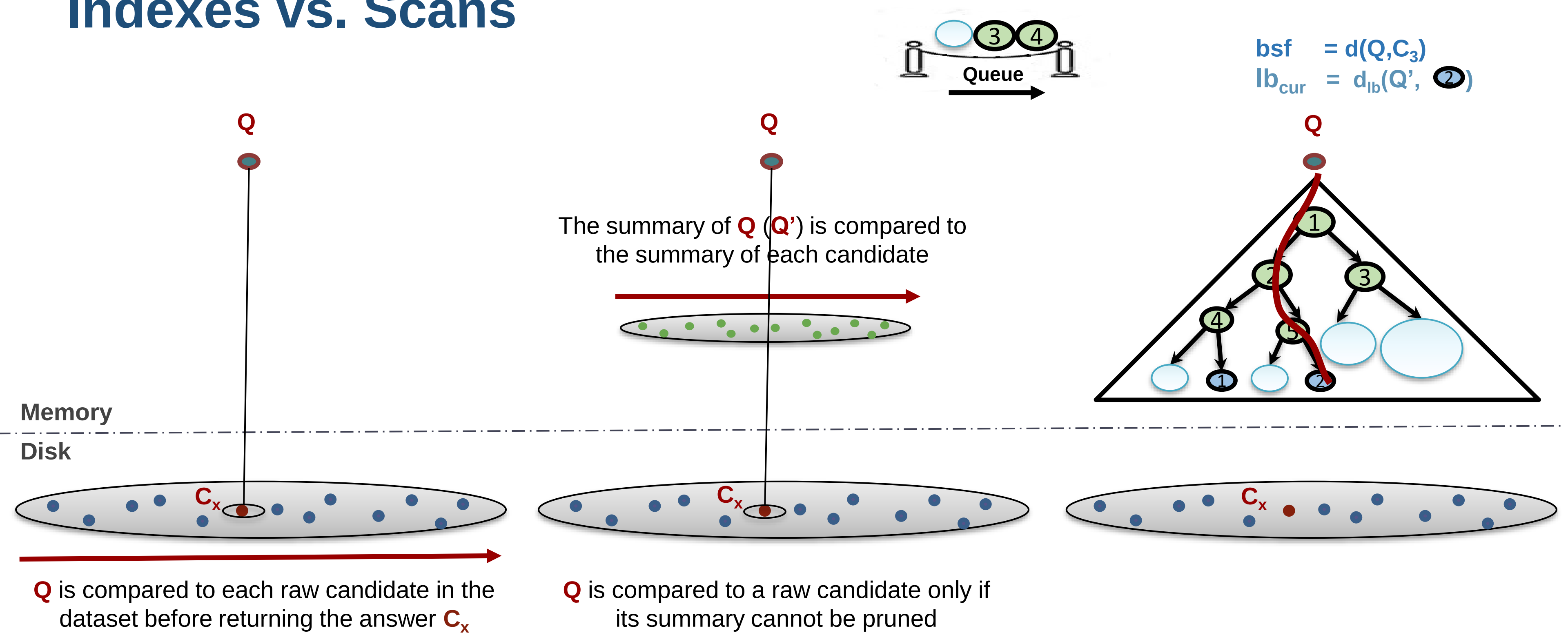
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

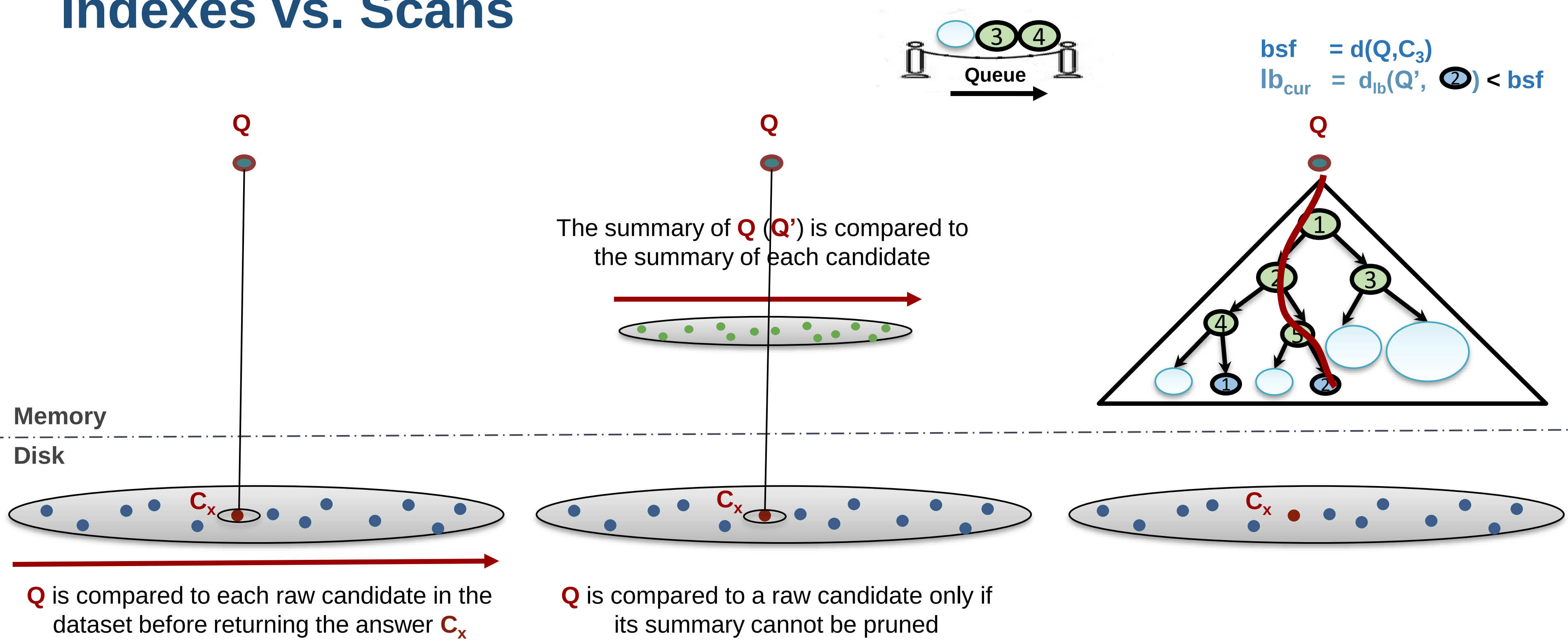
Indexes vs. Scans



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

Answering a similarity search query using different access paths

Indexes vs. Scans



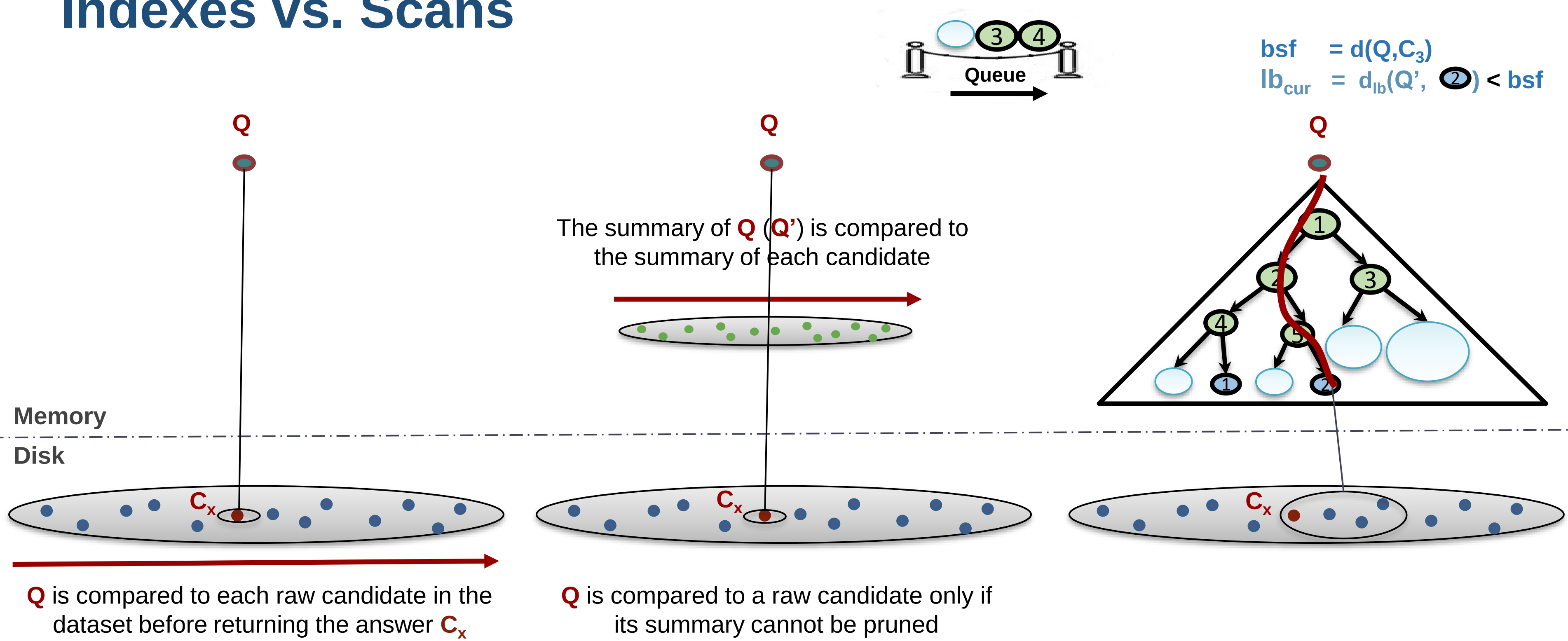
(a) Serial scan

(b) Skip-sequential scan

(c) Tree-based index

Answering a similarity search query using different access paths

Indexes vs. Scans



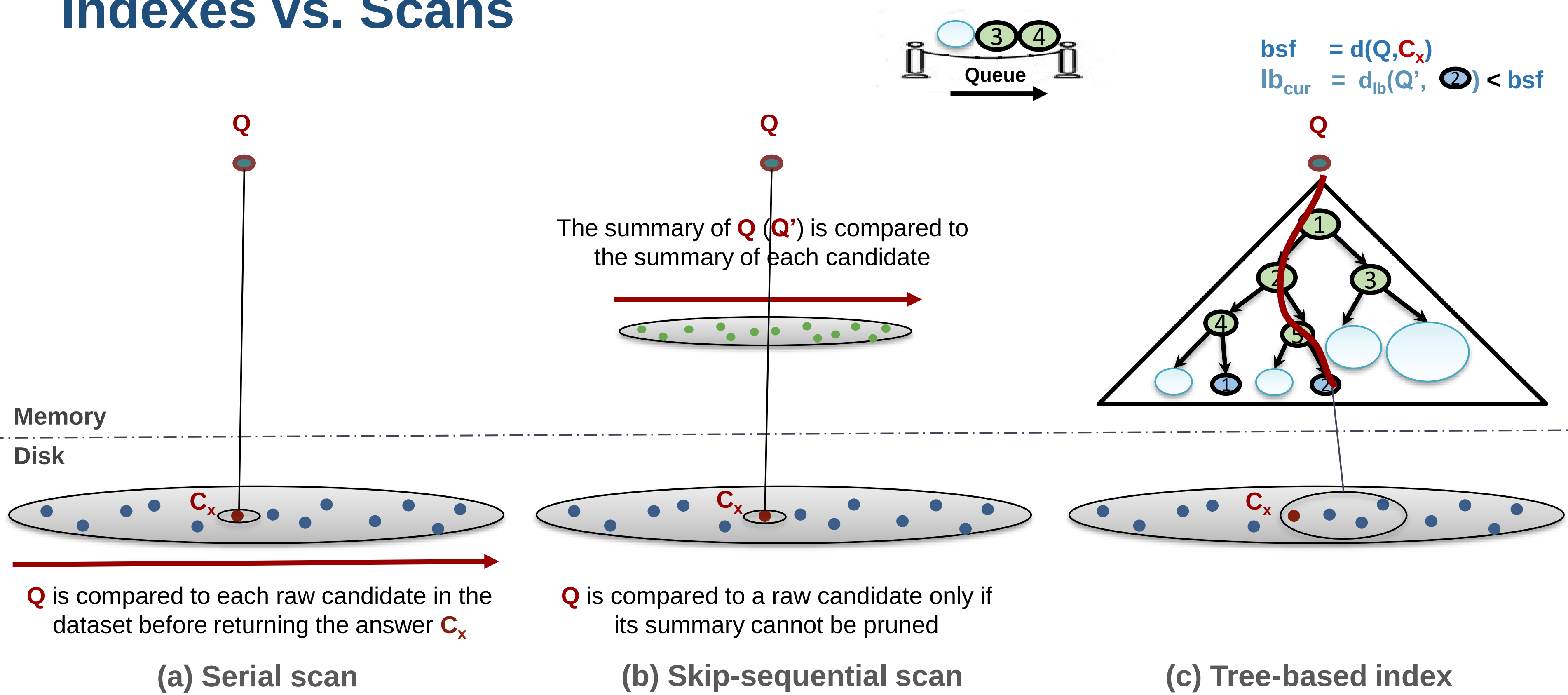
(a) Serial scan

(b) Skip-sequential scan

(c) Tree-based index

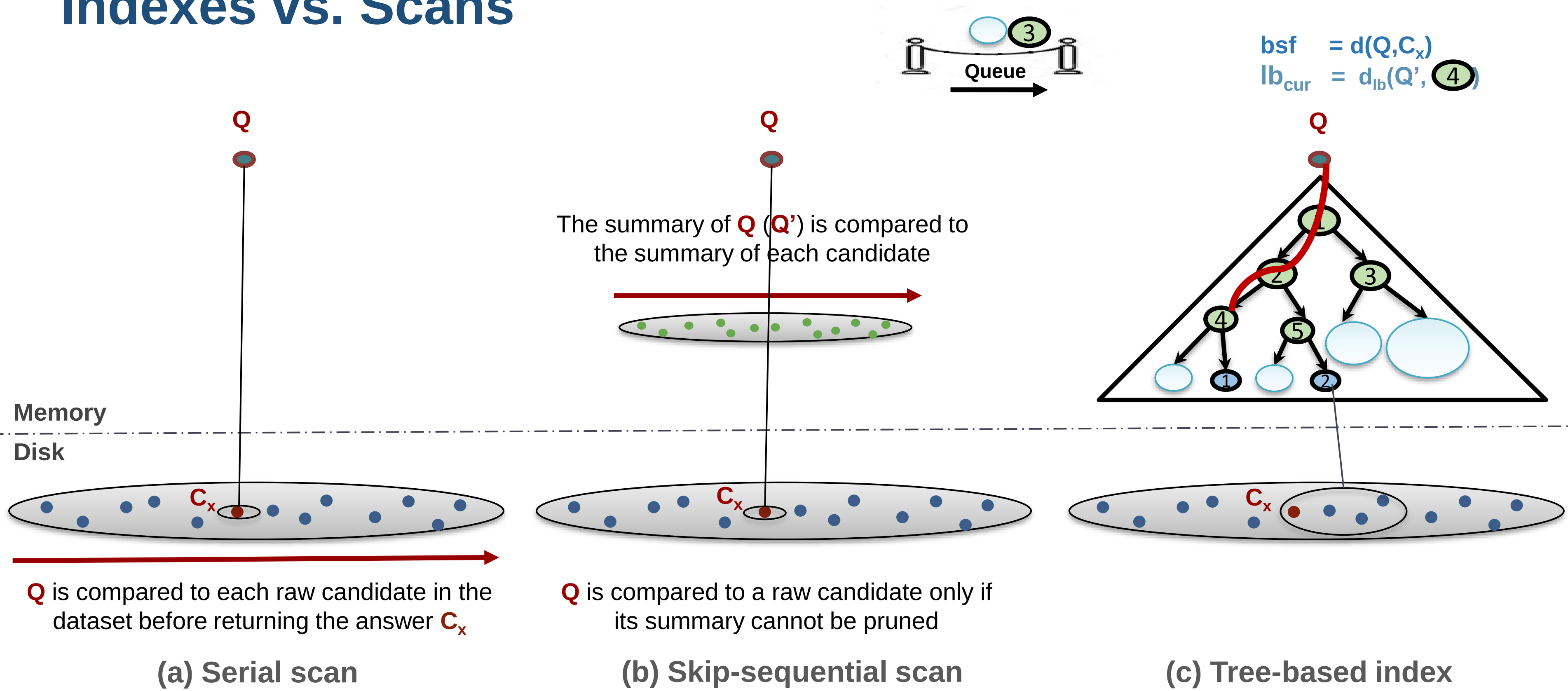
Answering a similarity search query using different access paths

Indexes vs. Scans



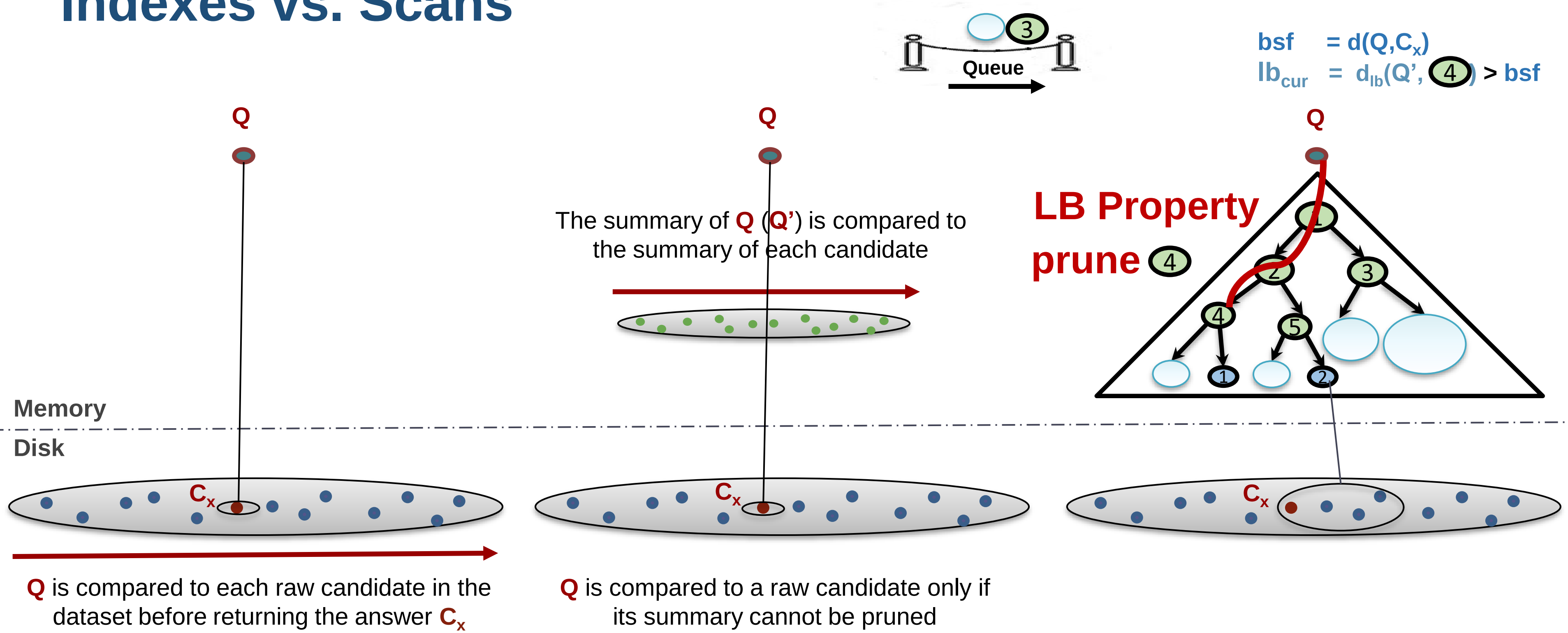
Answering a similarity search query using different access paths

Indexes vs. Scans



<u>Answering a similarity search query using different access paths</u>

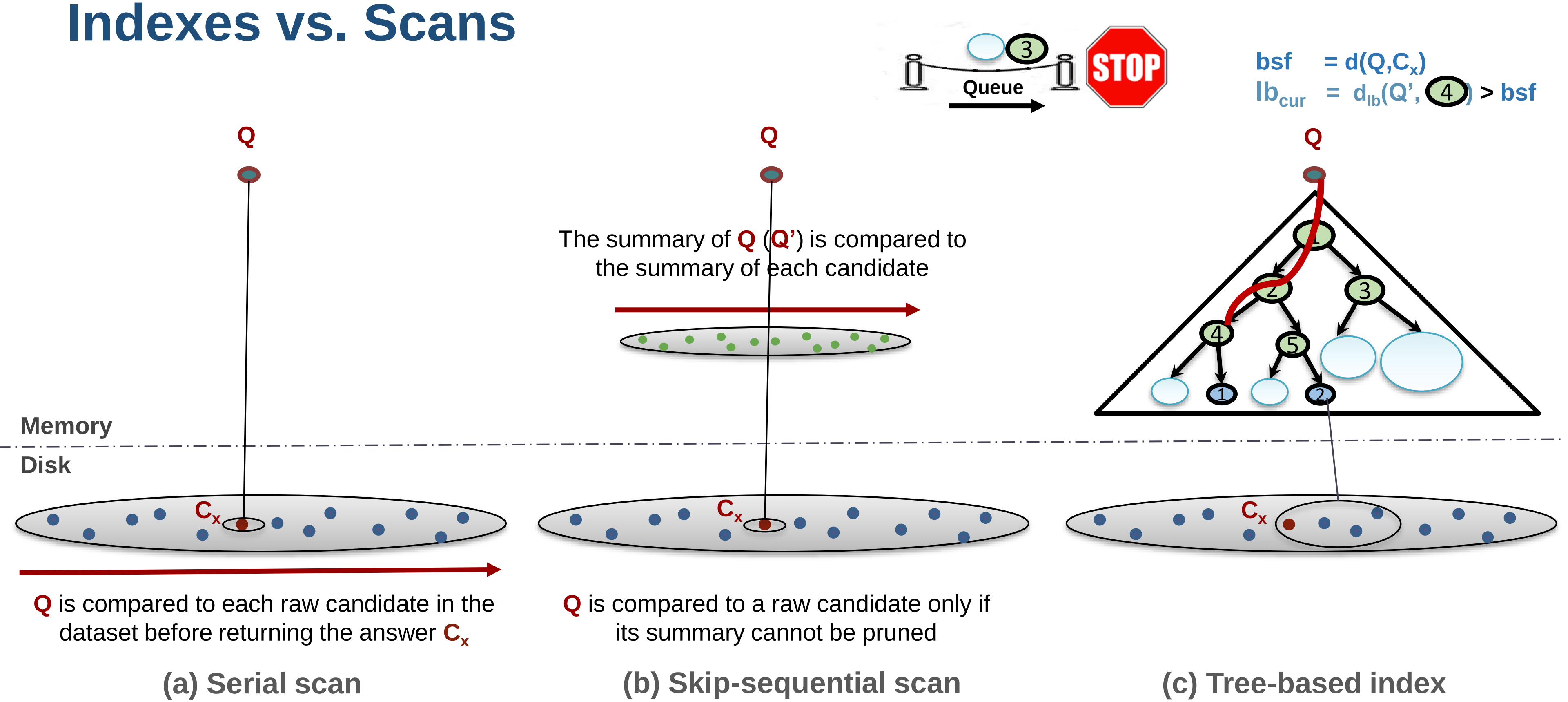
Indexes vs. Scans



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

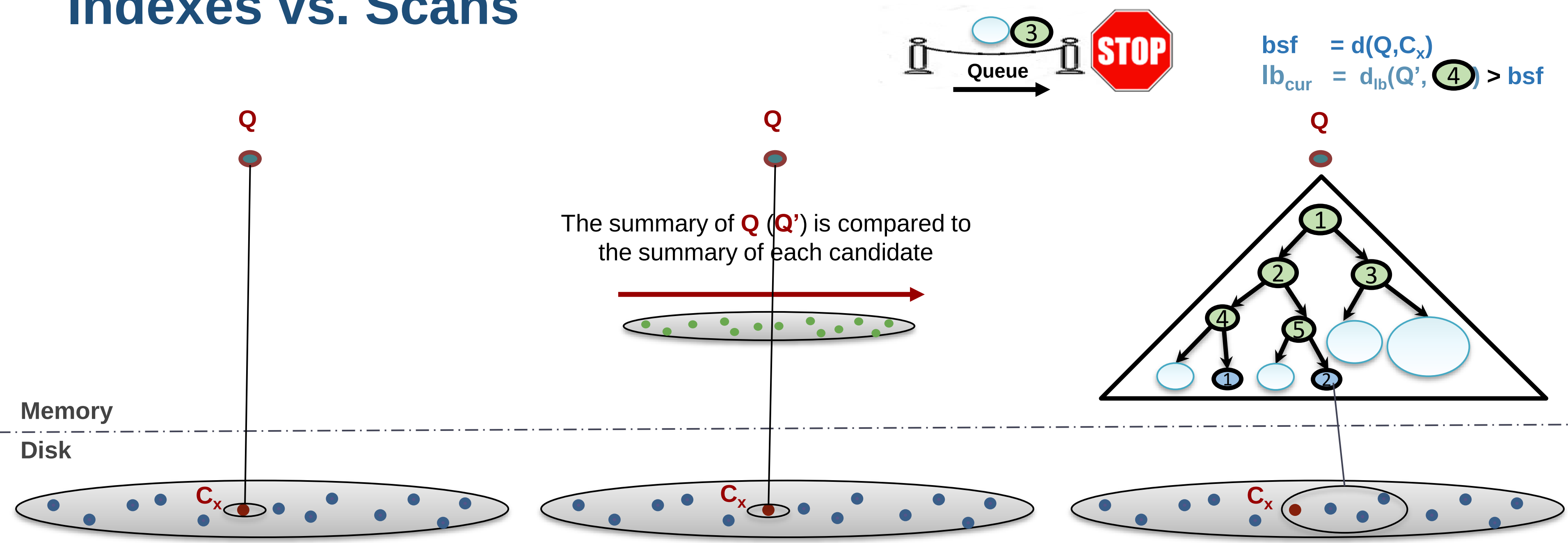
Answering a similarity search query using different access paths

Indexes vs. Scans



<u>Answering a similarity search query using different access paths</u>

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

Similarity Search

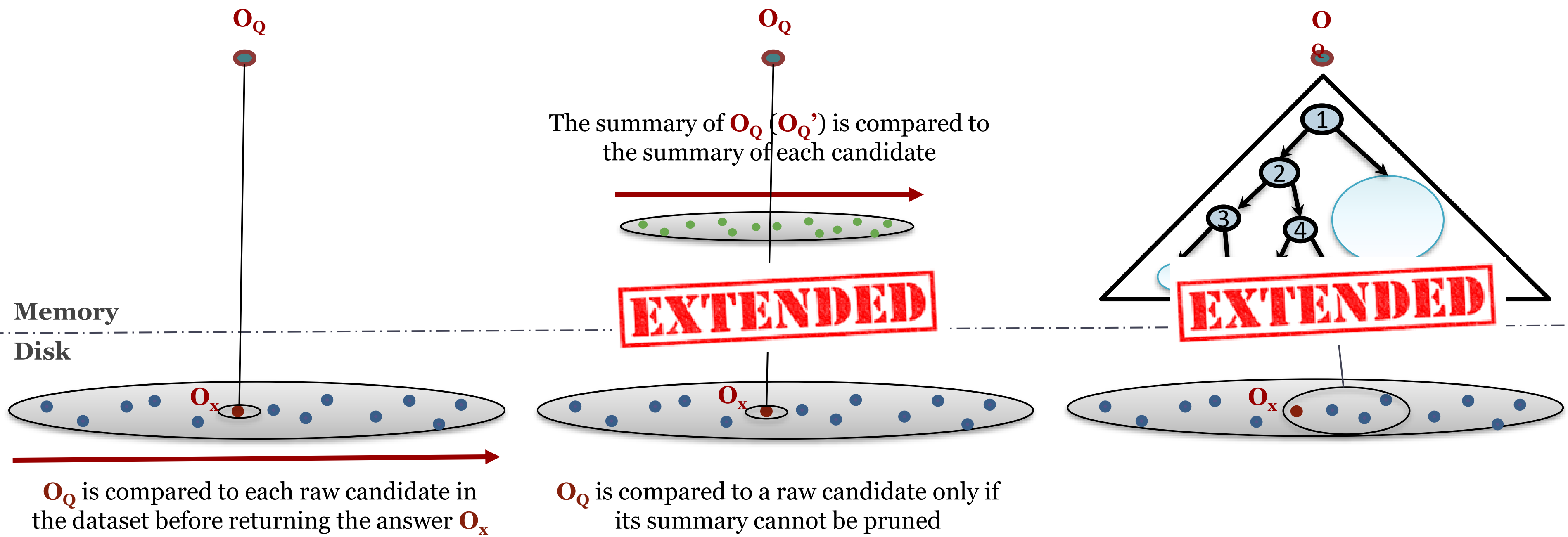
Data Series Extensions

Approximate Search

Publications

Echihabi-PVLDB‘19

Access Paths



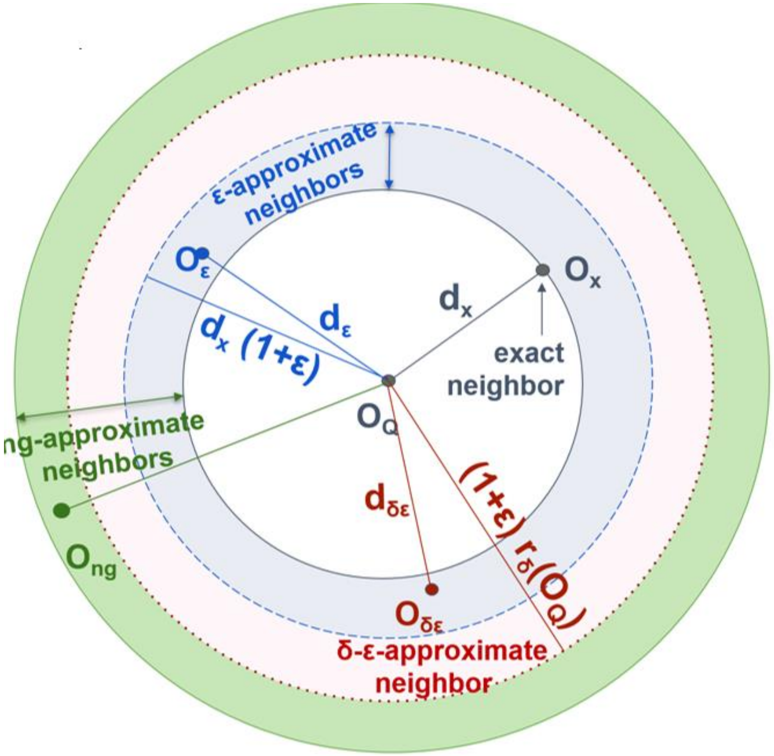
(a) Serial scan (b) Skip-sequential scan (c) Tree-based index

Answering a similarity search query using different access paths

Publications

Echihabi-PVLDB'19

Extensions: Skip-Sequential Scans

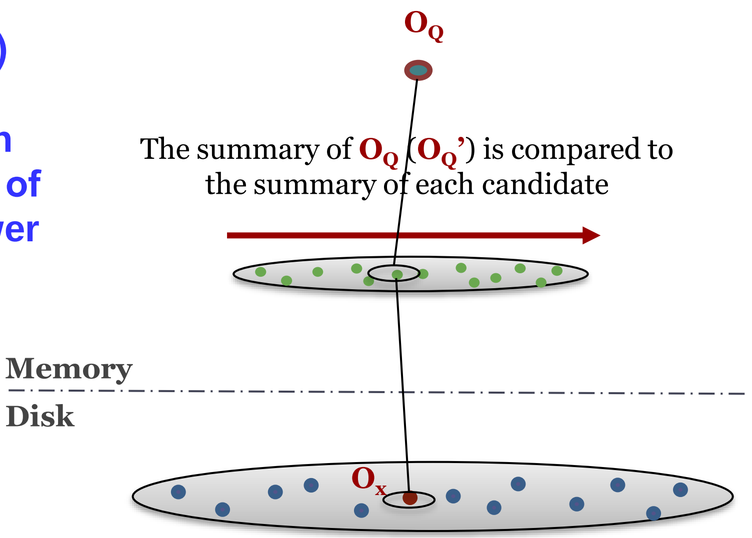


$d_\varepsilon <= d_x (1+\varepsilon)$

Result is within distance (1+ ε) of the exact answer

bsf $= d(O_Q, O_1)$

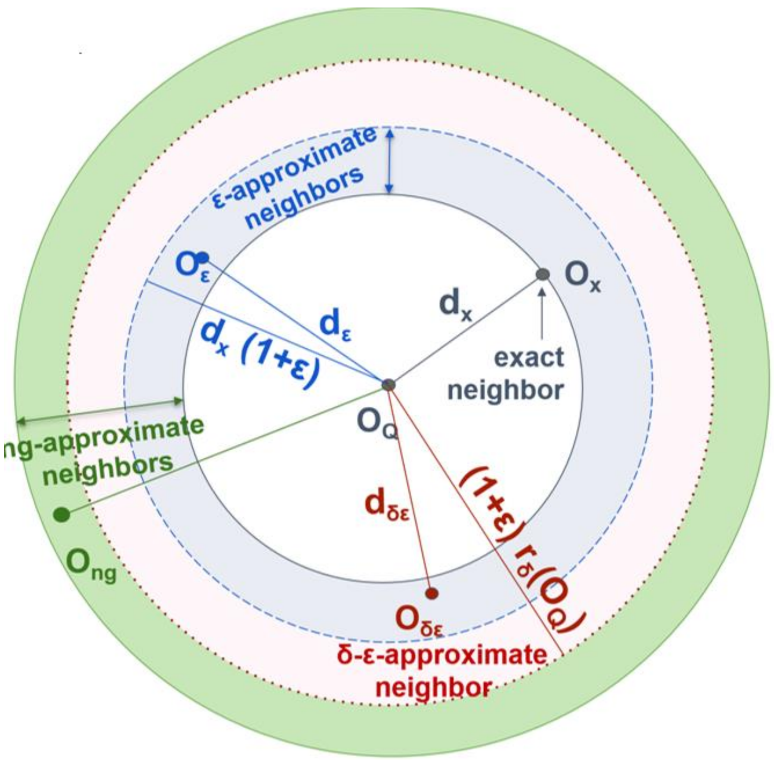
$lb_{cur} = d_{lb}(O_Q', O_x') <$ bsf



Publications

Echihabi-PVLDB‘19

Extensions: Skip-Sequential Scans



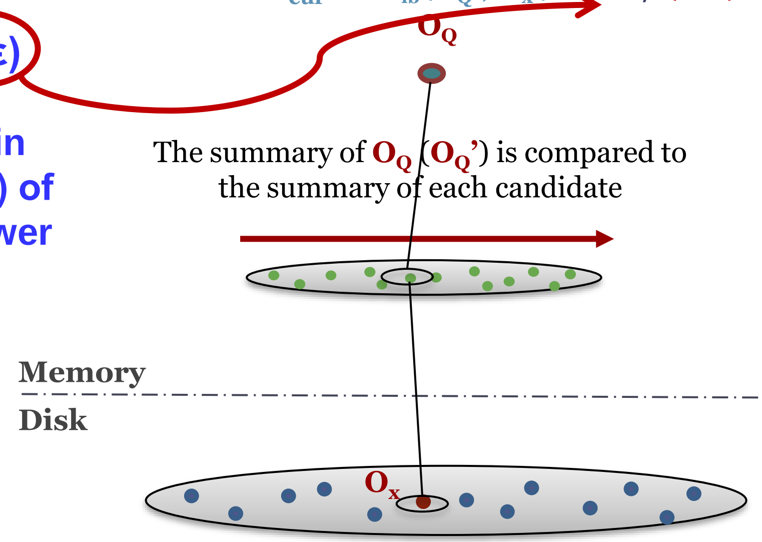
d_ε <= d_x(1+ε)

Result is within distance (1+ ε) of the exact answer

bsf = $d(O_Q, O_1)$

~~lb_{cur} = $d_{lb}(O_Q', O_x') <$ bsf~~

lb_{cur} = $d_{lb}(O_Q', O_x') <$ bsf / (1+ε)

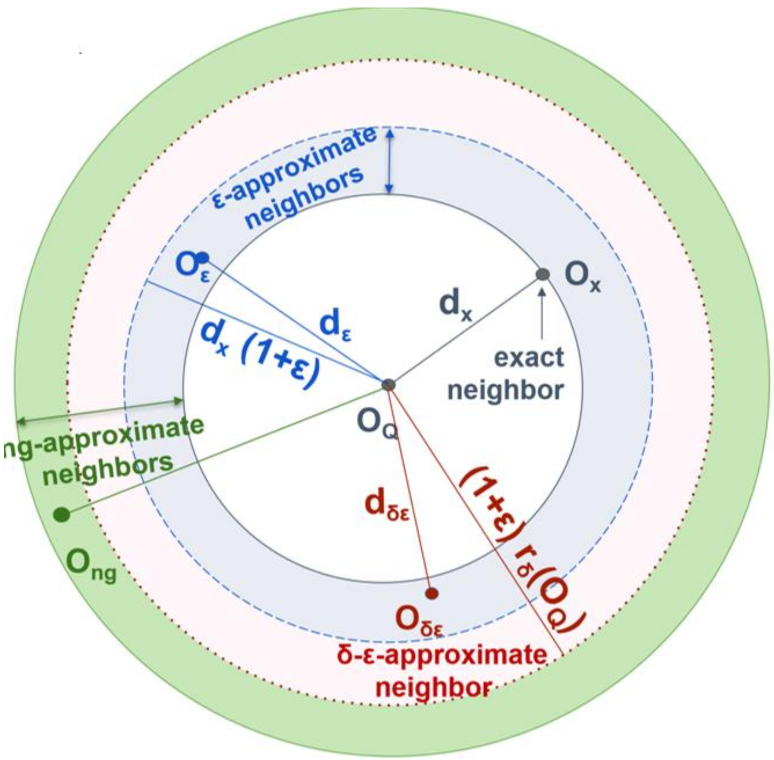


Publications

Echihabi-PVLDB'19

If $d_{lb}(O_Q', O_x') >= bsf / (1+\varepsilon)$
Then $bsf <= d(O_Q, O_x) (1+\varepsilon)$
i.e., $bsf <= d_x (1+\varepsilon)$

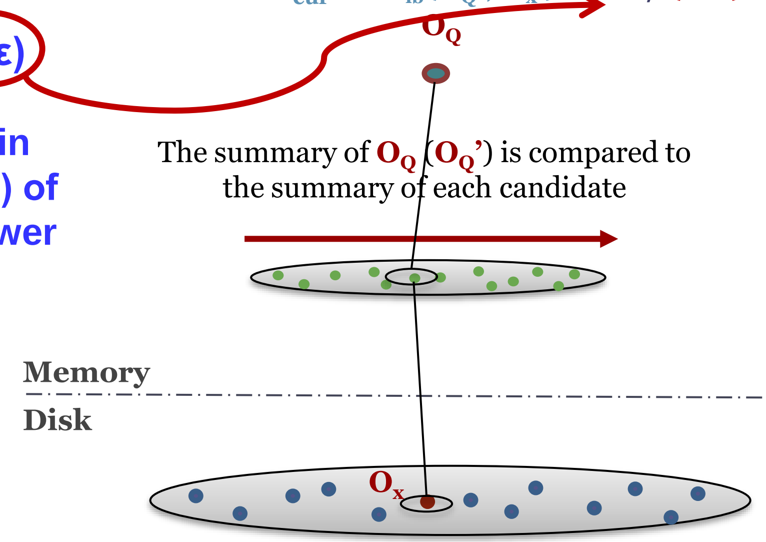
Extensions: Skip-Sequential Scans



$bsf \;\; = d(O_Q, O_1)$
$\cancel{lb_{cur} \;\; = d_{lb}(O_Q', O_x') < bsf}$
$lb_{cur} \;\; = d_{lb}(O_Q', O_x') < bsf / (1+\varepsilon)$

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

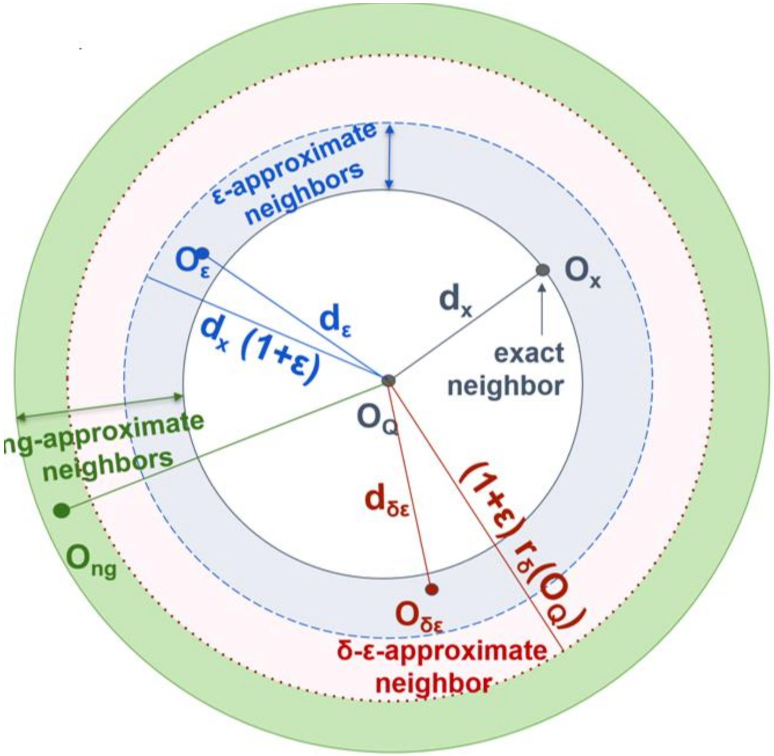
Result is within distance (1+ ε) of the exact answer



Publications

Echihabi-PVLDB'19

Extensions: Skip-Sequential Scans

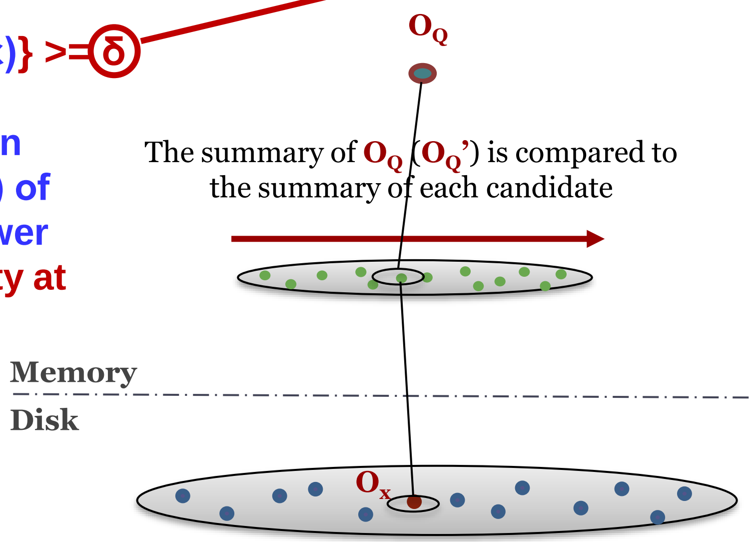


$P\{d_\varepsilon <= d_x (1+\varepsilon)\} >= \delta$

bsf $= d(O_Q,O_1)$

If bsf $<= (1+\varepsilon)\ r_\delta(O_Q)$ STOP

Result is within distance (1+ ε) of the exact answer with probability at least δ



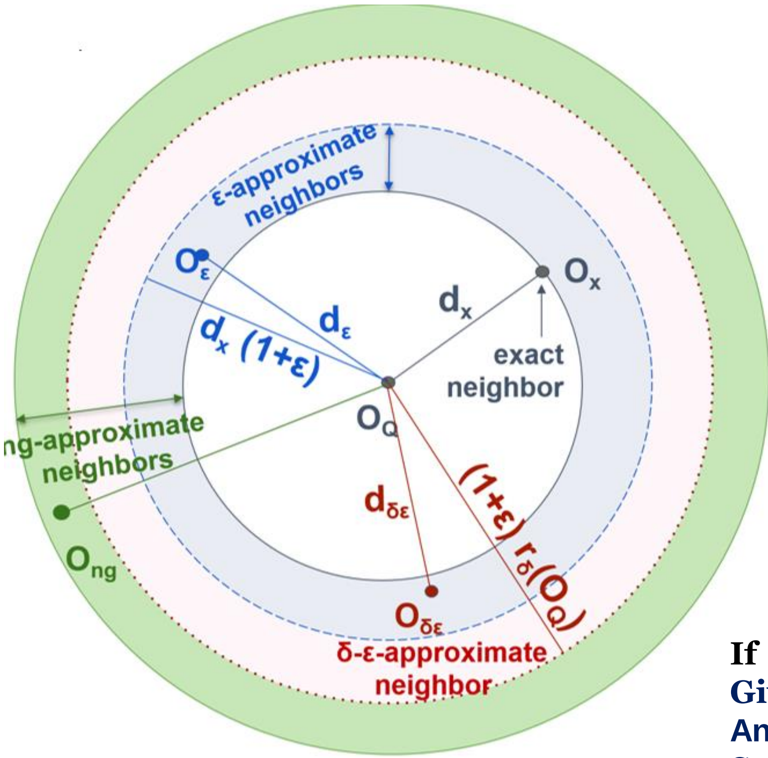
Publications

Echihabi-PVLDB'19

Extensions: Skip-Sequential Scans

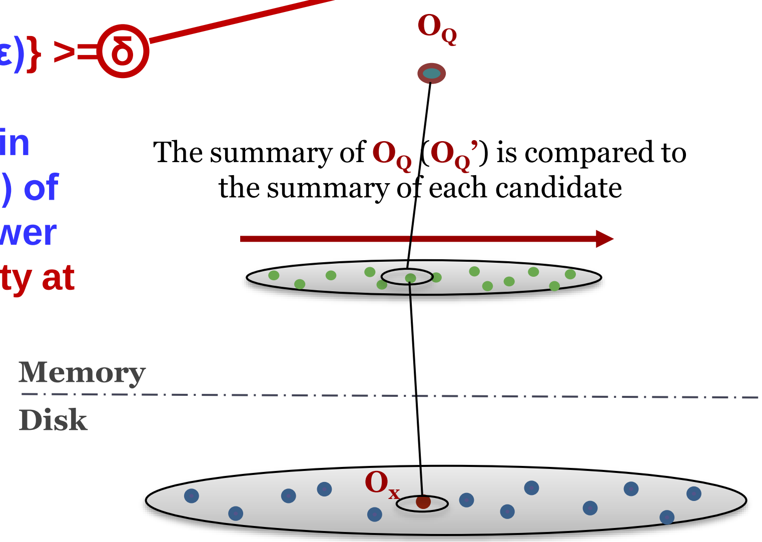
bsf $= d(O_Q, O_1)$

If bsf $<= (1+\varepsilon)\ r_\delta(O_Q)$ STOP



$P\{d_\varepsilon <= d_x (1+\varepsilon)\} >= \delta$

Result is within distance (1+ ε) of the exact answer with probability at least δ



If bsf $<= (1+\varepsilon)\ r_\delta(O_Q)$

Given that $P\{d_x > r_\delta(O_Q)\} >= \delta$, i.e, $P\{d_x <= r_\delta(O_Q)\} < 1-\delta$

And bsf / $(1+\varepsilon) <= r_\delta(O_Q)$ Then $P\{d_x <=$ bsf / $(1+\varepsilon)\} < 1-\delta$ *

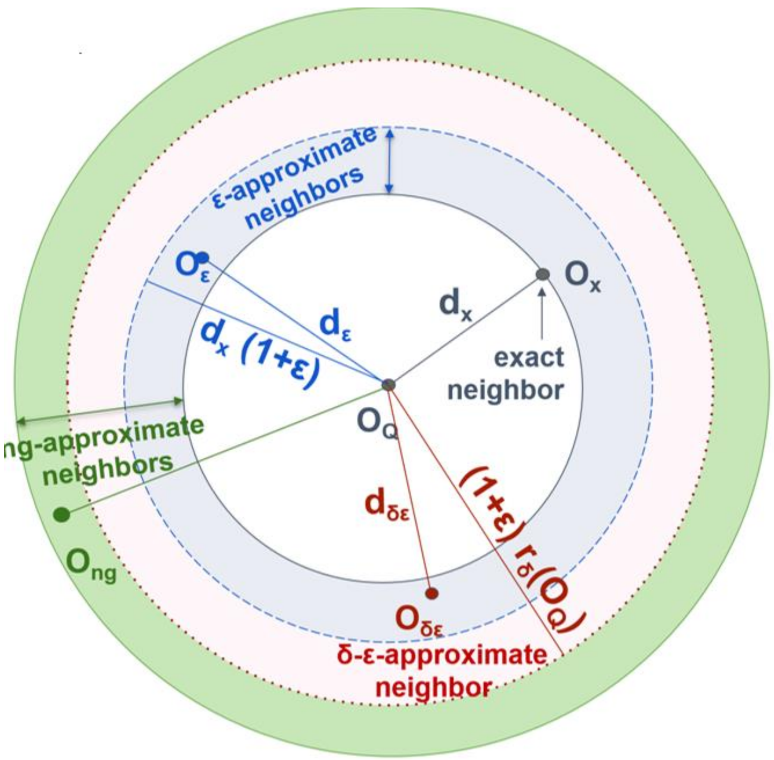
So $P\{d_x >$ bsf / $(1+\varepsilon)\} >= \delta$., i.e., $P\{$bsf $< (1+\varepsilon)\ d_x)\} >= \delta$

* We assume the monotonicity of the distribution of nearest neighbors of O_Q

Publications

Echihabi-PVLDB'19

Extensions: Tree Indexes

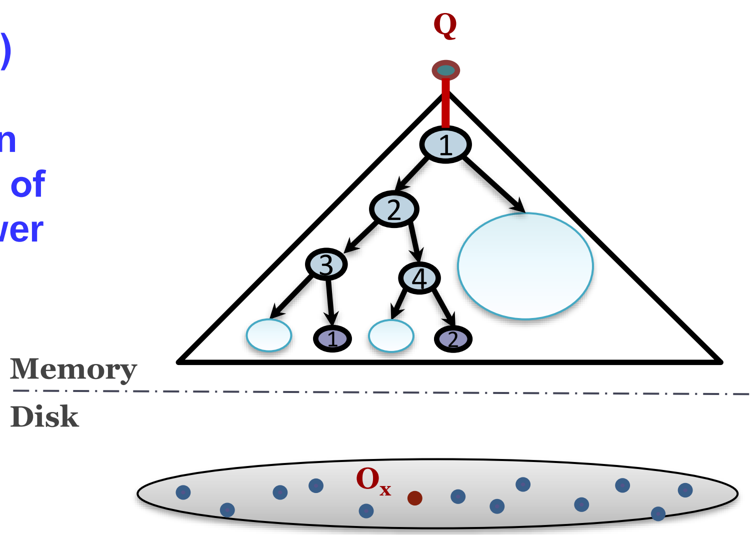


$d_\varepsilon \leq d_x\,(1+\varepsilon)$

Result is within distance (1+ ε) of the exact answer

$bsf = d(O_Q, O_3)$

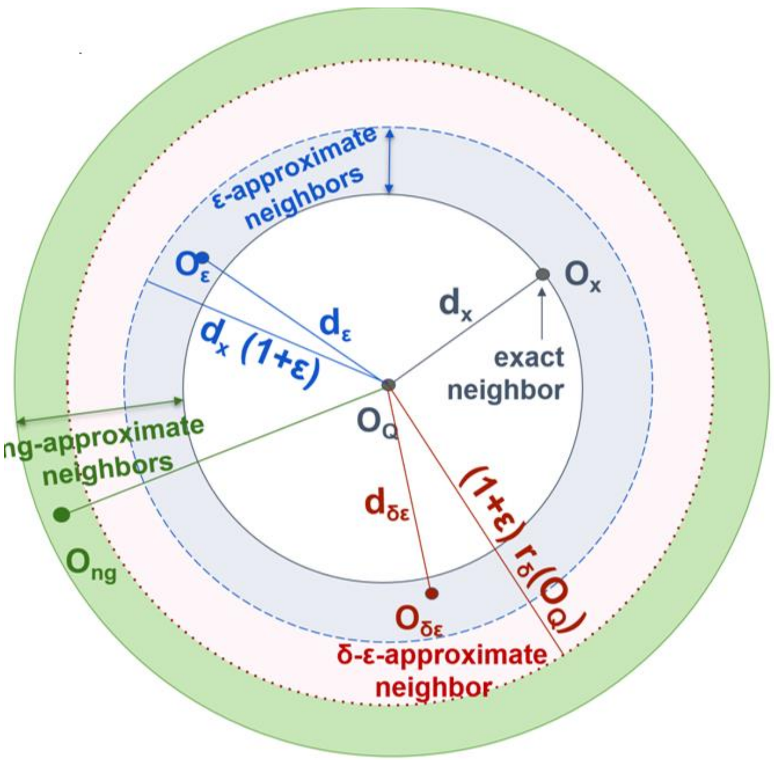
$lb_{cur} = d_{lb}(O_Q, 1) < bsf$



Publications

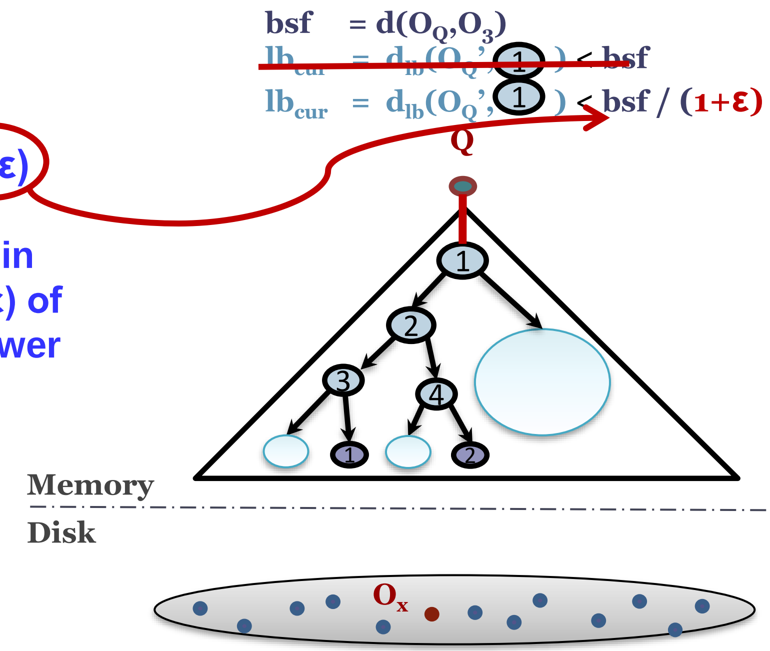
Echihabi-PVLDB'19

Extensions: Tree Indexes



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

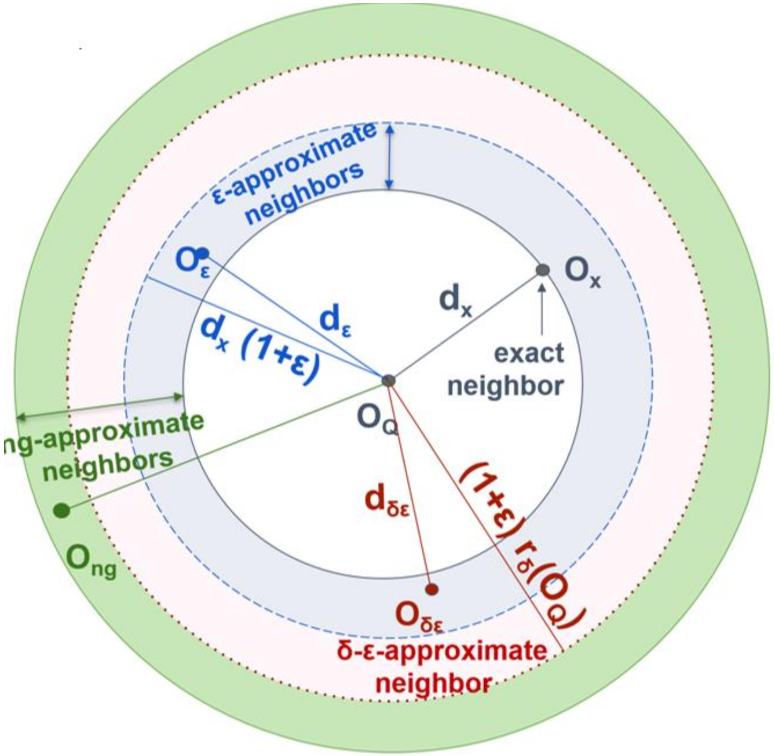
Result is within distance (1+ ε) of the exact answer



Publications

Echihabi-PVLDB‘19

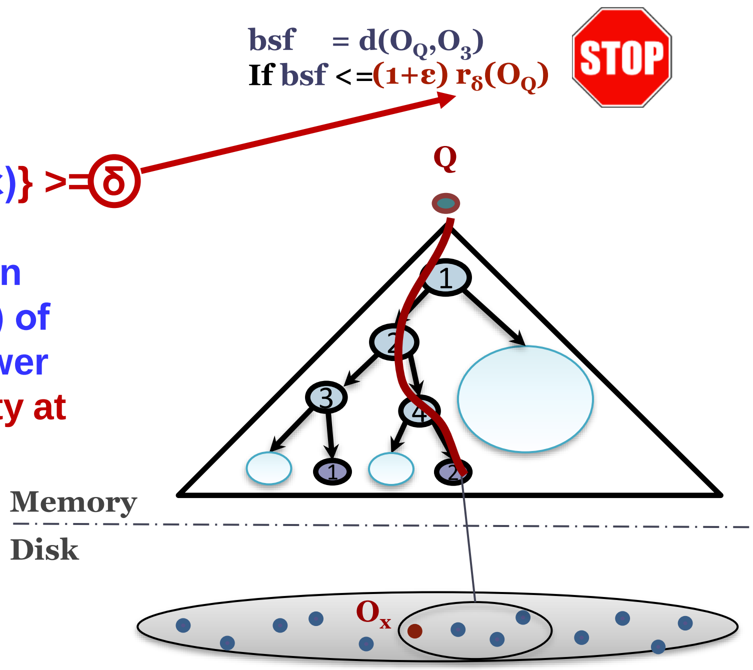
Extensions: Tree Indexes



$P\{d_\varepsilon <= d_x\ (1+\varepsilon)\} >= \delta$

Result is within distance (1+ ε) of the exact answer with probability at least δ

bsf = $d(O_Q,O_3)$
If bsf <= $(1+\varepsilon)\ r_\delta(O_Q)$



Publications

Echihabi-PVLDB'18

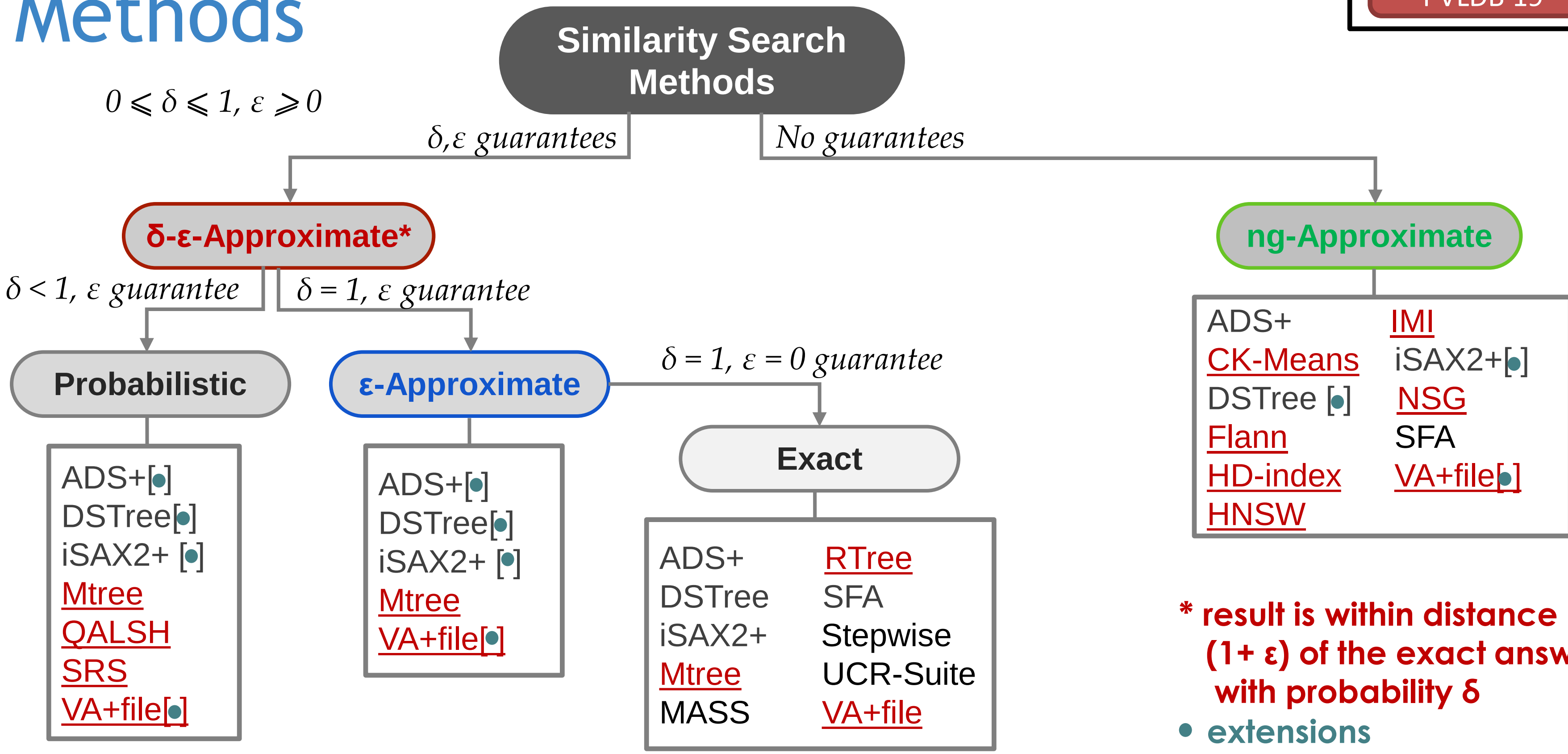
Echihabi-PVLDB'19

Techniques for data Series

Techniques for High-D vectors

Methods

$0 \leqslant \delta \leqslant 1, \varepsilon \geqslant 0$



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

• extensions

Questions?

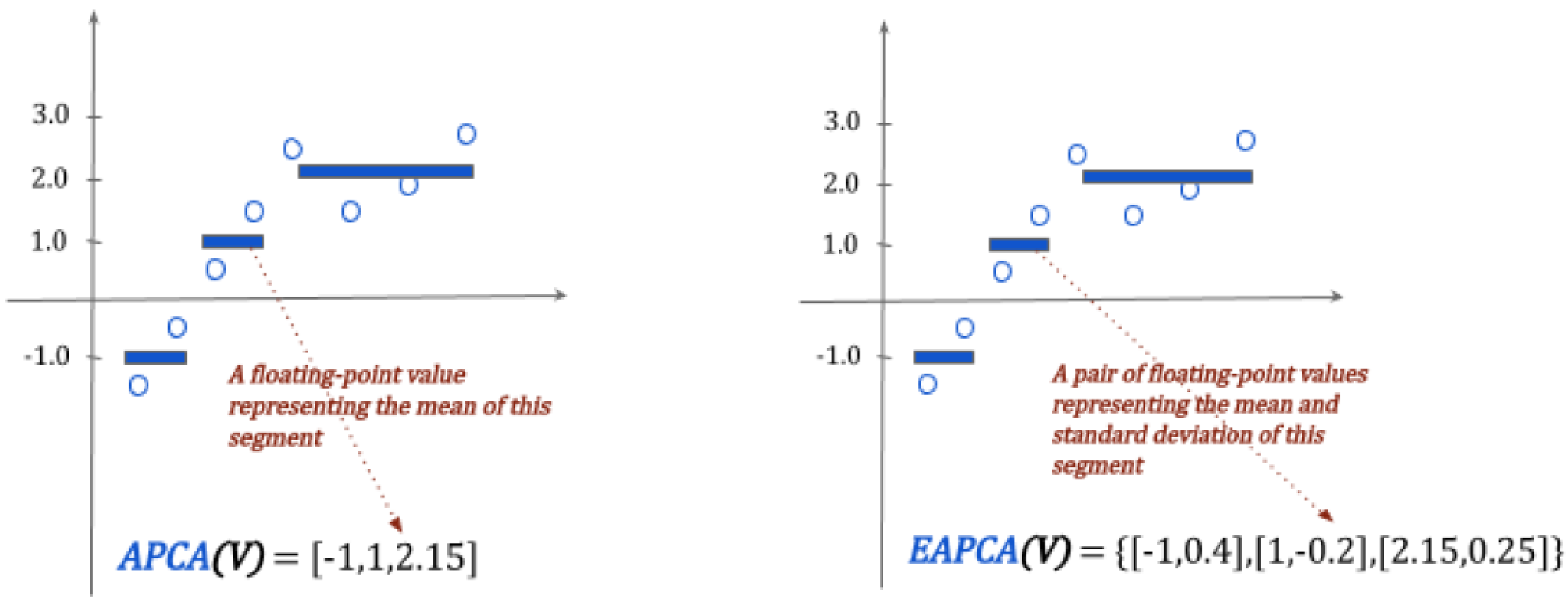
Data Series Indexing

DSTree
Summarization

Publications

Wang-PVLDB'13

V = [-1.5,-0.5, 0.5, 1.5, 2.5, 1.5,2, 2.6]



(a) APCA

(b) EAPCA

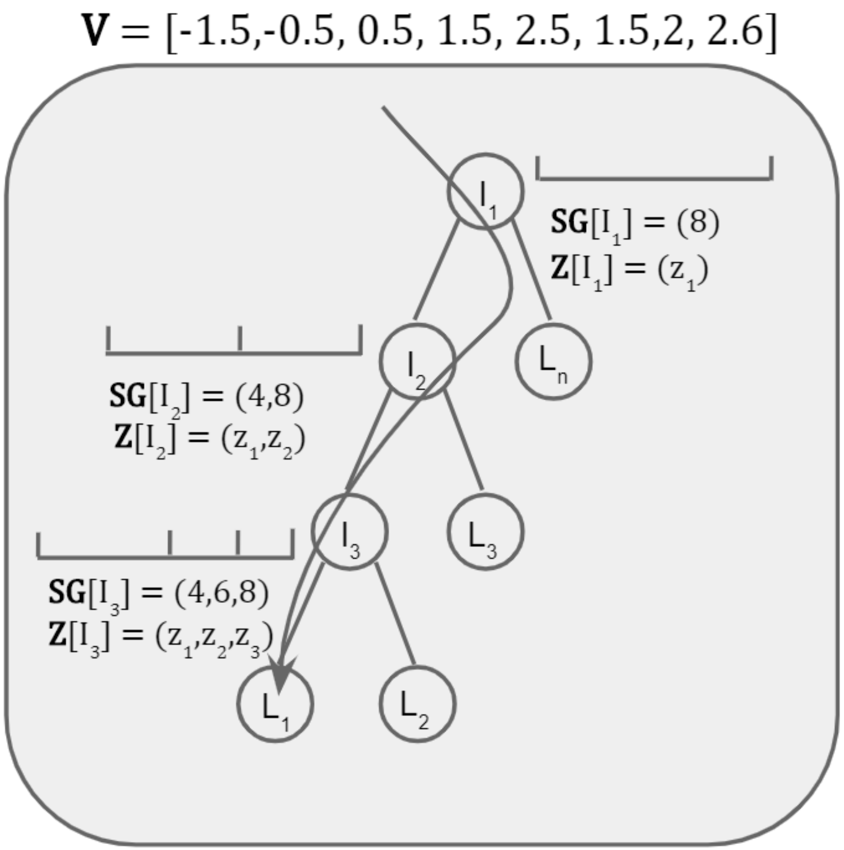
Intertwined with indexing

<u>The APCA and EAPCA representations</u>

Publications

Wang-PVLDB'13

DSTree
Indexing



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

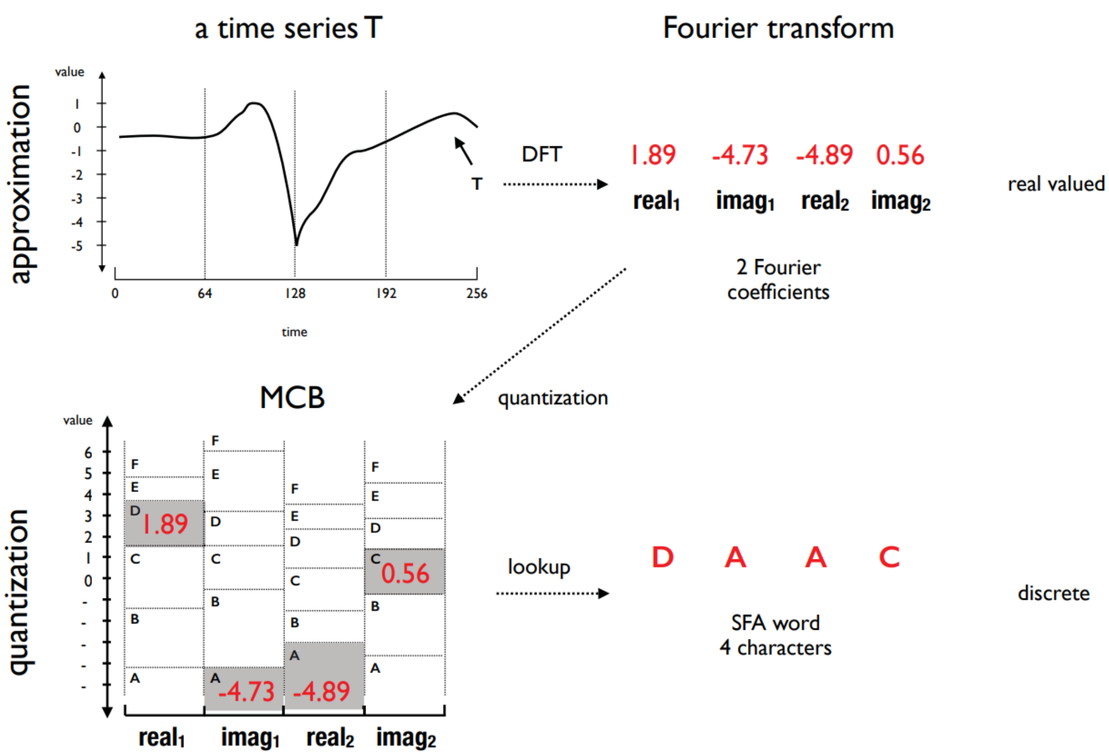
Each Leaf node also :
- ❑ stores its raw vectors in a separate disk file

Symbolic Fourier Approximation (SFA)

Summarization

Publications

Schafer-EDBT'12



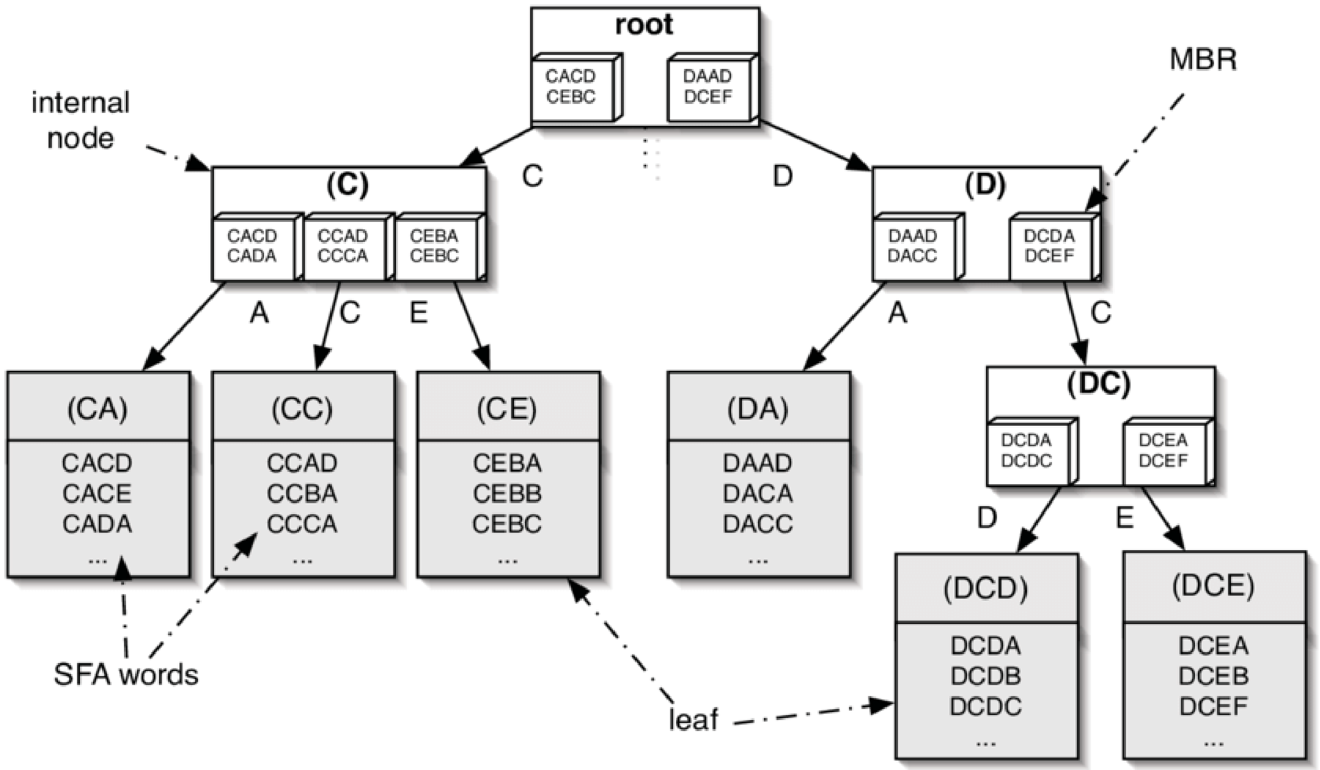
The SFA representation*

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

SFA Indexing

Publications

Schafer-EDBT'12



The SFA Trie*

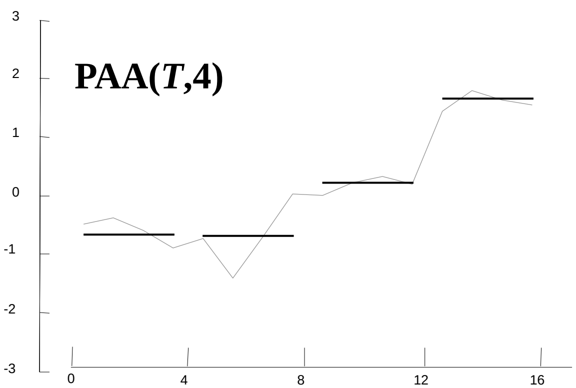
*https://www2.informatik.hu-berlin.de/~schaefpa/talks/scalable_classification.pptx

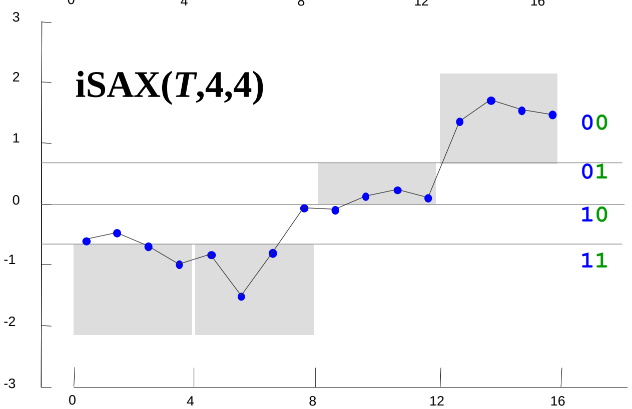
Publications

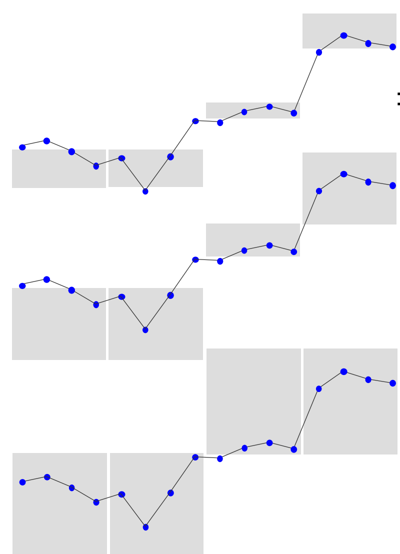
Shieh-KDD‘08

iSAX Family
iSAX Summarization

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







= {110 ,110 ,0111 ,000}

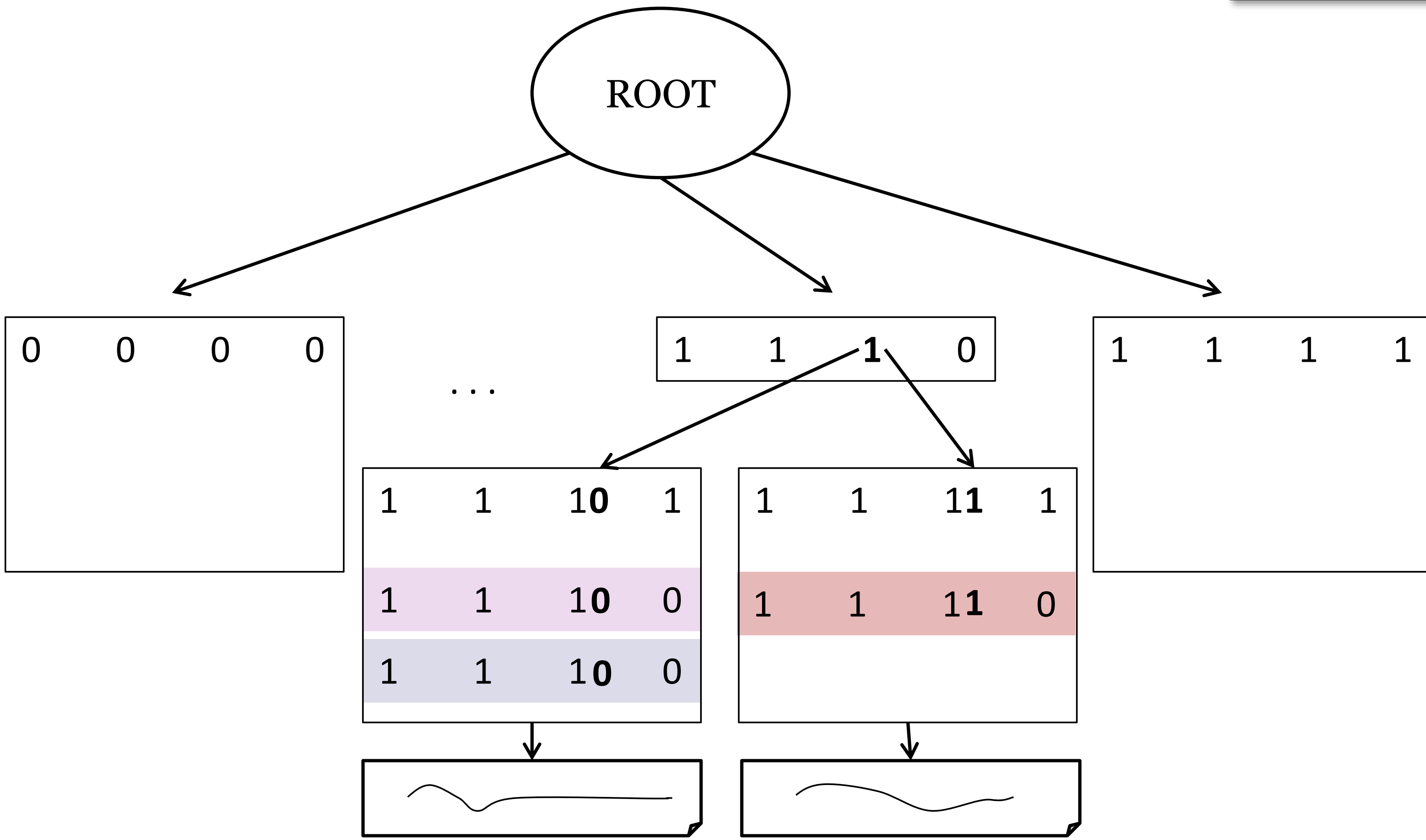
= {11 ,11 ,011 ,00 }

= {1 ,1 ,0 ,0 }

iSAX-Indexing

Publications

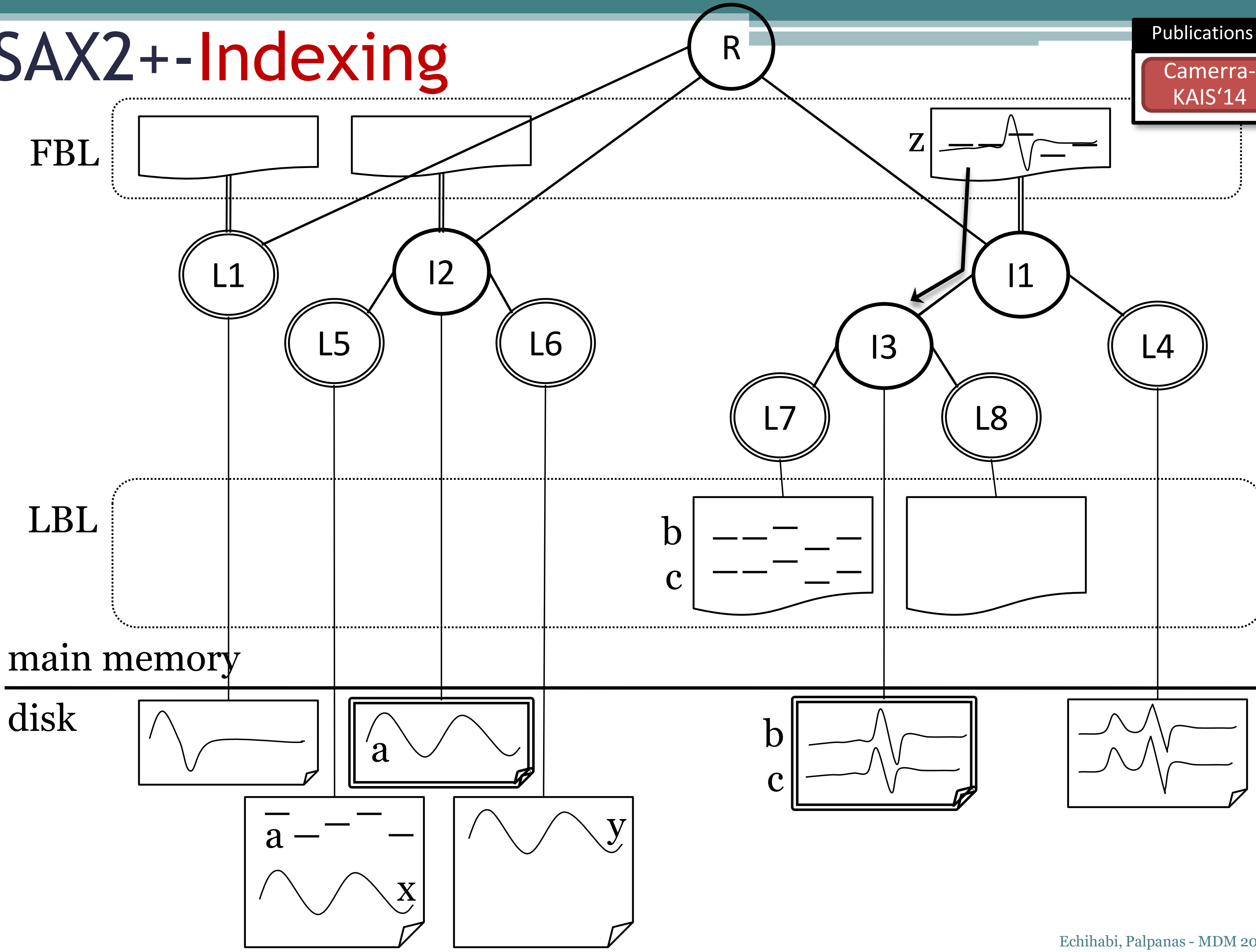
Shieh-KDD'08



iSAX2+-Indexing

Publications

Camerra-KAIS'14



Publications

Zoumbatianos-SIGMOD'14

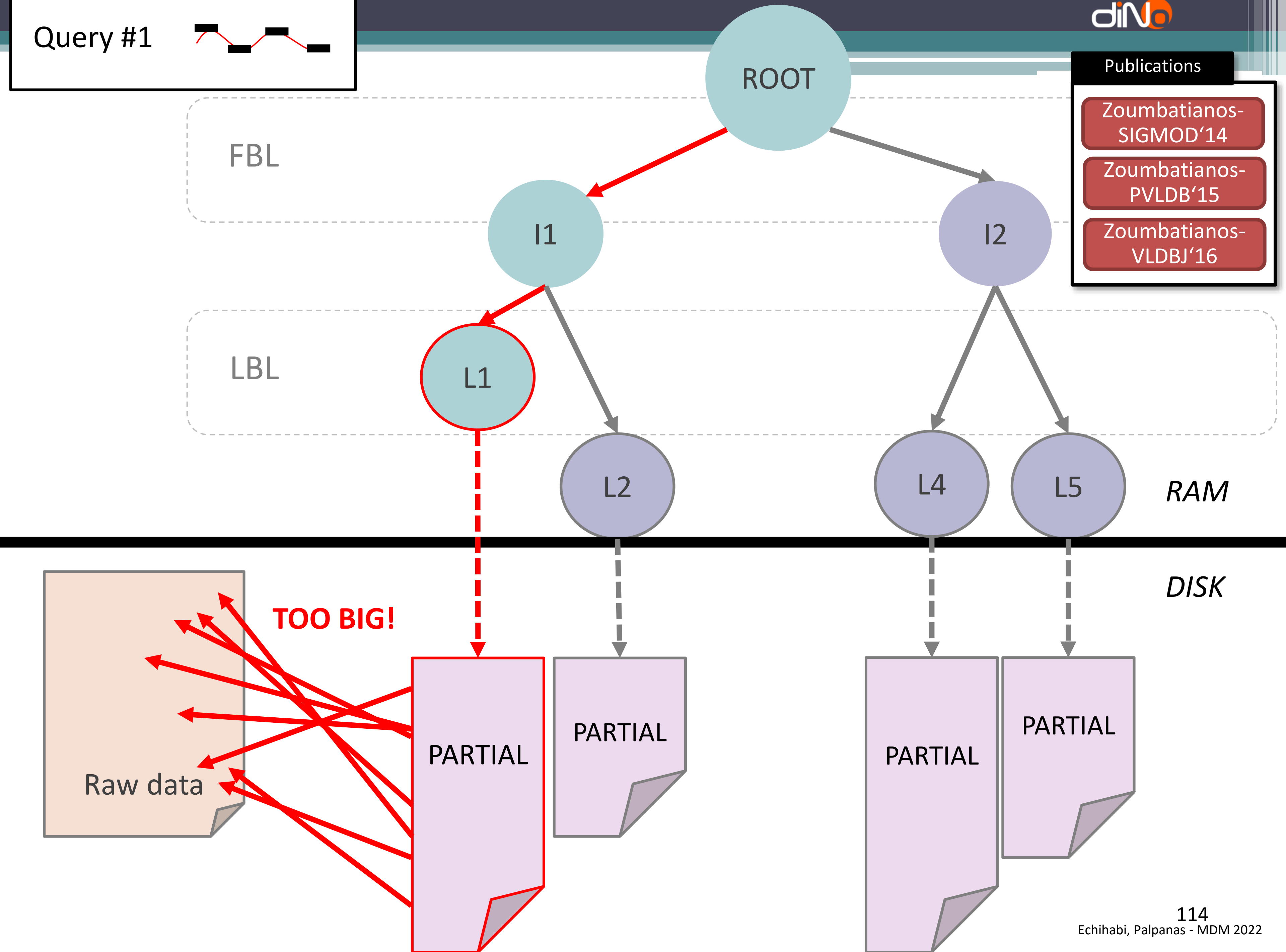
Zoumbatianos-PVLDB'15

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

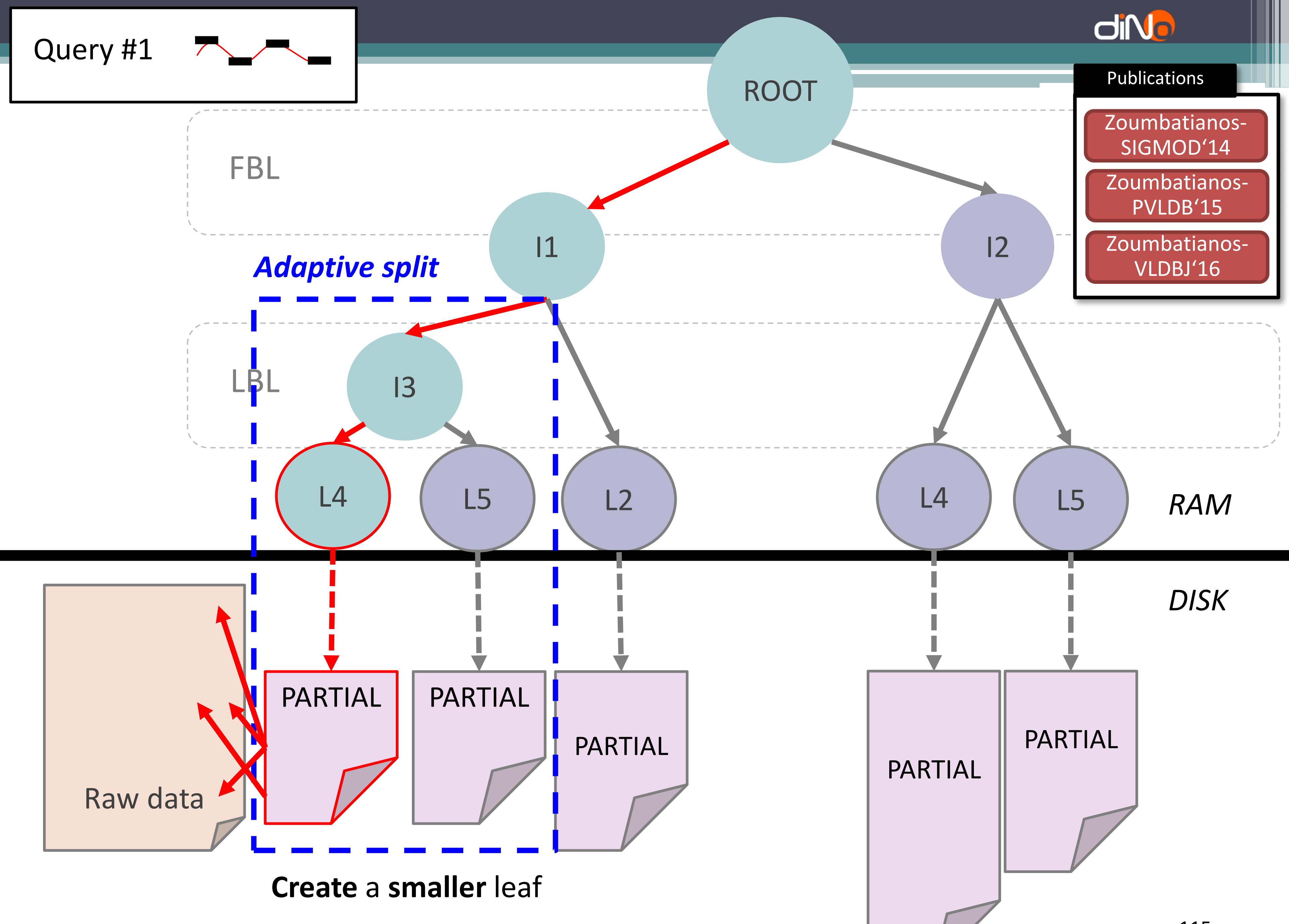


Publications

Zoumbatianos-SIGMOD‘14

Zoumbatianos-PVLDB‘15

Zoumbatianos-VLDBJ‘16

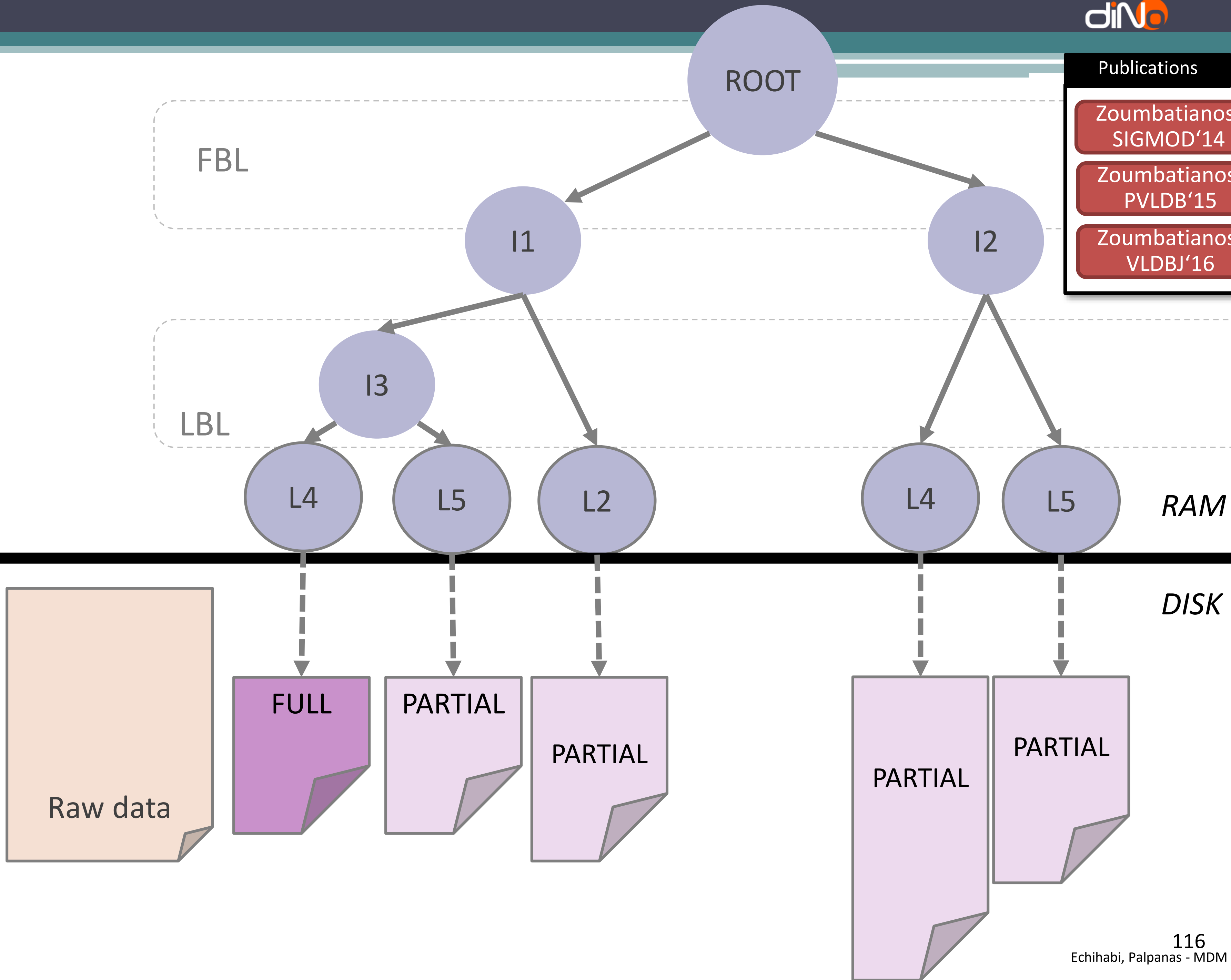

Query #1
Publications
Zoumbatianos-SIGMOD'14
Zoumbatianos-PVLDB'15
Zoumbatianos-VLDBJ'16
ROOT
FBL
I1
I2
Adaptive split
LBL
I3
L4
L5
L2
L4
L5
RAM
DISK
Raw data
PARTIAL
PARTIAL
PARTIAL
PARTIAL
PARTIAL
Create a **smaller** leaf

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

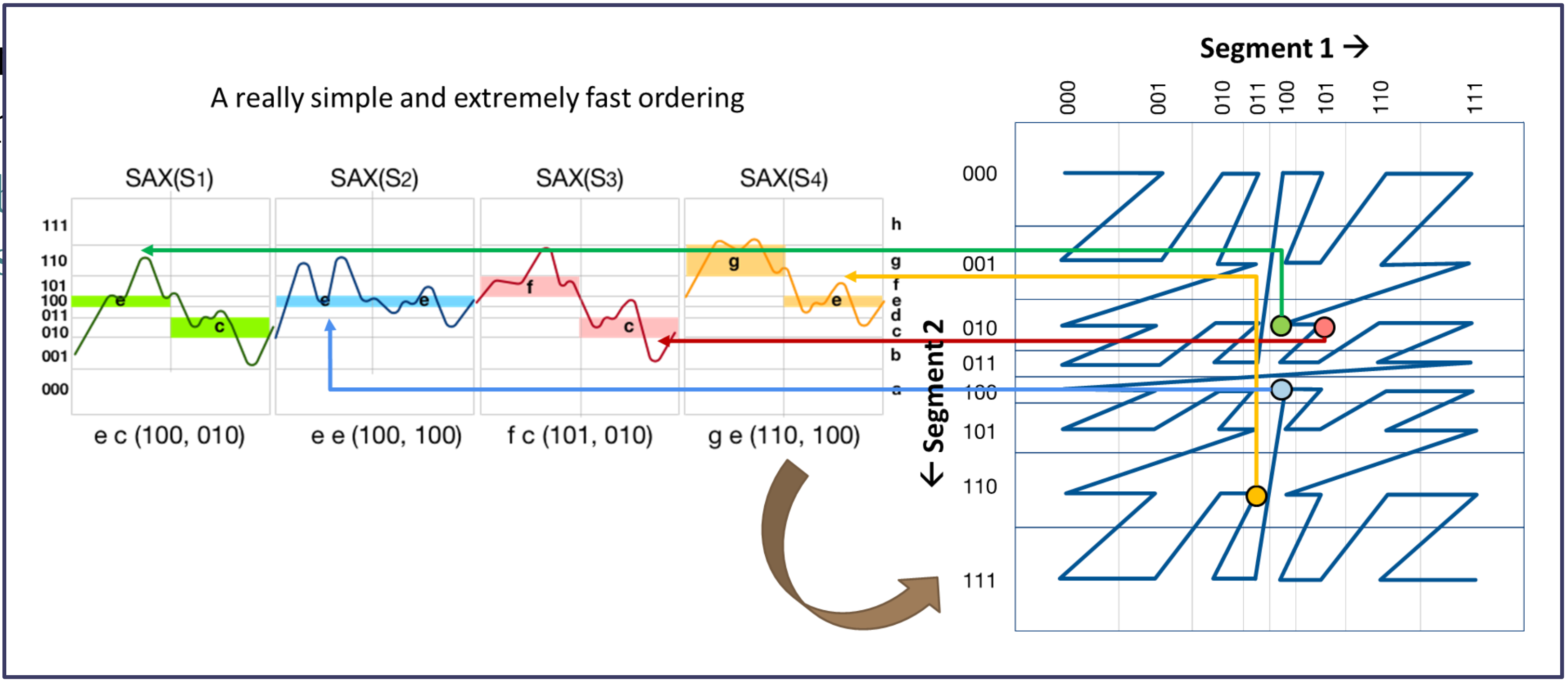
Publications

Kondylakis-PVLDB'18

Kondylakis-SIGMOD'19

Coconut

- cu
  tim
  - I
    s



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

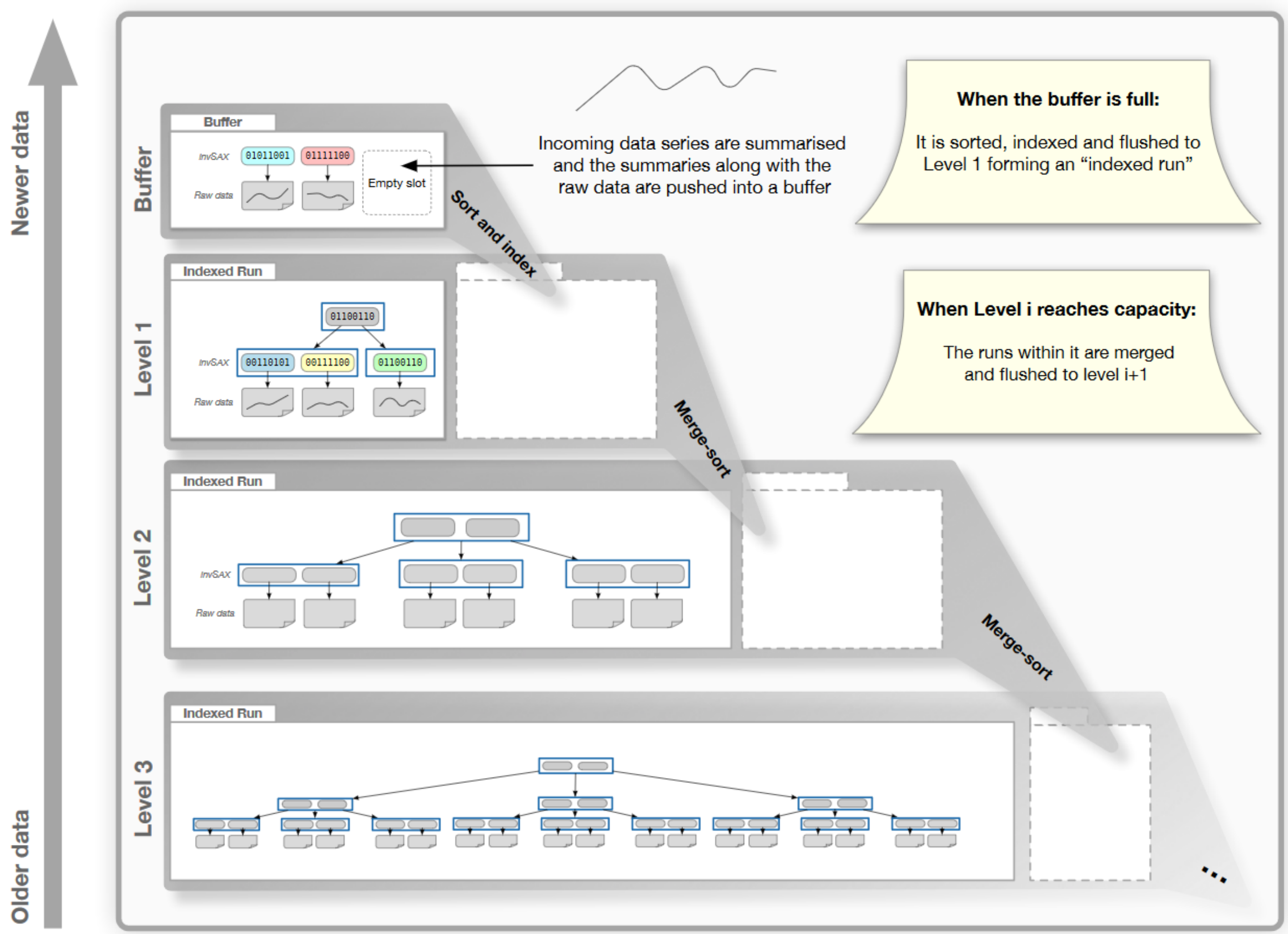
Coconut

- 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, …

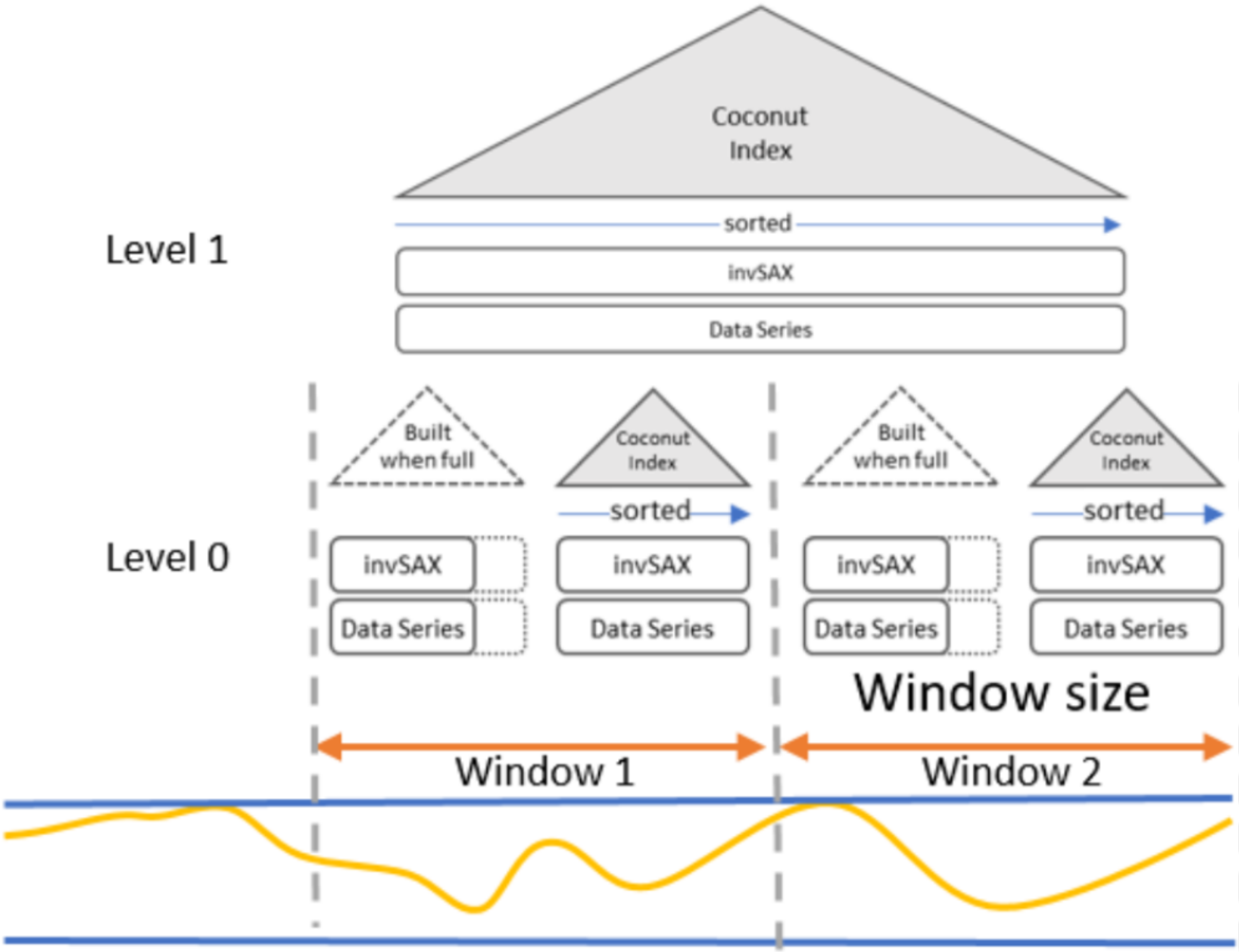
Publications

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

Coconut-LSM



Coconut-LSM



Publications

- Kondylakis-PVLDB'18
- Kondylakis-SIGMOD'19
- Kondylakis-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

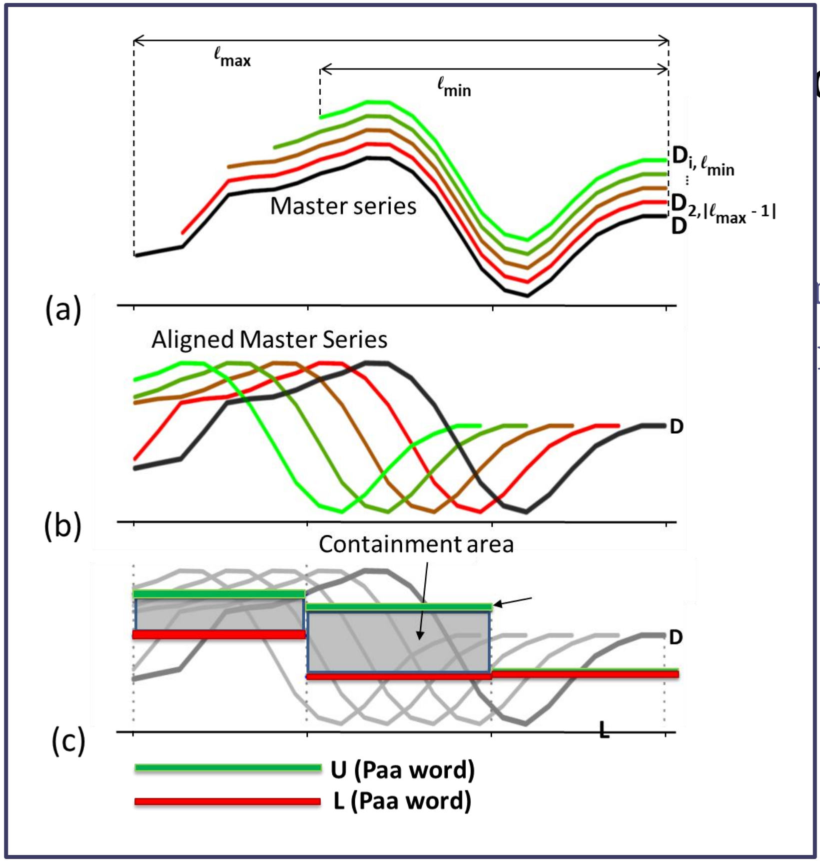
Publications

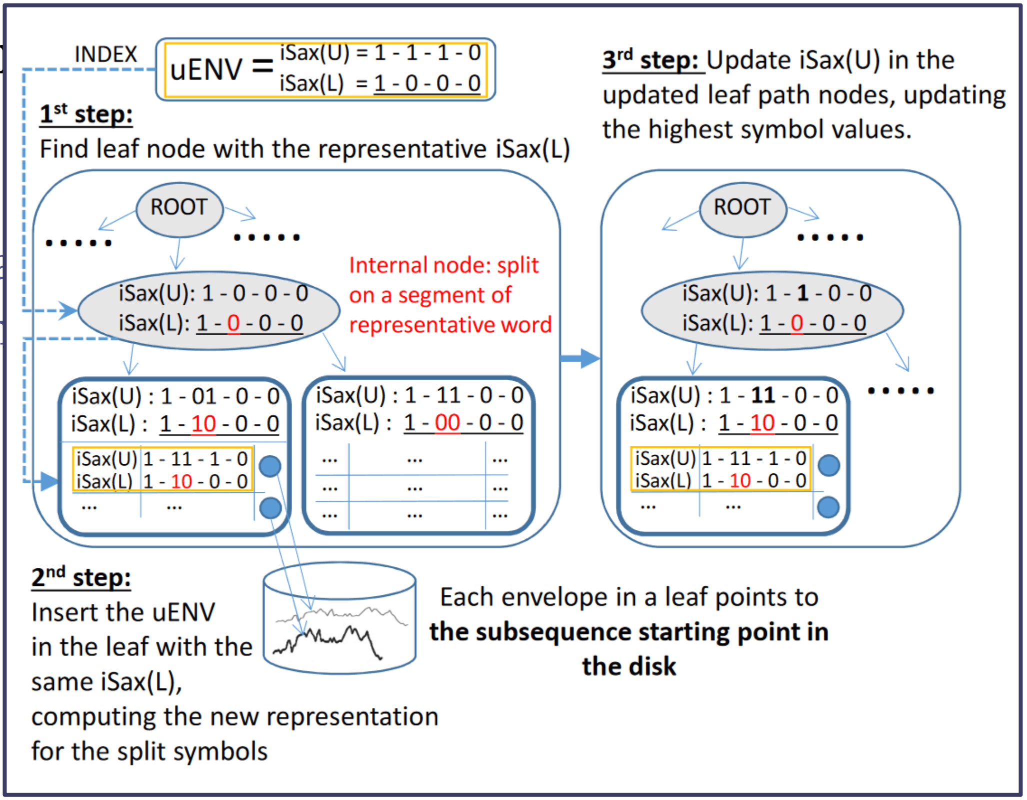
Linardi-ICDE'18

Linardi-PVLDB'19

Linardi-VLDBJ'20

ULISSE





Publications

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

ULISSE

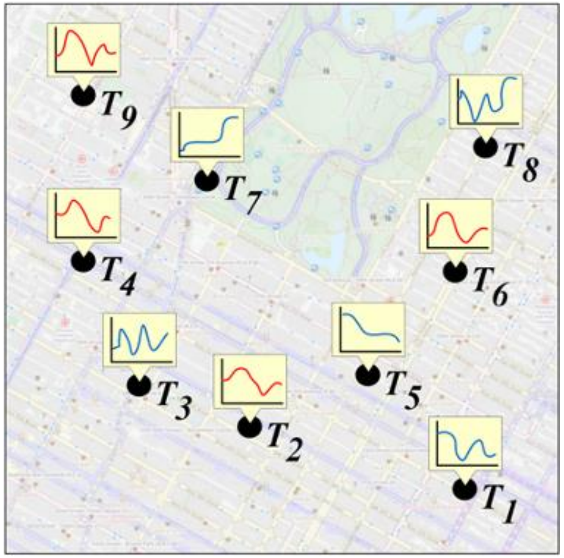
- ULISSE: current solution for variable-length queries
  - single-index support for
    - queries of variable lengths
    - Z-normalized + non Z-normalized data
    - Euclidean + DTW distance measures
  - orders of magnitude faster than competing approaches

Publications

Chatzigeorgakidis et al. SIGSPATIAL/GIS'17

Geolocated Data Series

- search both on spatial proximity and data series similarity



Publications

Chatzigeorgakidis et al.
SIGSPATIAL/GIS'17

Geolocated Data Series

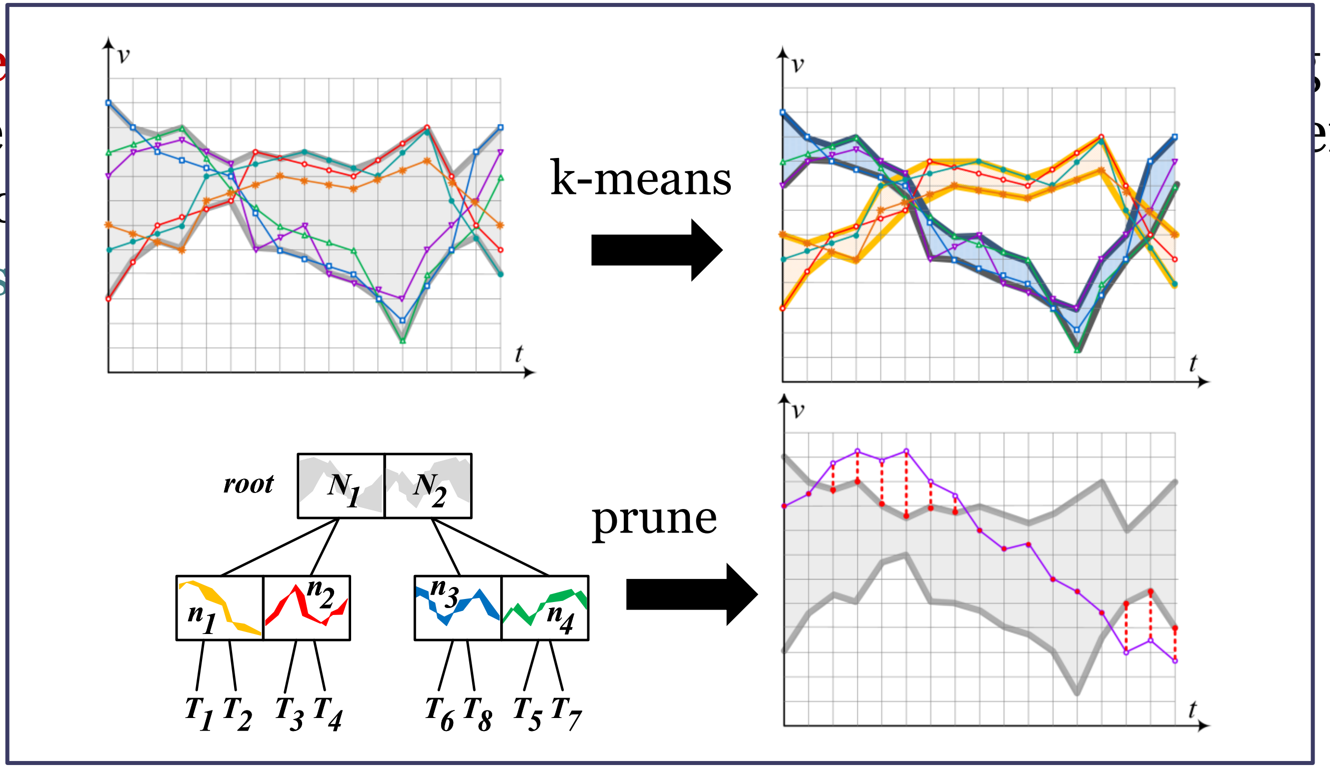
- search both on spatial proximity and data series similarity
- BTSR-Tree: hybrid index that combines Minimum Bounding Rectangles (MBR) and bundled Minimum Bounding Time Series (MBTS) to prune the search space
  - prunes subtrees that cannot contain any results

Publications

Chatzigeorgakidis et al. SIGSPATIAL/GIS'17

Geolocated Data Series

- search both on spatial proximity and data series similarity
- BTSR-Tre … Rectangle … eries (MBTS) to
  - prunes s

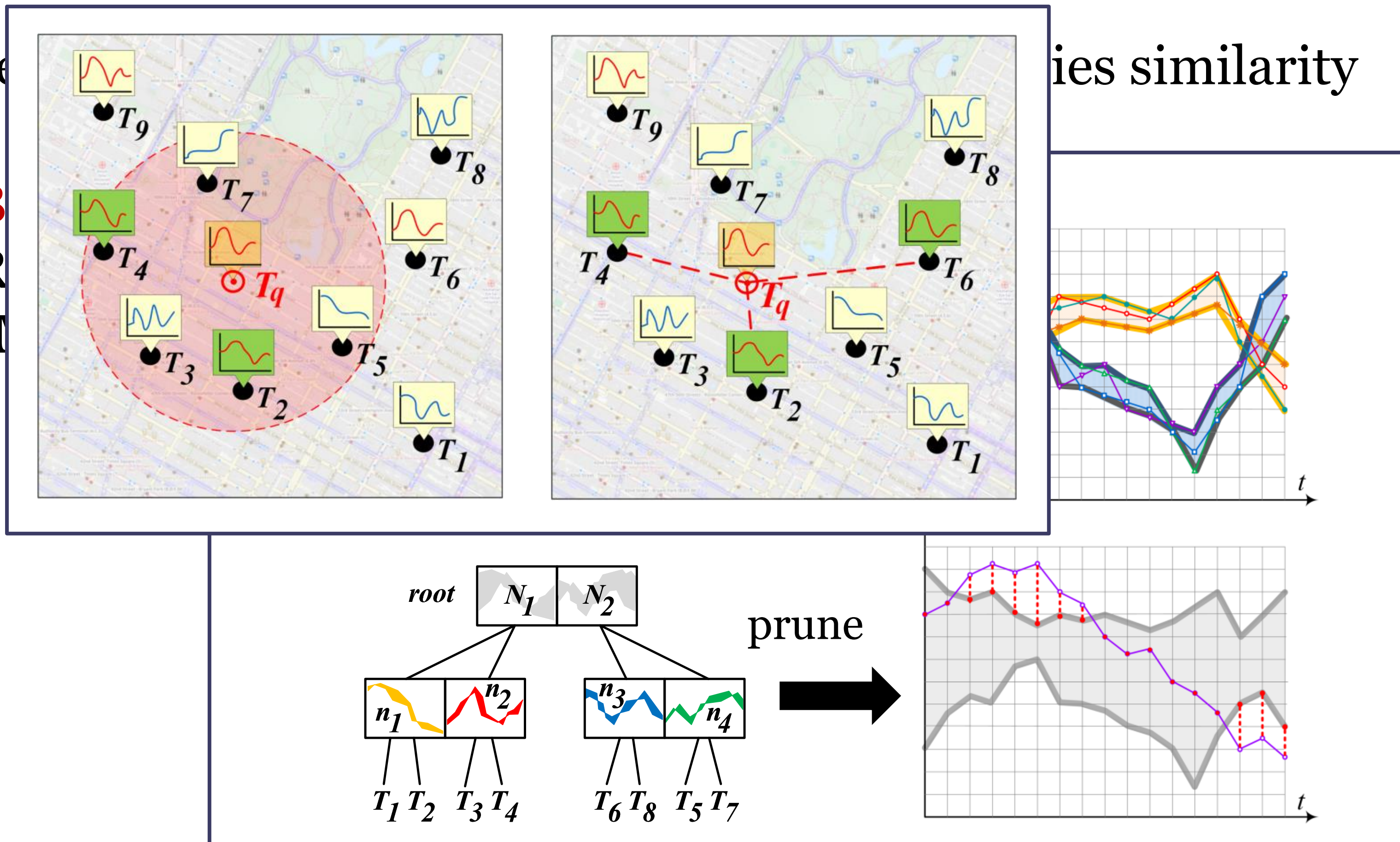


Publications

Chatzigeorgakidis et al. SIGSPATIAL/GIS'17

Geolocated Data Series

- se… …ies similarity
- B… R… (I… …eries



Publications

Chatzigeorgakidis et al. SIGSPATIAL/GIS'17

Chatzigeorgakidis et al. SIGSPATIAL/GIS'18

Geolocated Data Series

- search both on spatial proximity and data series similarity
- BTSR-Tree: hybrid index that combines Minimum Bounding Rectangles (MBR) and bundled Minimum Bounding Time Series (MBTS) to prune the search space
  - prunes subtrees that cannot contain any results
- HSJ: hybrid similarity join on geolocated data series using the BTSR-Tree
  - per- and cross-partition search in parallel (adjacent bands/boxes)

Publications

Chatzigeorgakidis et al. SIGSPATIAL/GIS'17

Chatzigeorgakidis et al. SIGSPATIAL/GIS'18

Chatzigeorgakidis et al. Elsev. Big Data Res. '19

Geolocated Data Series

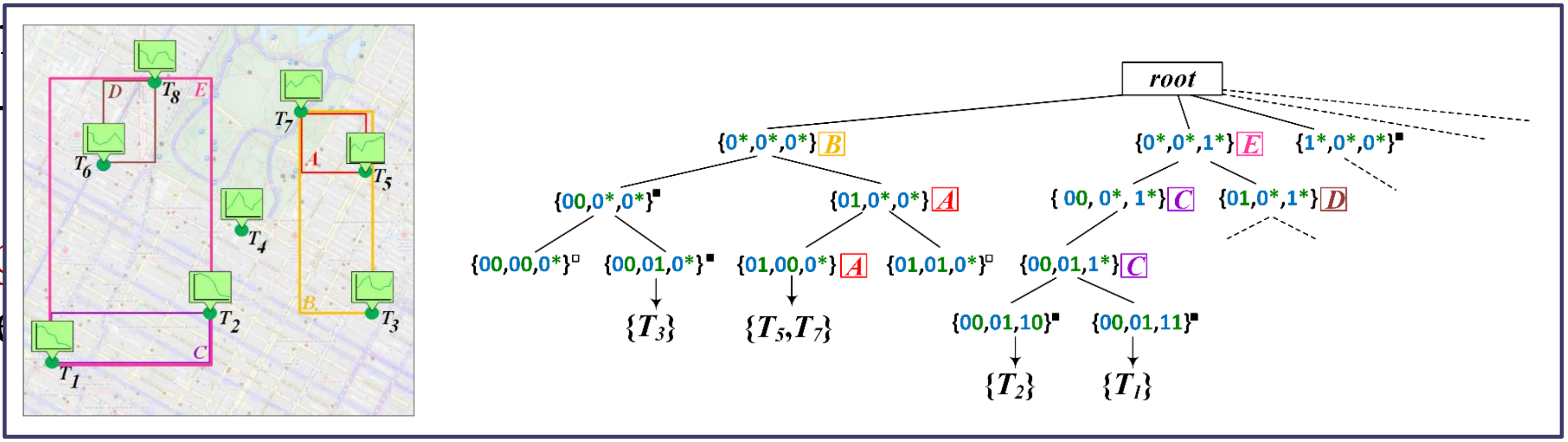
- search both on spatial proximity and data series similarity
- BTSR-Tree: hybrid index that combines Minimum Bounding Rectangles (MBR) and bundled Minimum Bounding Time Series (MBTS) to prune the search space
  - prunes subtrees that cannot contain any results
- HSJ: hybrid similarity join on geolocated data series using the BTSR-Tree
  - per- and cross-partition search in parallel (adjacent bands/boxes)
- VisExp: interactive visual exploration on geolocated data series using either geo-iSAX or BTSR-Tree
  - geo-iSAX: iSAX index - nodes augmented with MBR data

Publications

- Chatzigeorgakidis et al. SIGSPATIAL/GIS'17
- Chatzigeorgakidis et al. SIGSPATIAL/GIS'18
- Chatzigeorgakidis et al. Elsev. Big Data Res. '19

Geolocated Data Series

- search both on spatial proximity and data series similarity
- BTSR-Tree: hybrid index that combines Minimum Bounding Rectangles (MBR) and bundled Minimum Bounding Time Series (MBTS) to prune the search space
  - prunes subtrees that cannot contain any results
- HSJ: h
  BTSR-
  - per-
- VisExp
  using
  - geo-

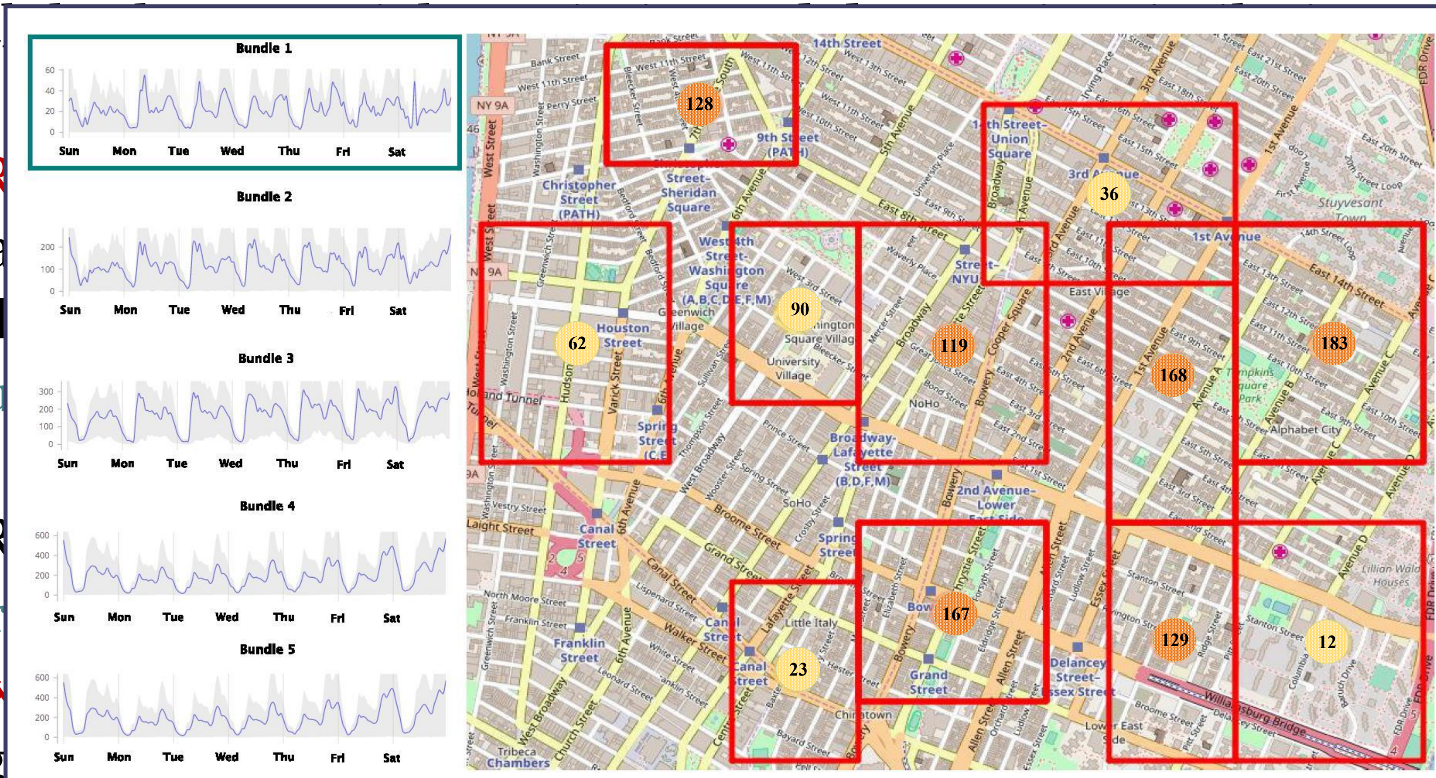


Publications

- Chatzigeorgakidis et al. SIGSPATIAL/GIS'17
- Chatzigeorgakidis et al. SIGSPATIAL/GIS'18
- Chatzigeorgakidis et al. Elsev. Big Data Res. '19

Geolocated Data Series

- searc
- BTSR
  Recta ries
  (MBT
  - pru
- HSJ:
  BTSR
  - per
- VisEx
  using
  - geo-



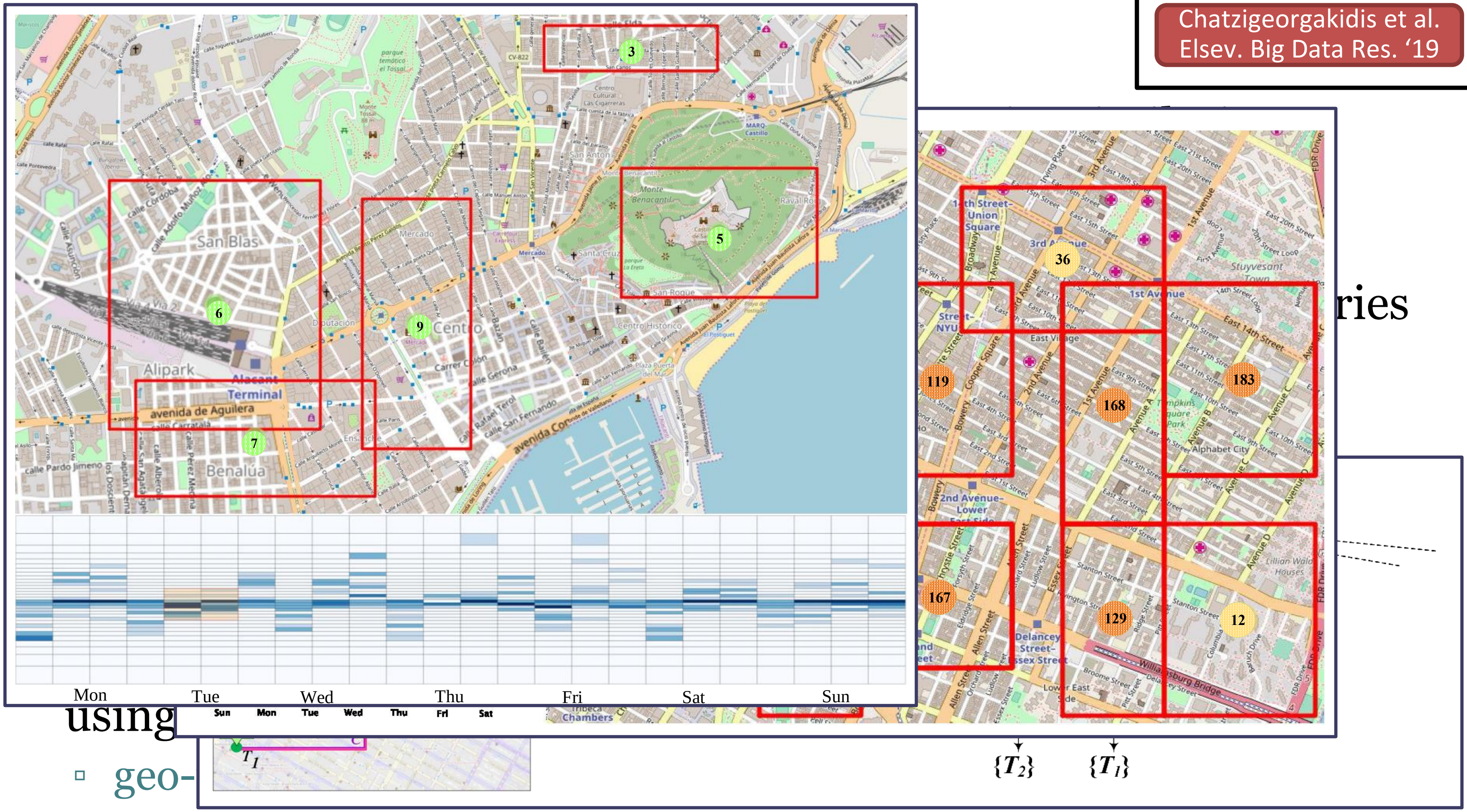
Geolocated Data Series

Publications

Chatzigeorgakidis et al. SIGSPATIAL/GIS'17

Chatzigeorgakidis et al. SIGSPATIAL/GIS'18

Chatzigeorgakidis et al. Elsev. Big Data Res. '19



ries

using

▫ geo-

Publications

Chatzigeorgakidis et al.
SIGSPATIAL/GIS'20

spaTScope

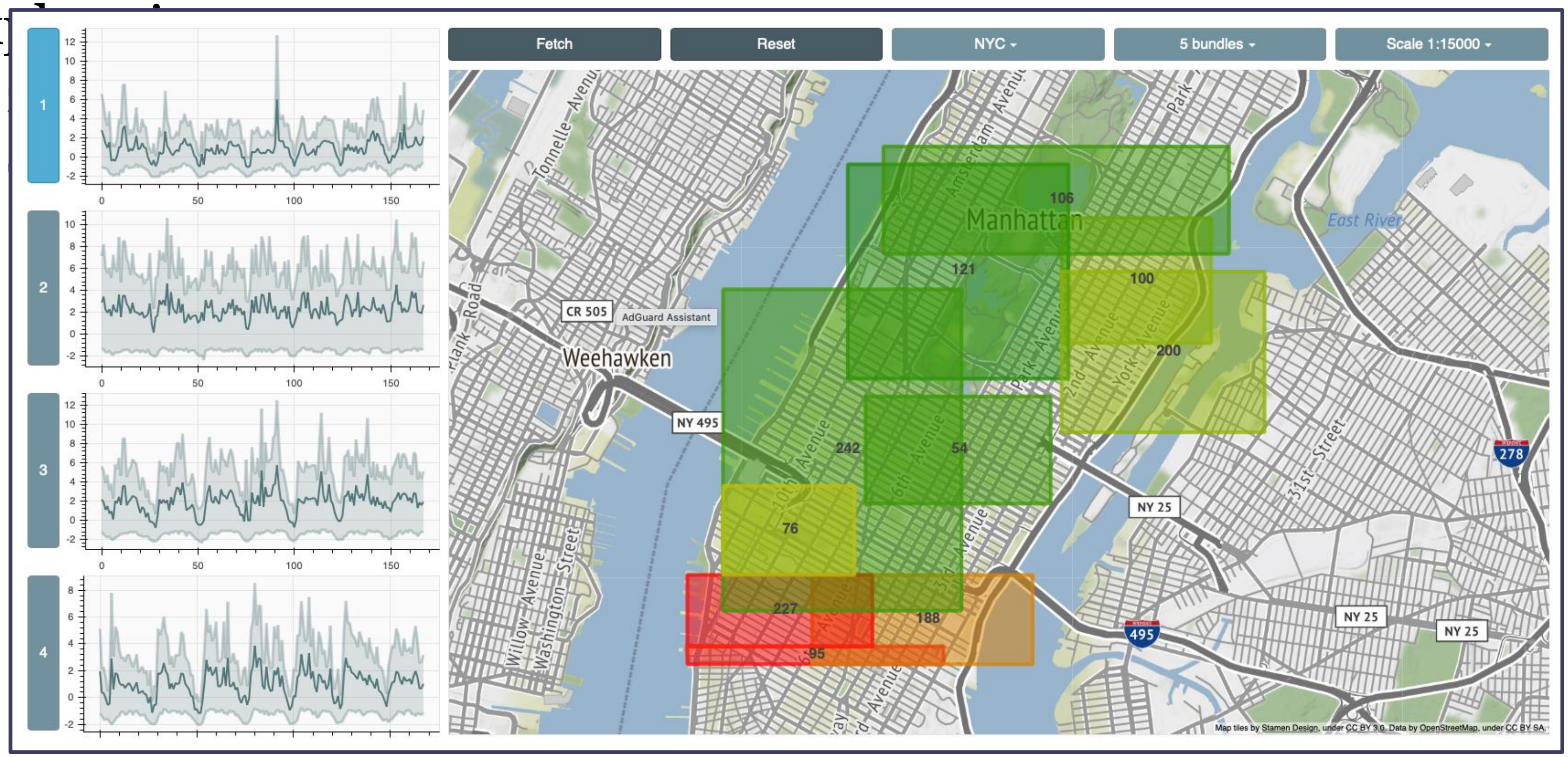
- interactive demo application for visual geolocated data series exploration
  - zoom-in/out, pan the map and receive summaries of geolocated data series and corresponding MBRs

Publications

Chatzigeorgakidis et al.
SIGSPATIAL/GIS'20

spaTScope

- interactive demo application for visual geolocated data series ex

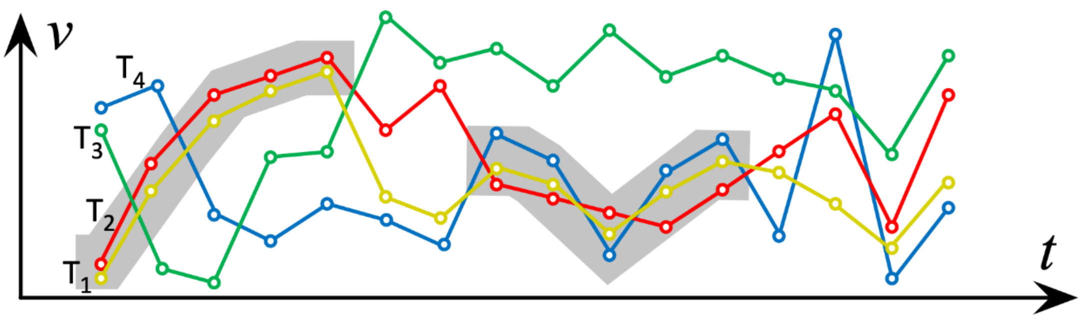


Publications

Chatzigeorgakidis et al. SSTD’19

Twin Subsequence Search

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



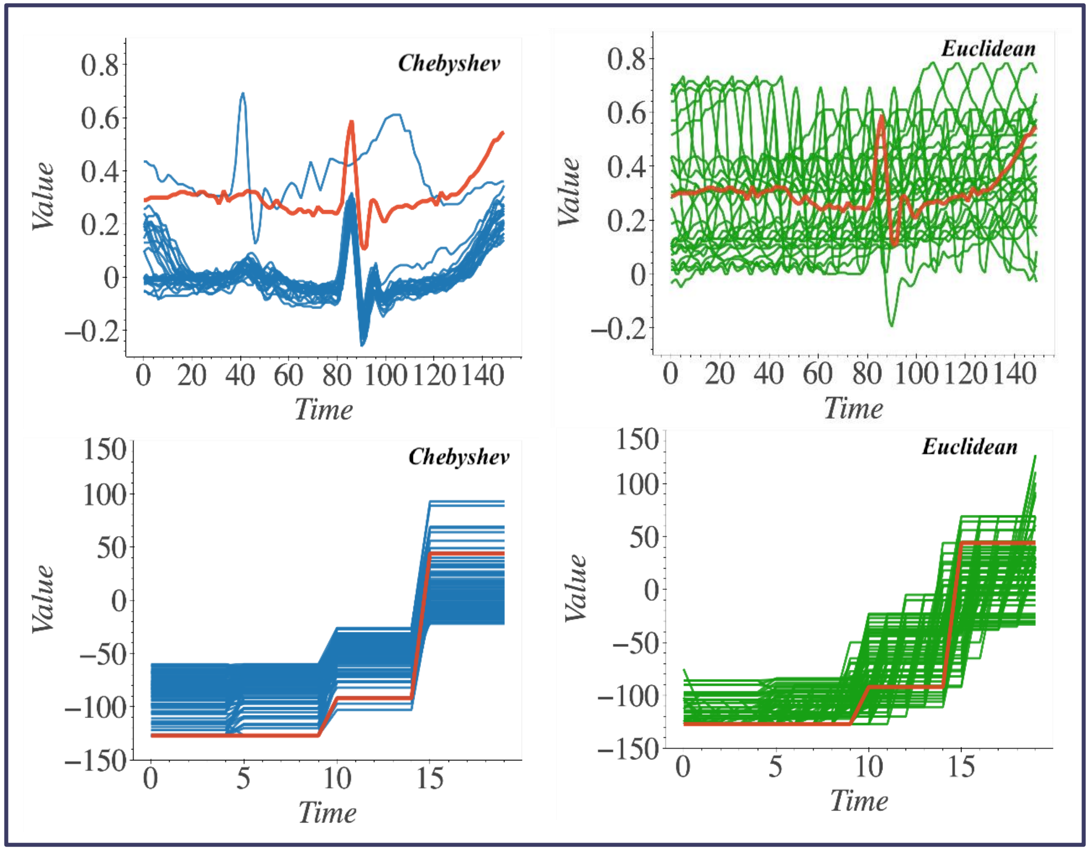
Publications

Chatzigeorgakidis et al. SSTD'19

Twin Subsequence Search

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

results that are Chebyshev-similar to the red queries



results that are Euclidean-similar to the red queries

Publications

Chatzigeorgakidis et al. SSTD'19

Twin Subsequence Search

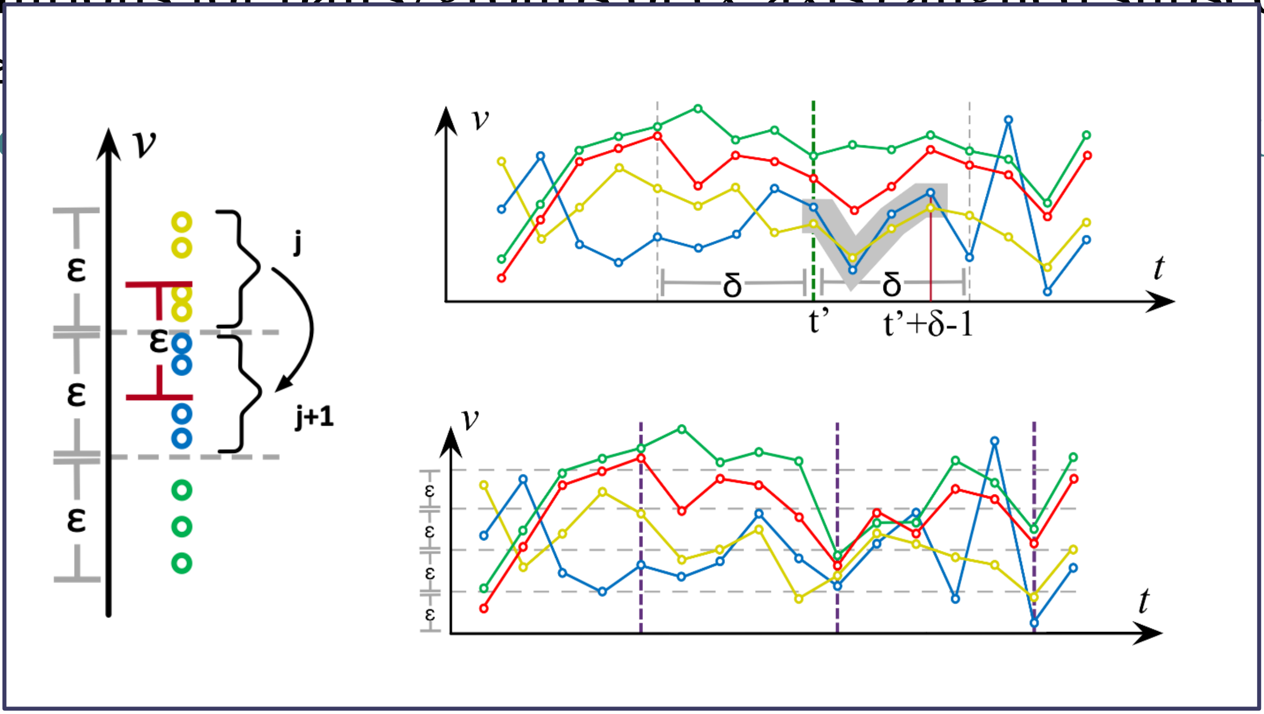
- discover subsequences, where distance between points is always < ε
- SL/CP: solutions for pairs/groups of (x-axis) aligned subsequences of length ≥δ, within large collections of (short) data series
  - prunes search space by discretizing values, and using checkpoints

Publications

Chatzigeorgakidis et al. SSTD'19

Twin Subsequence Search

- 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

Twin Subsequence Search

- discover subsequences, where distance between points is always < ε
- SL/CP: solutions for pairs/groups of (x-axis) aligned subsequences of length ≥δ, 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 geolocated
  - BTSR-Tree index on segmented data series, with bit-vectors that mark continuity of same series across segments

Publications

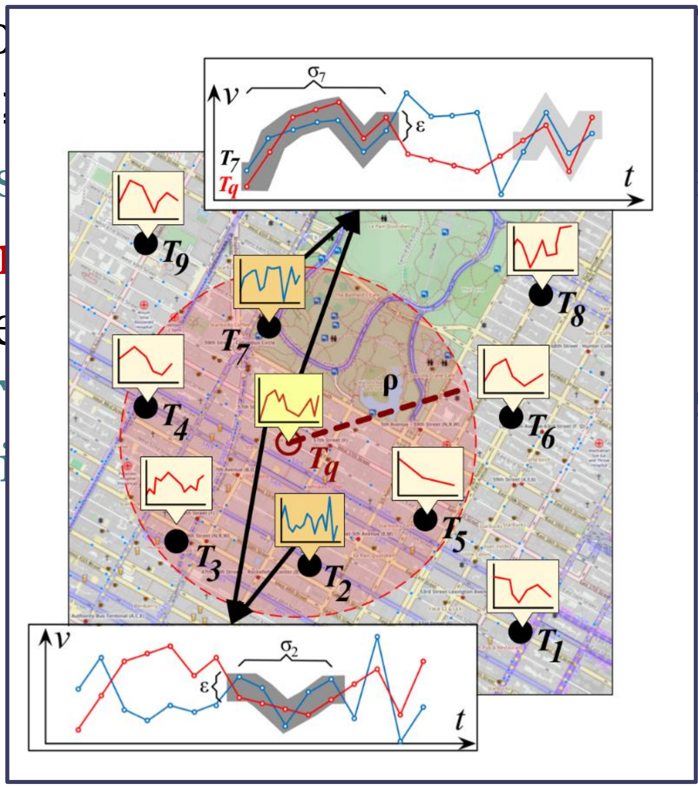
Chatzigeorgakidis et al. SSTD'19

Chatzigeorgakidis et al. SIGSPATIAL/GIS'19

Twin Subsequence Search

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

- SL/CP: so (x-axis) aligned subsequences of length of (short) data series
  - prunes s alues, and using checkpoints
- SBTSR-T gned subsequences within large colle s, which are geo-located
  - BTSR-T series, with bit-vectors that mark continu ents



Publications

Chatzigeorgakidis et al. EDBT'21

Twin Subsequence Search

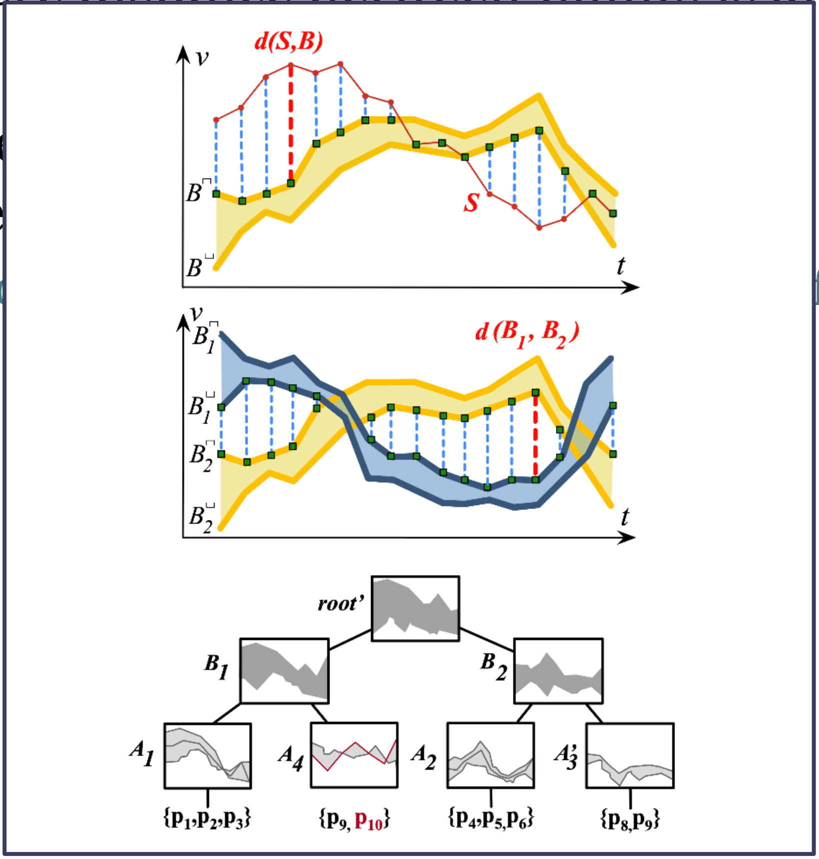
- discover subsequences, where distance between points is always < ε
- TS-Index: solution for subsequences of a long data series *T* that are similar to a (short) query sequence of length *l*
  - k-ary balanced index, built on per-point min/max envelopes of all *l*-length subsequences of *T*

Publications

Chatzigeorgakidis et al. EDBT'21

Twin Subsequence Search

- discover subsequences, where distance between points is always < ε
- TS-Index: solution for subse… that are similar to a (short) query se…
  - k-ary balanced index, built … f all *l*-length subsequences of *T*



Publications

Chatzigeorgakidis et al. EDBT'21

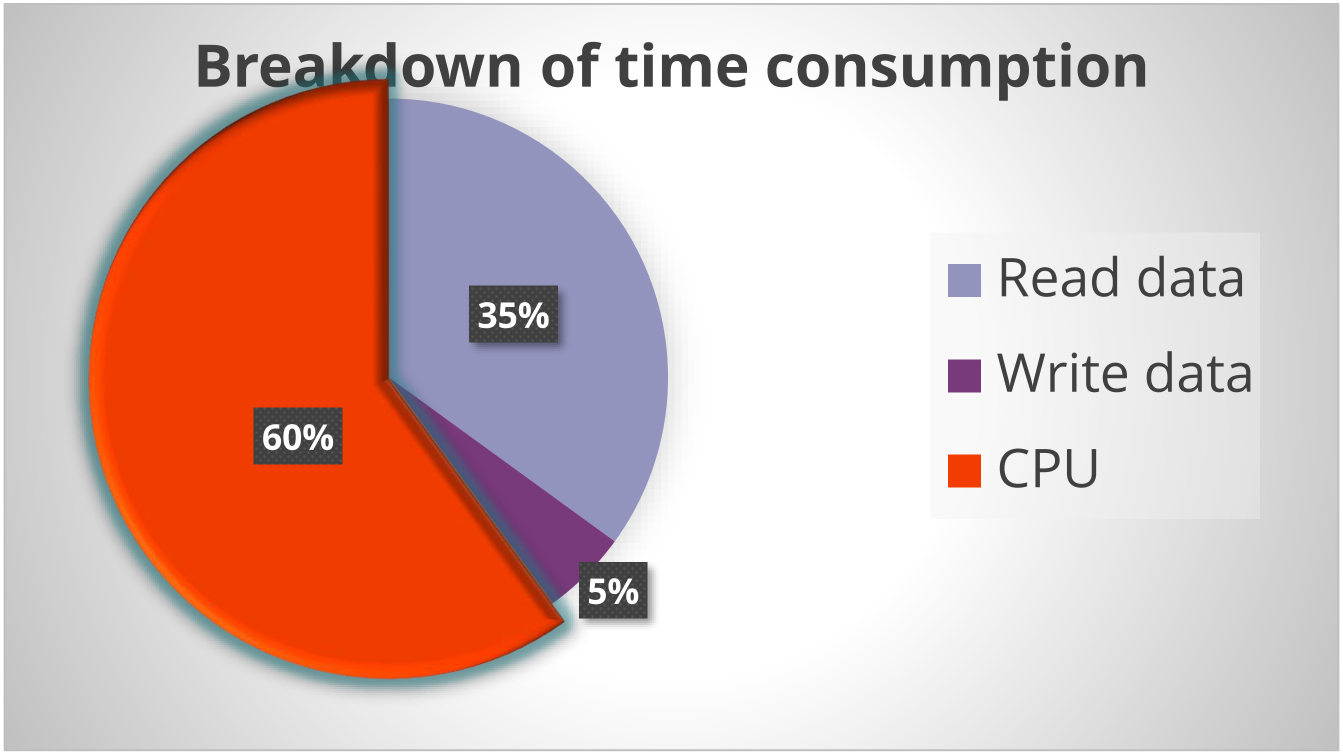
Chatzigeorgakidis et al. TKDE'22

Twin Subsequence Search

- discover subsequences, where distance between points is always < ε
- 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*
- TS-Index OPT: memory footprint and bulk-loading optimizations for TS-Index
  - build index bottom-up after sorting and grouping the subsequences using a z-order space filling curve

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

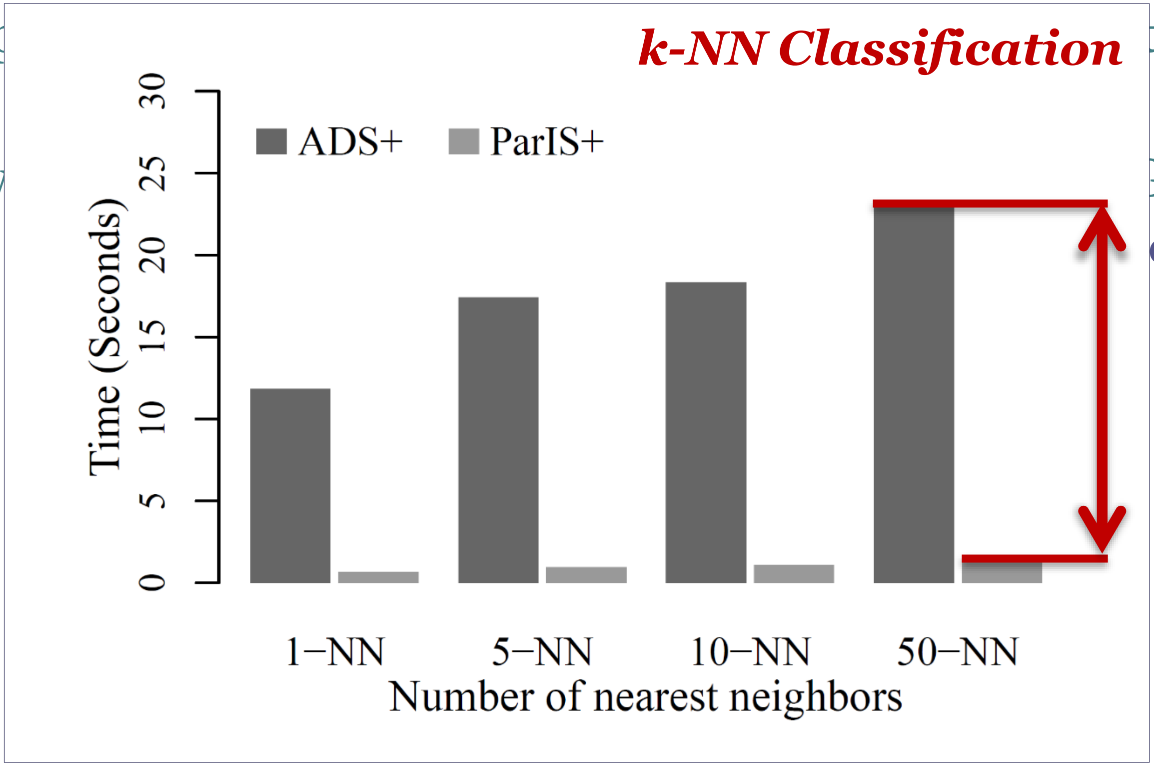
Publications

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



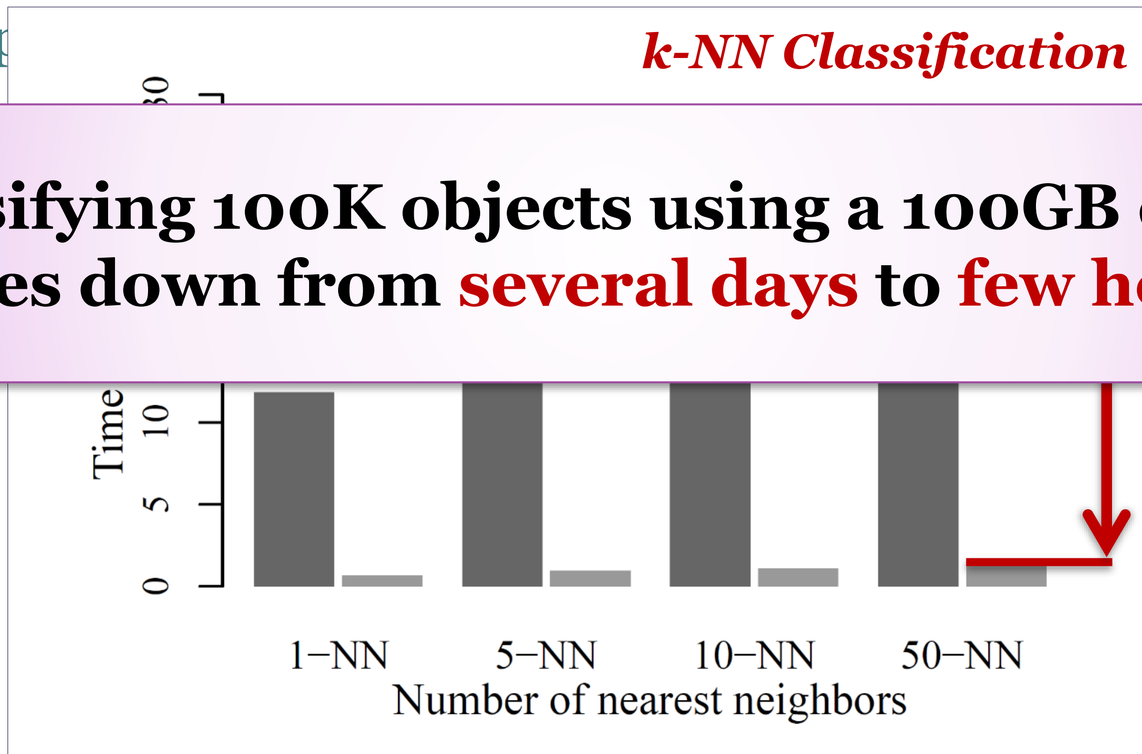
Publications

Peng-BigData'18

Peng-TKDE'20

ParIS+
Parallel Indexing of Sequences

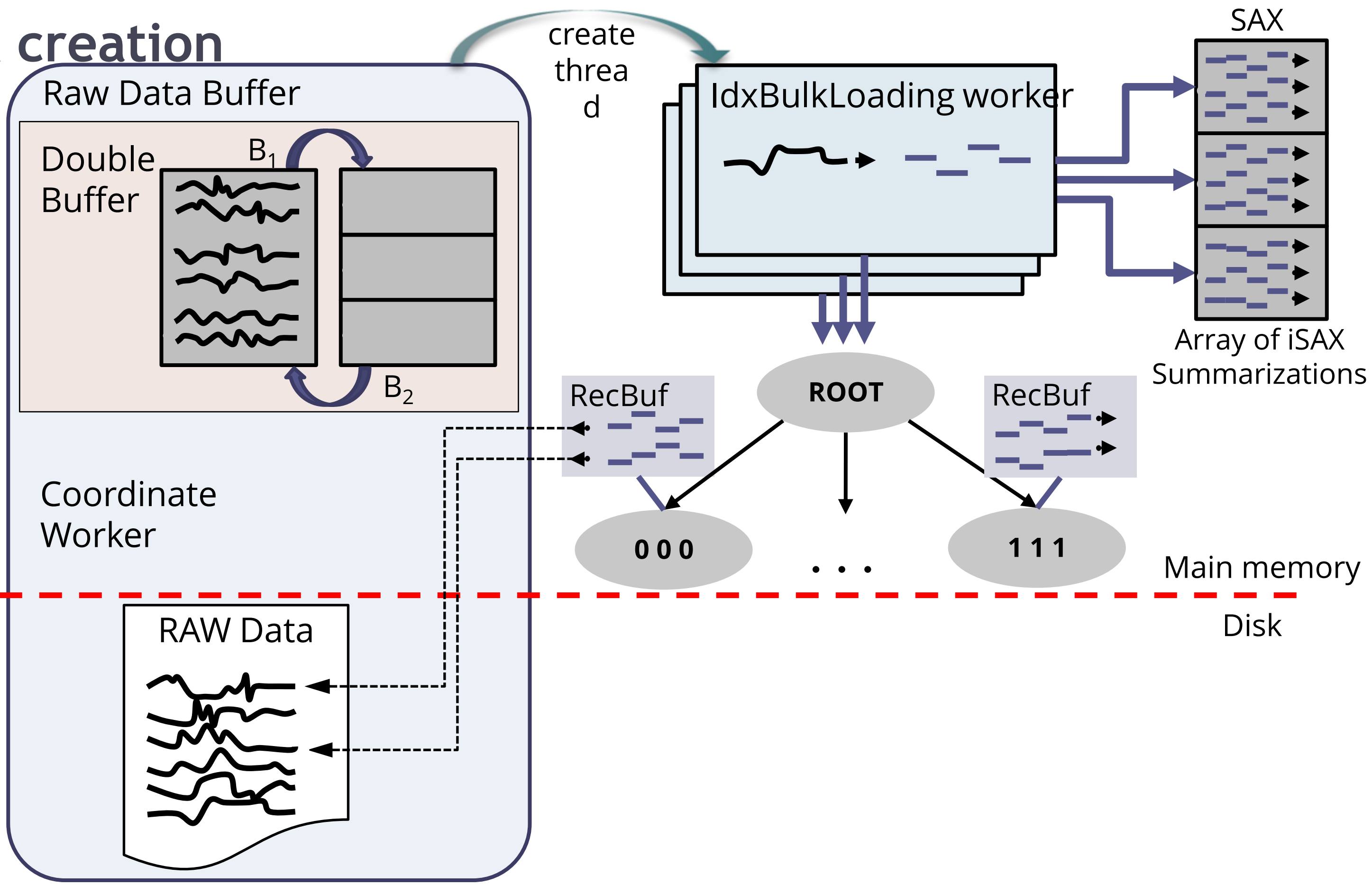
- solution for SIMD, multi-core, multi-socket architectures
  - comp... tion



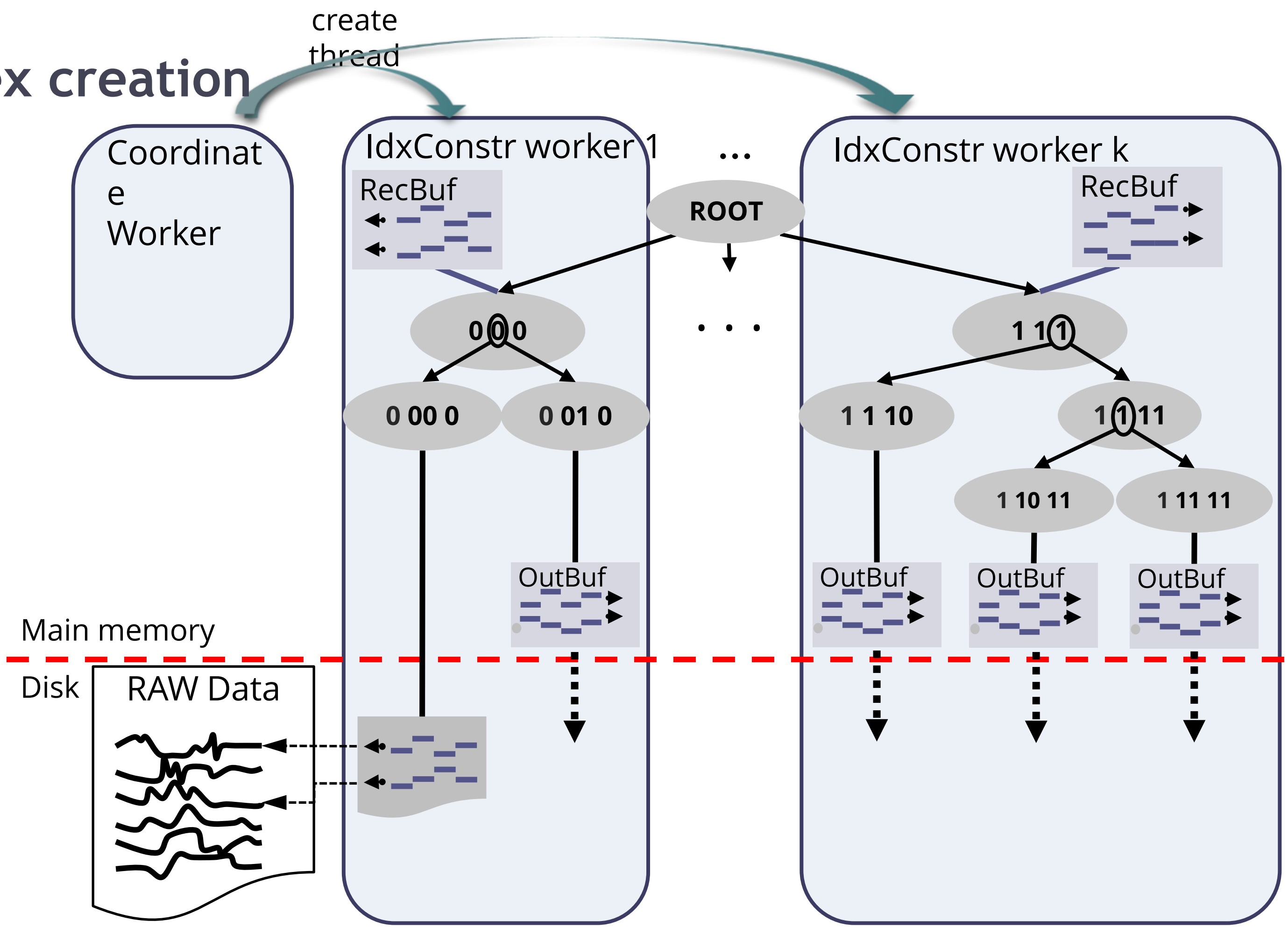
18x faster

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

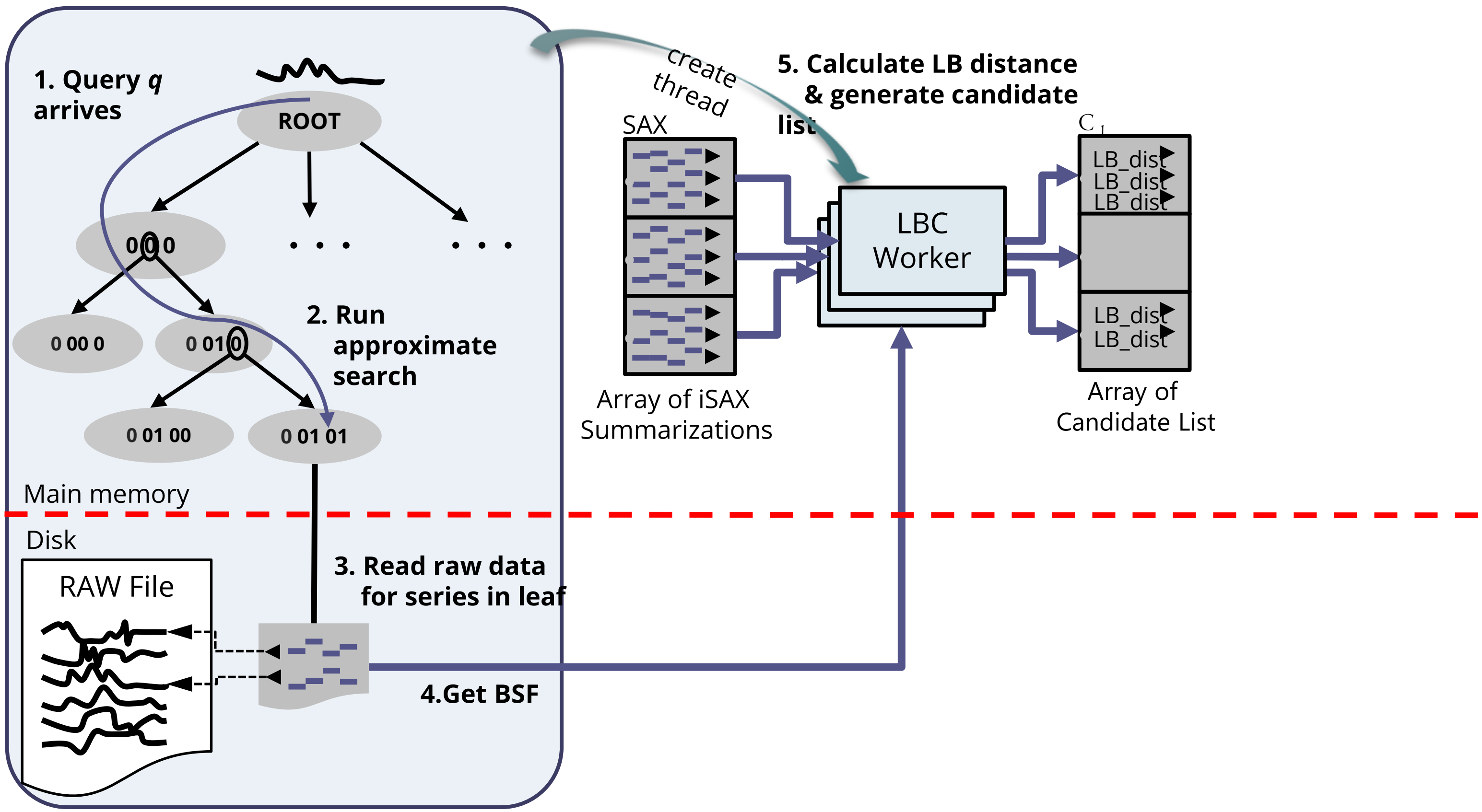
Index creation



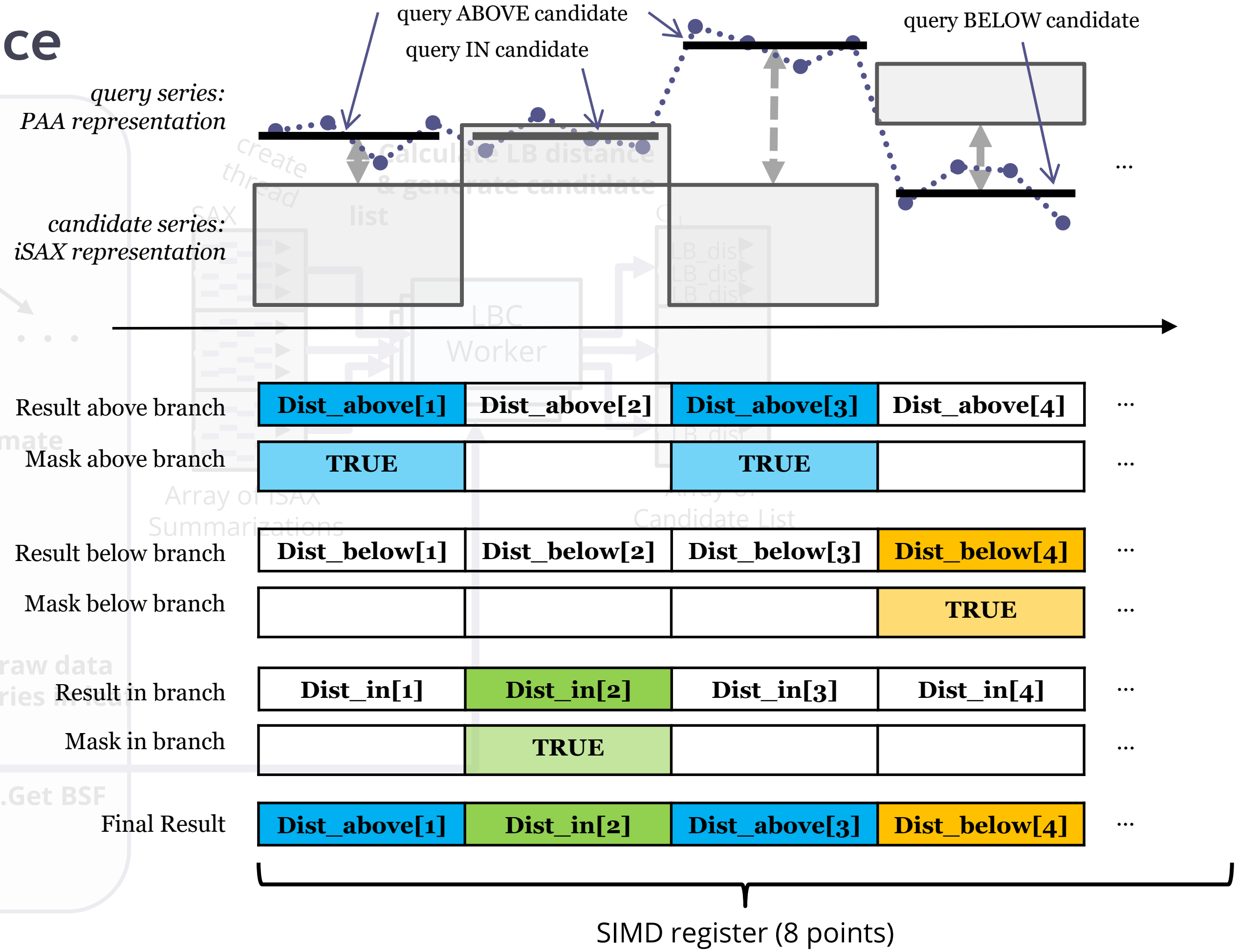
Index creation



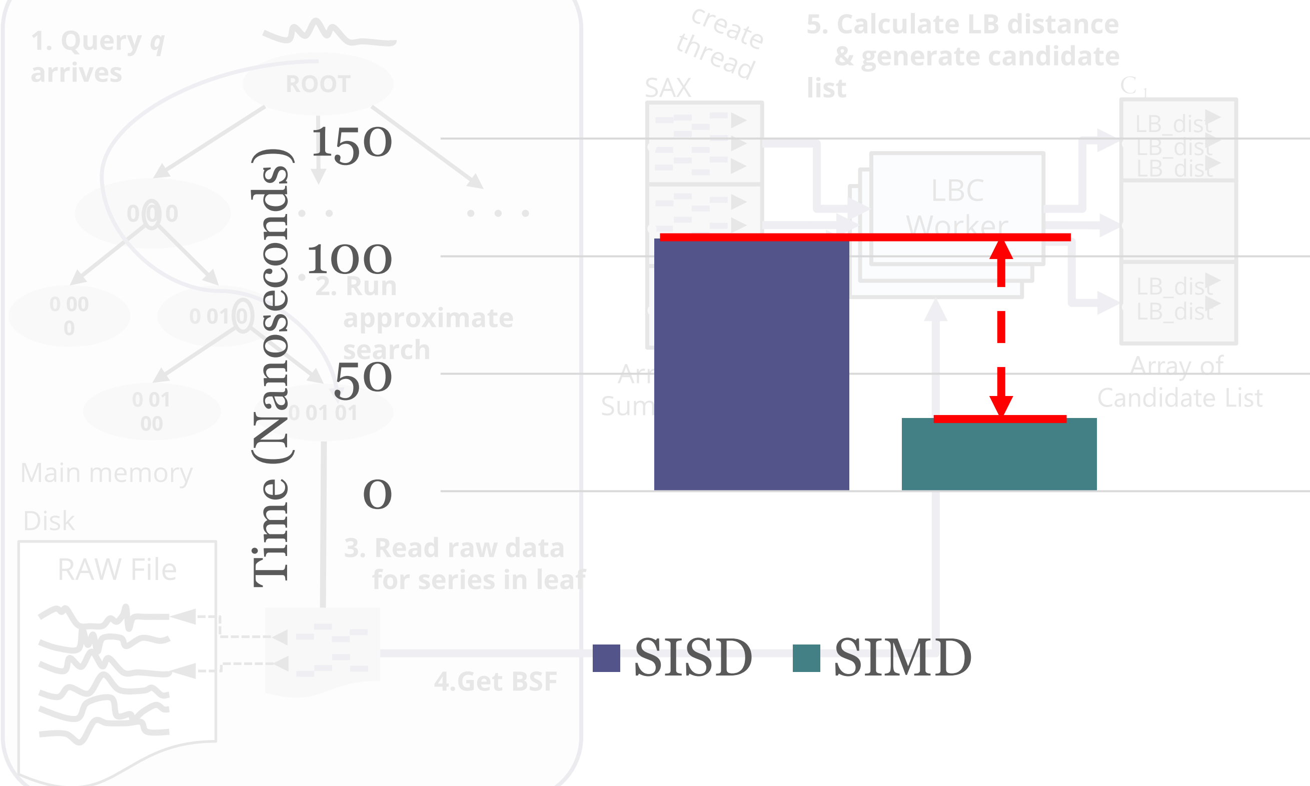
ParIS+ exact query answering



Lower-Bound Distance Calculation in SIMD

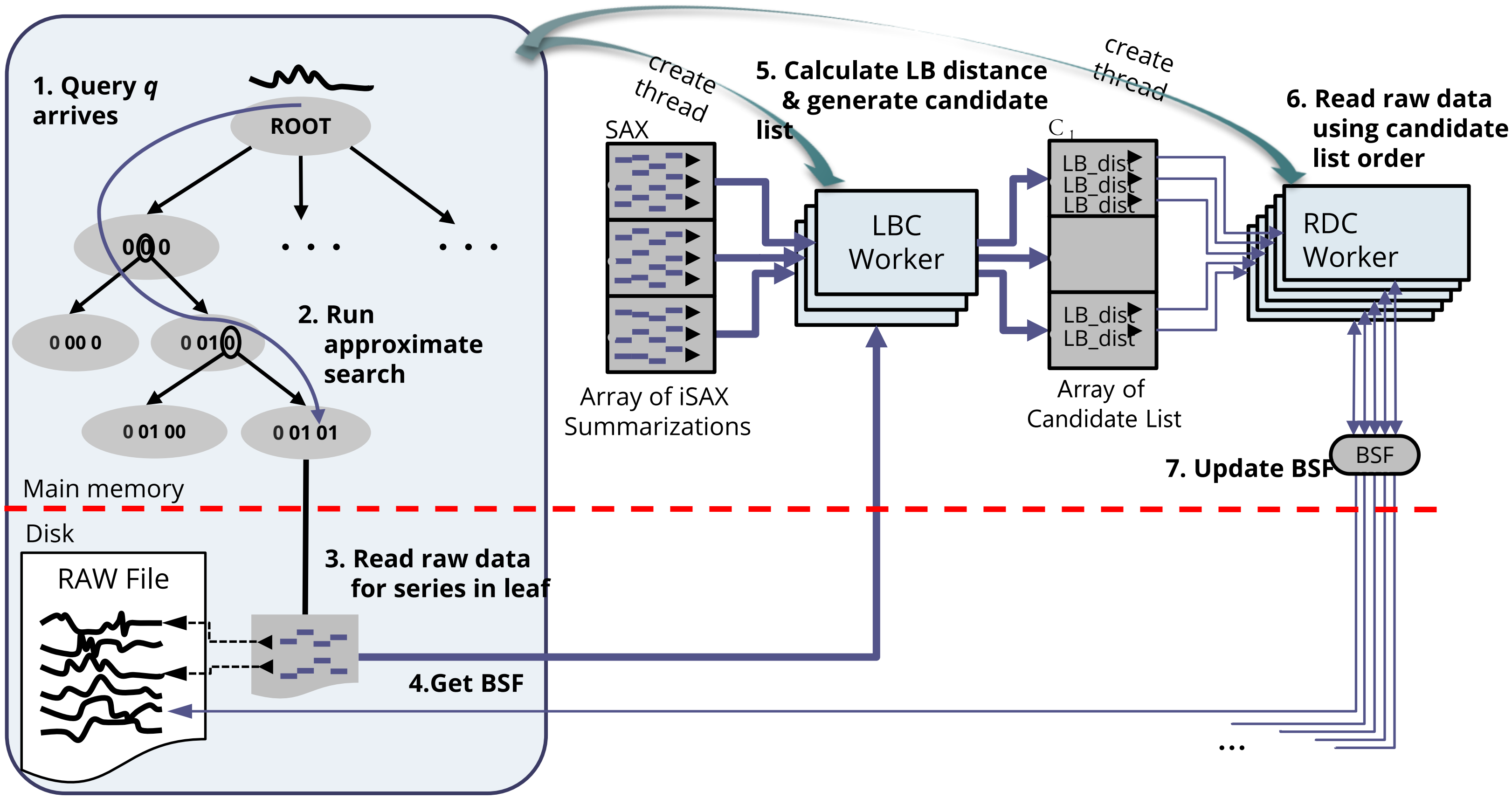


Lower-Bound Distance Calculation in SIMD



SIMD lower bounds are 3.4x faster

ParIS+ exact query answering



Publications

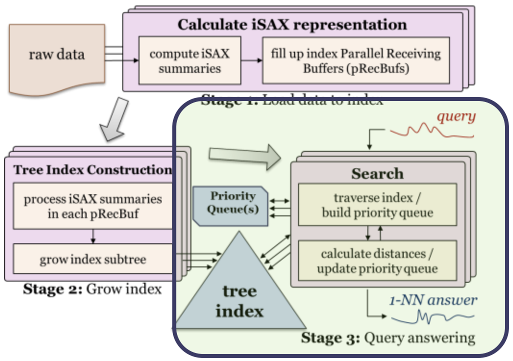
Peng-ICDE'20

Peng-VLDBJ'21

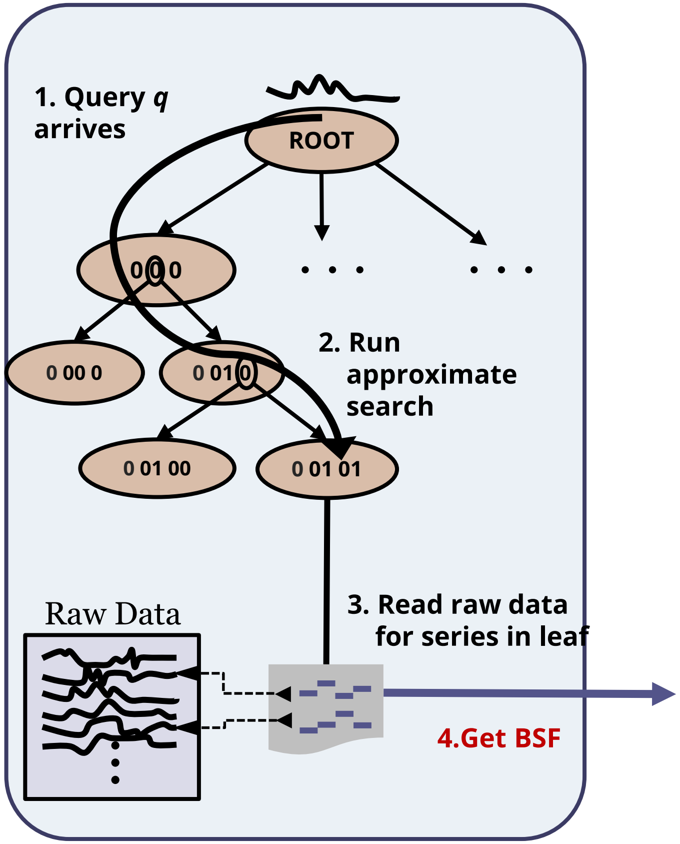
MESSI
In-Memory Data Series Index

- 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

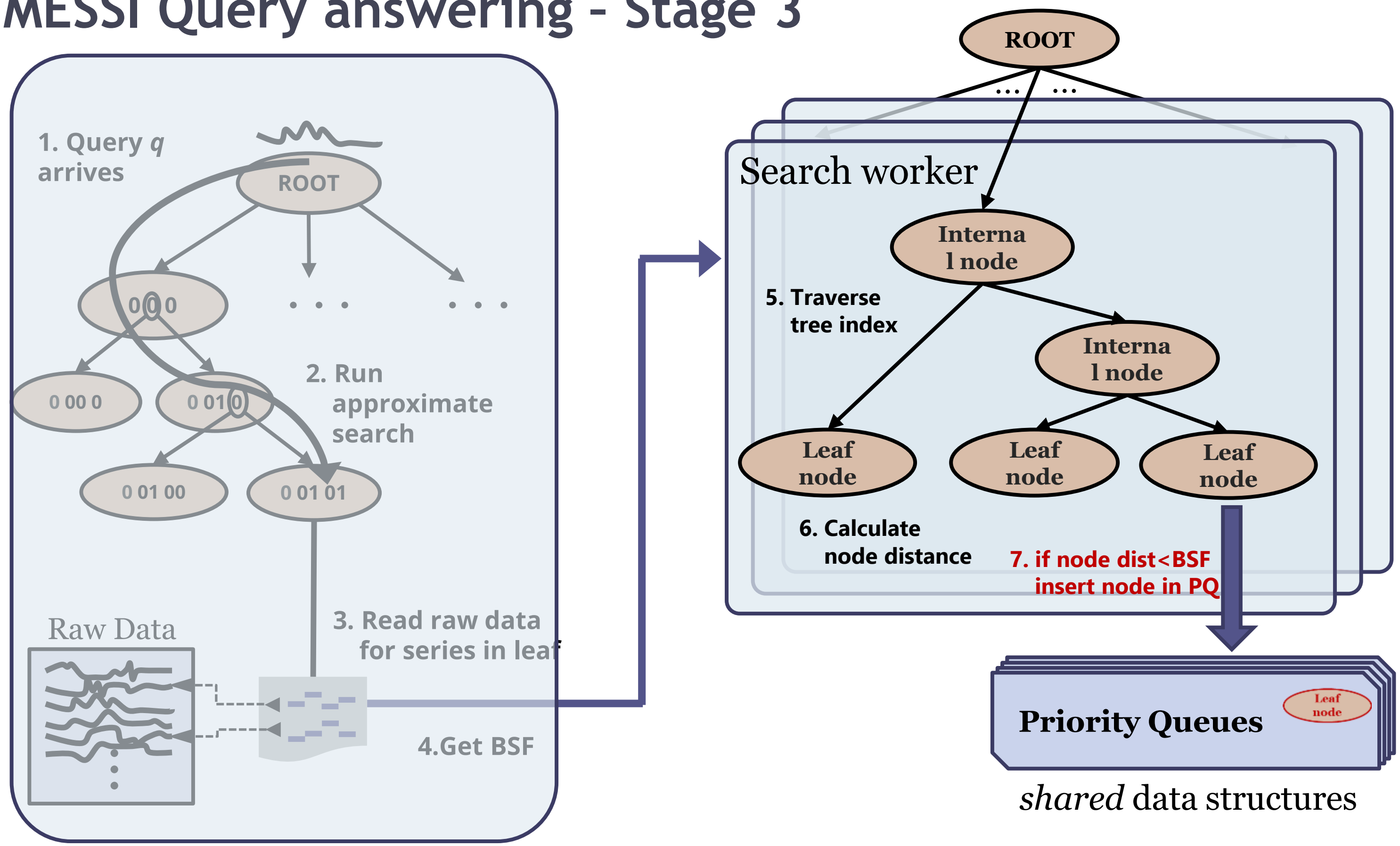
MESSI Query answering - Stage 3



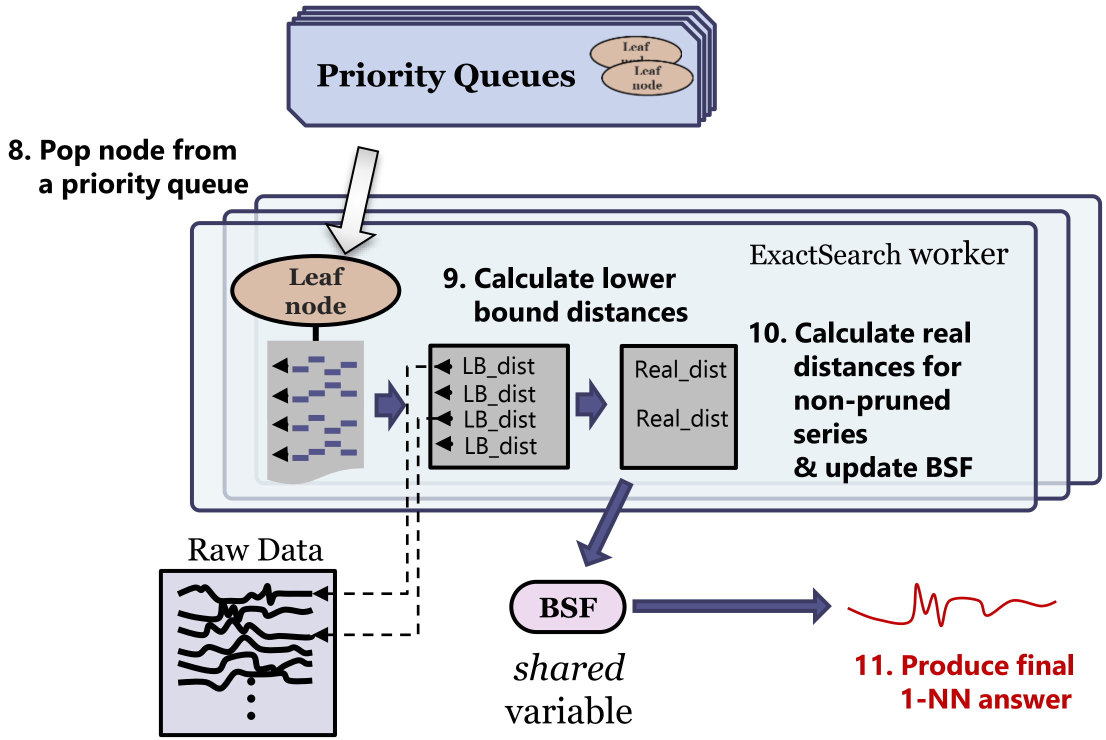
MESSI Query answering - Stage 3



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

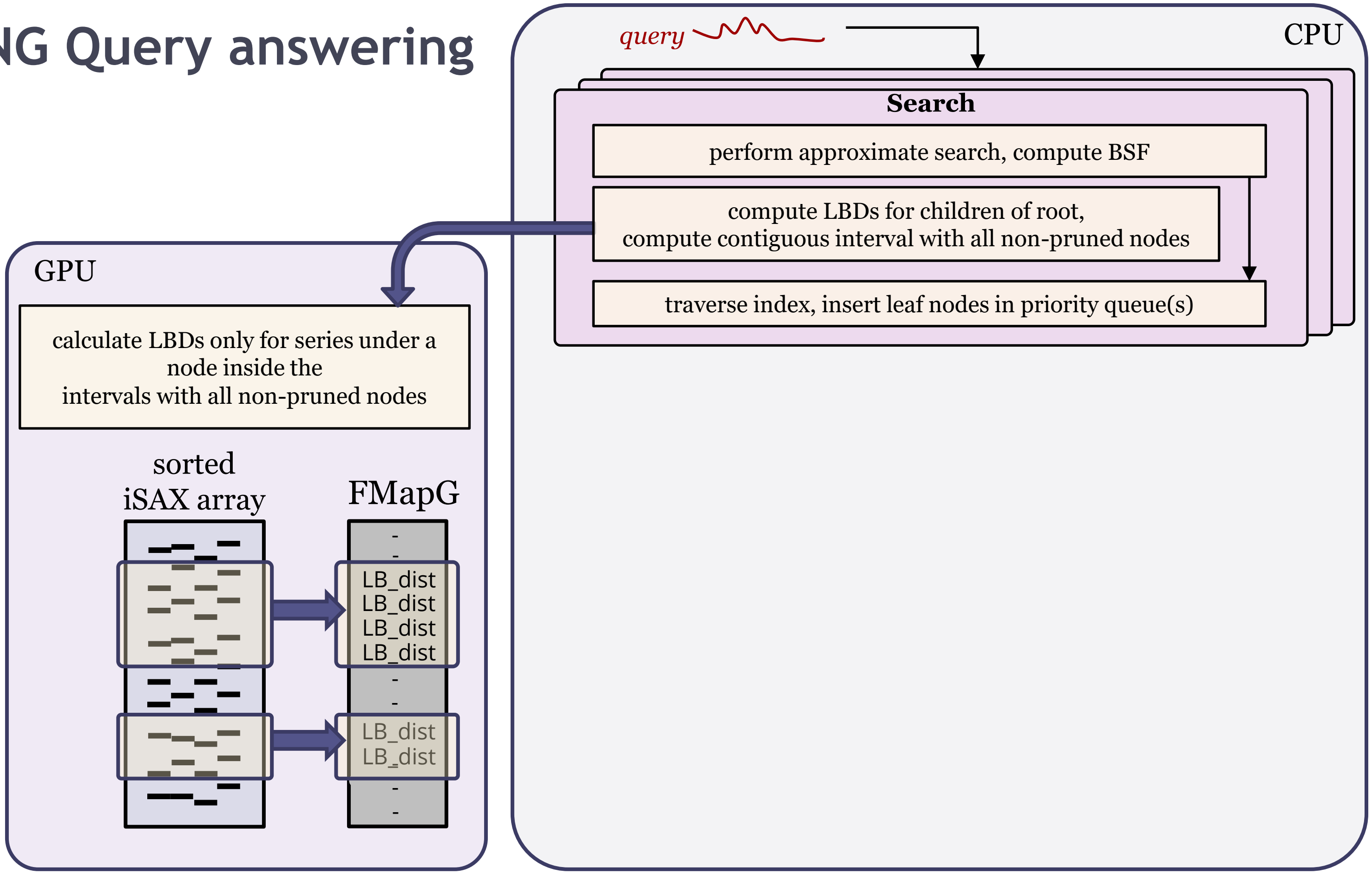
GPUs for Data Series Similarity Search

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

- 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

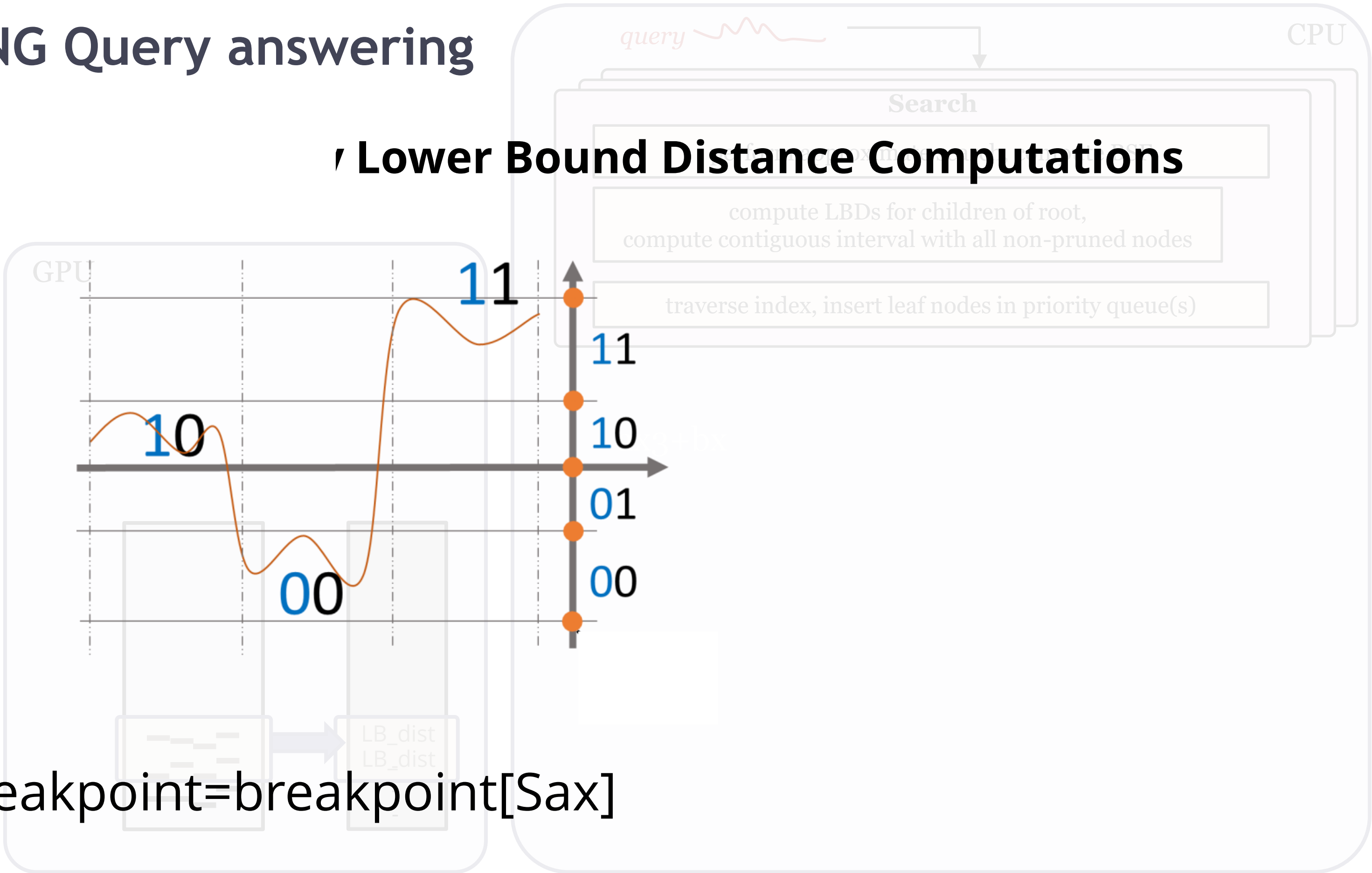
  - trade-offs will change as GPU and interconnects technology advances

SING Query answering



SING Query answering

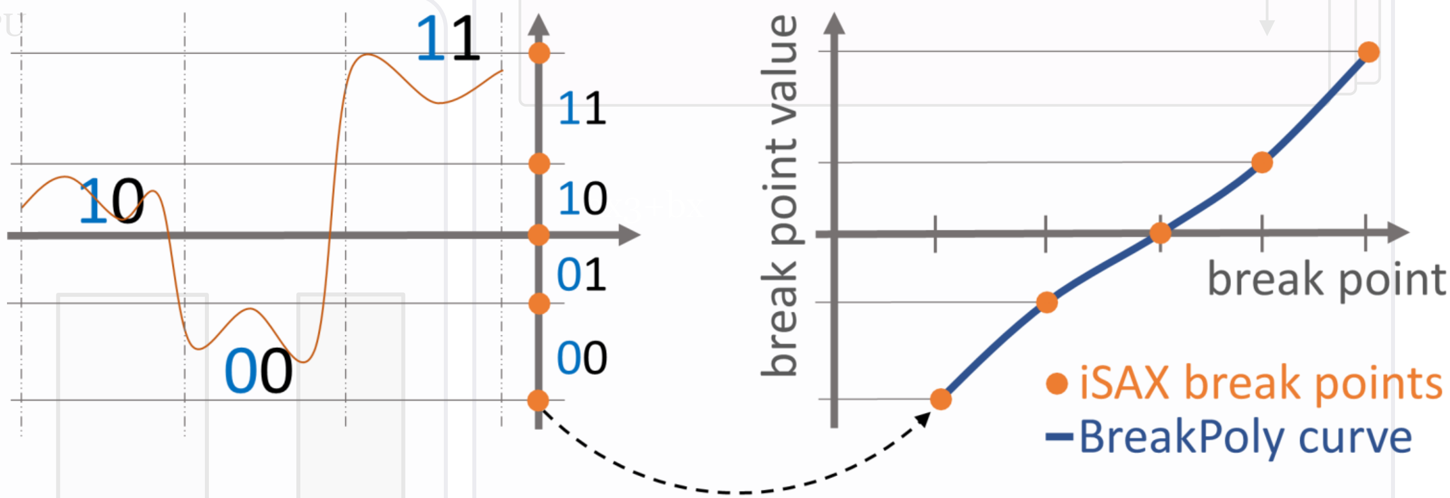
Lower Bound Distance Computations



Breakpoint=breakpoint[Sax]

SING Query answering

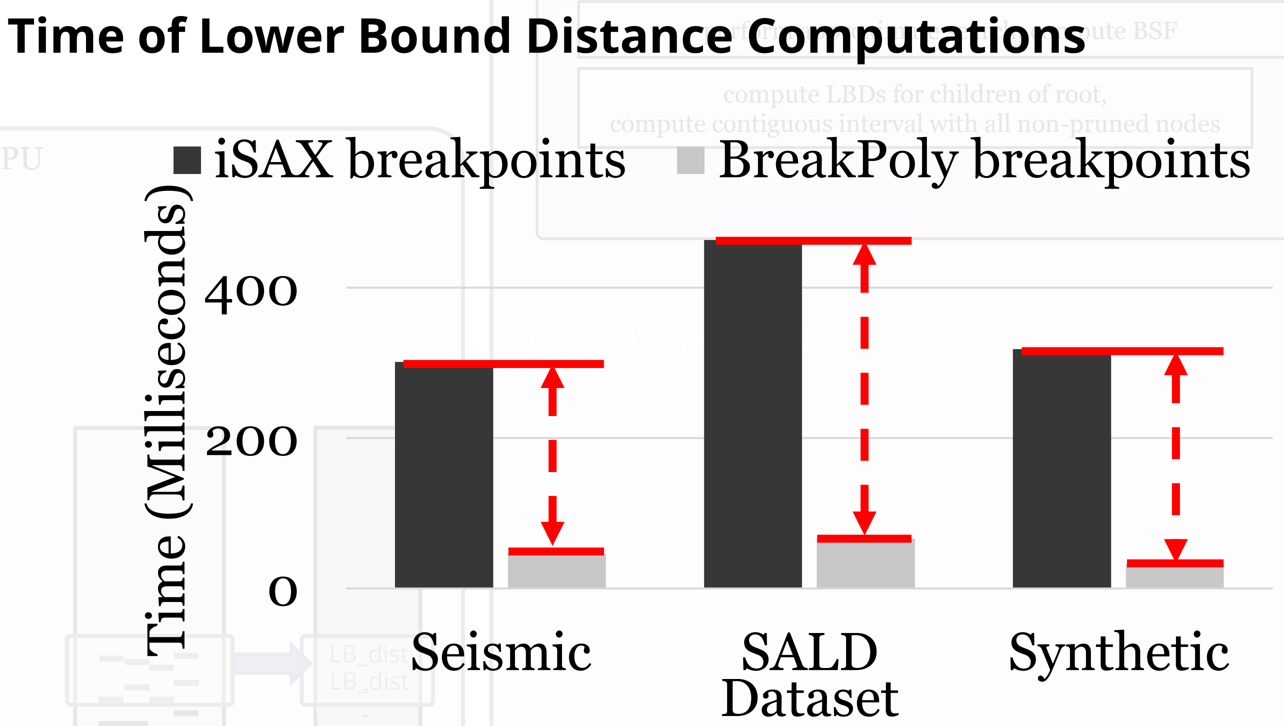
GPU-friendly Lower Bound Distance Computations



Breakpoint=breakpoint[Sax]

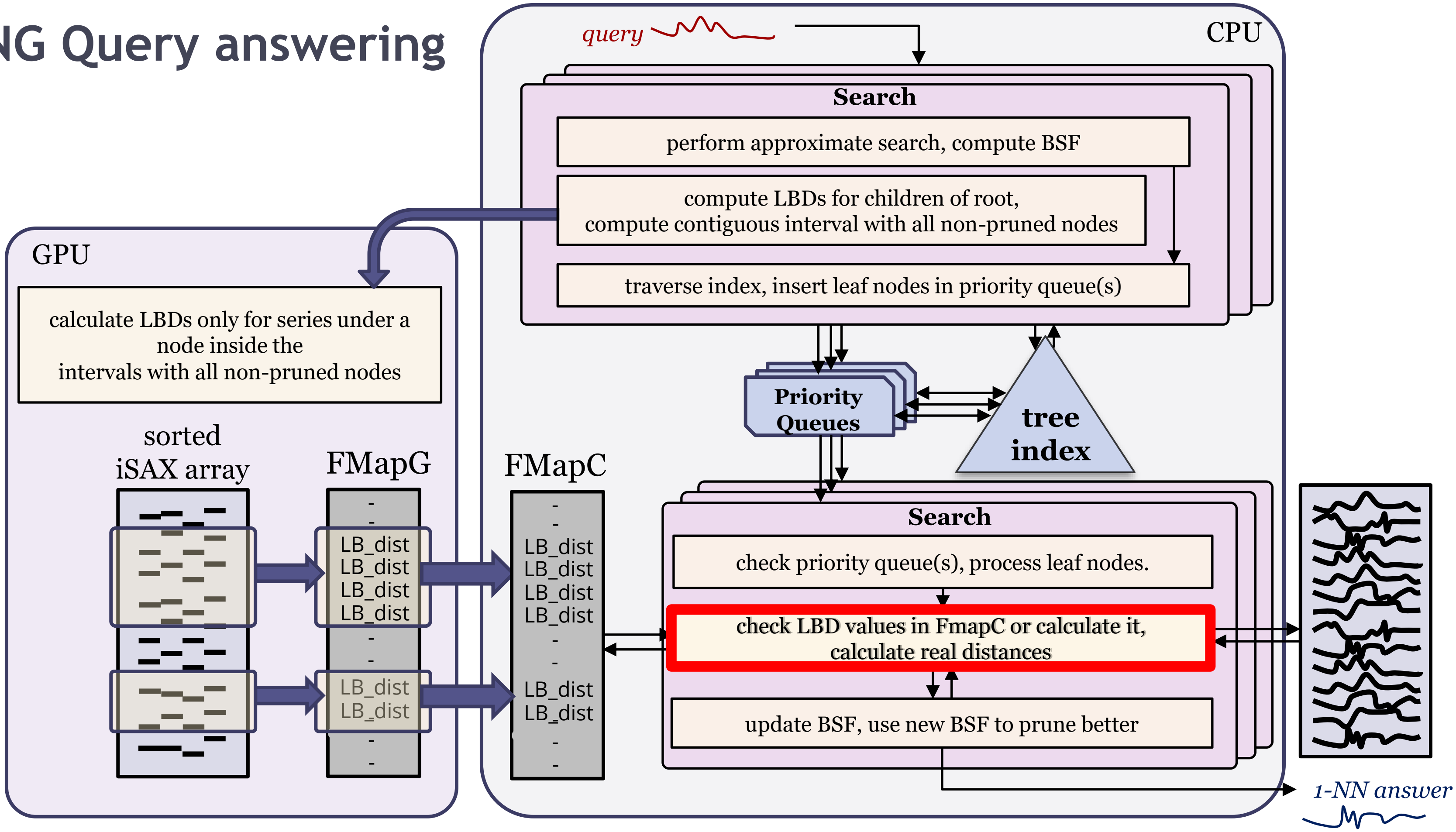
BreakPoly(Sax)=a*(Sax)^3+b*(Sax)

SING Query answering



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

SING Query answering



DPiSAX
Distributed Partitioned iSAX

Publications

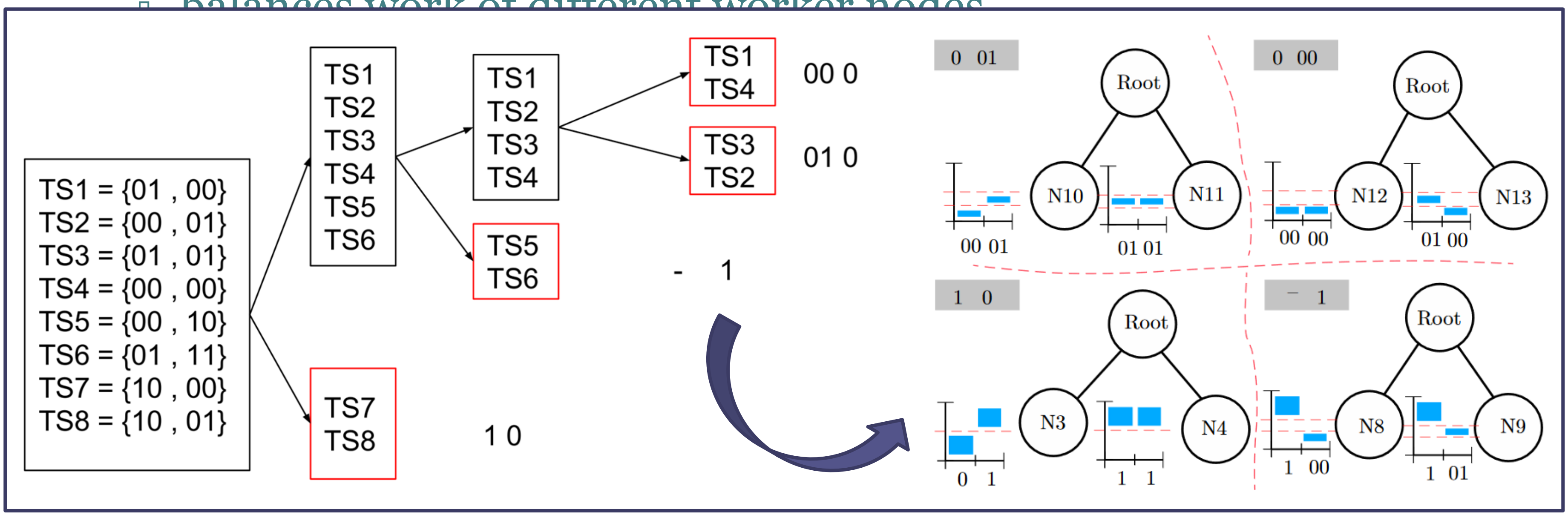
- Yagoubi-ICDM'17
- Yagoubi-TKDE'18
- Lavchenko-KAIS'20

- 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

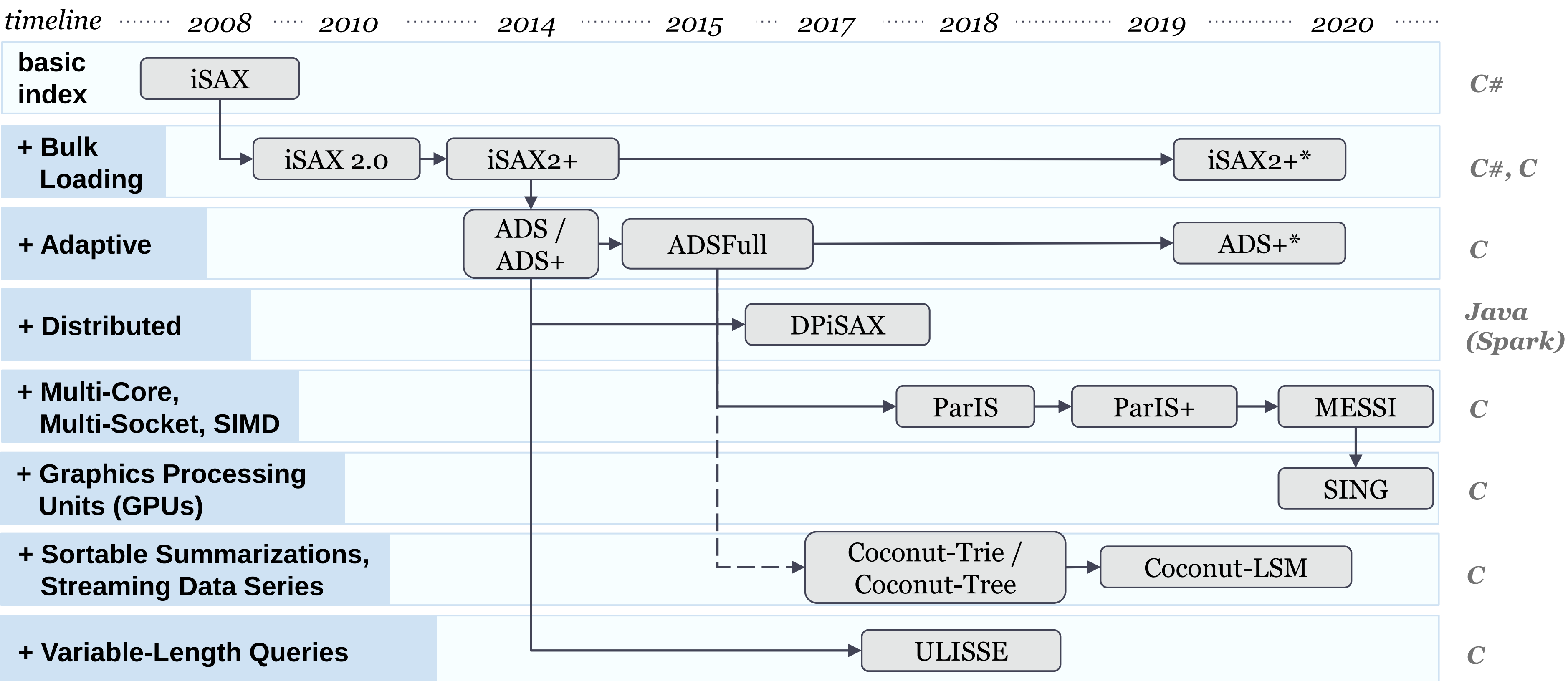
- Yagoubi-ICDM'17
- Yagoubi-TKDE'18
- Lavchenko-KAIS'20

- 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)

iSAX Index Family Lineage Tree

Publications

Palpanas-ISIP'19

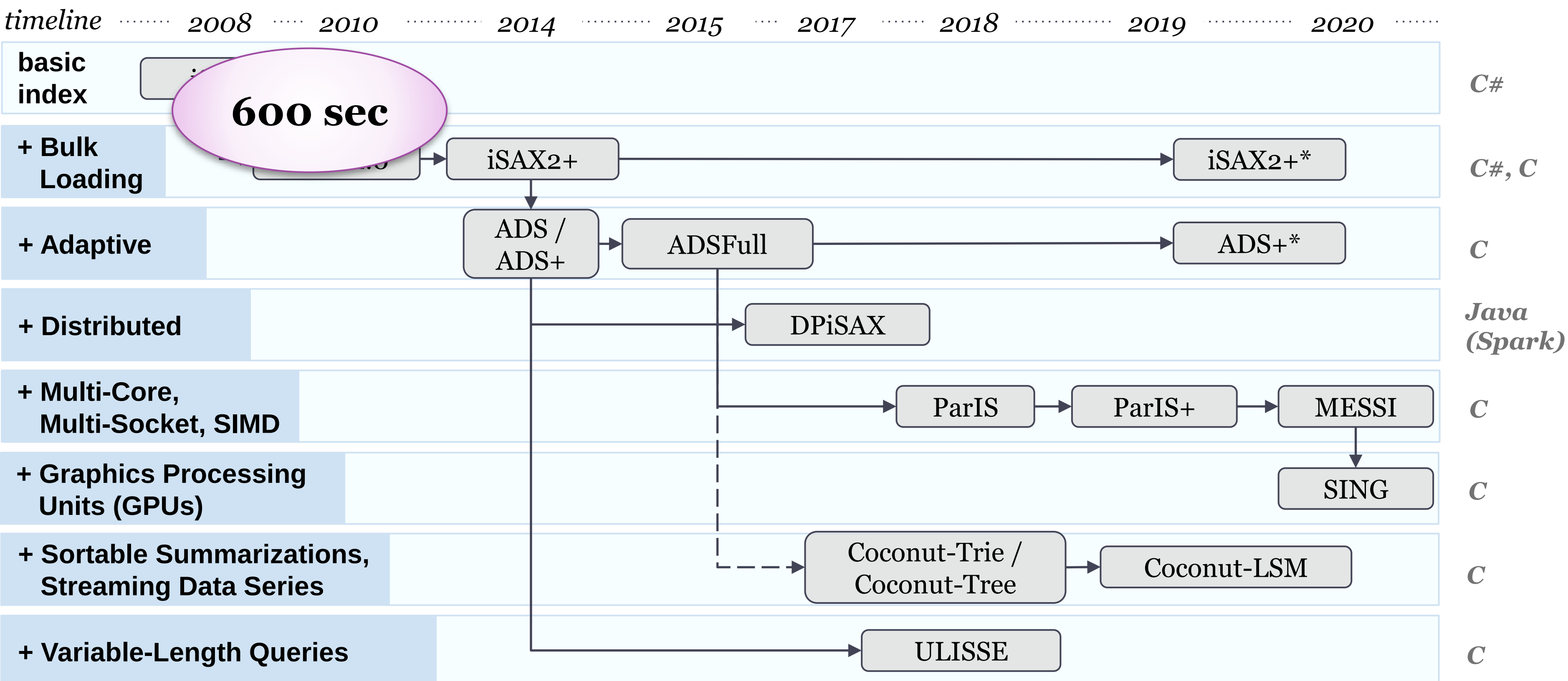


Timeline depicted on top; implementation languages marked on the right. Solid arrows denote inheritance of index design; dashed arrows denote inheritance of some of the design features; two new versions of iSAX2+/ADS+ marked with asterisk support approximate similarity search with deterministic and probabilistic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

Publications

Palpanas-ISIP'19

iSAX Index Family Lineage Tree



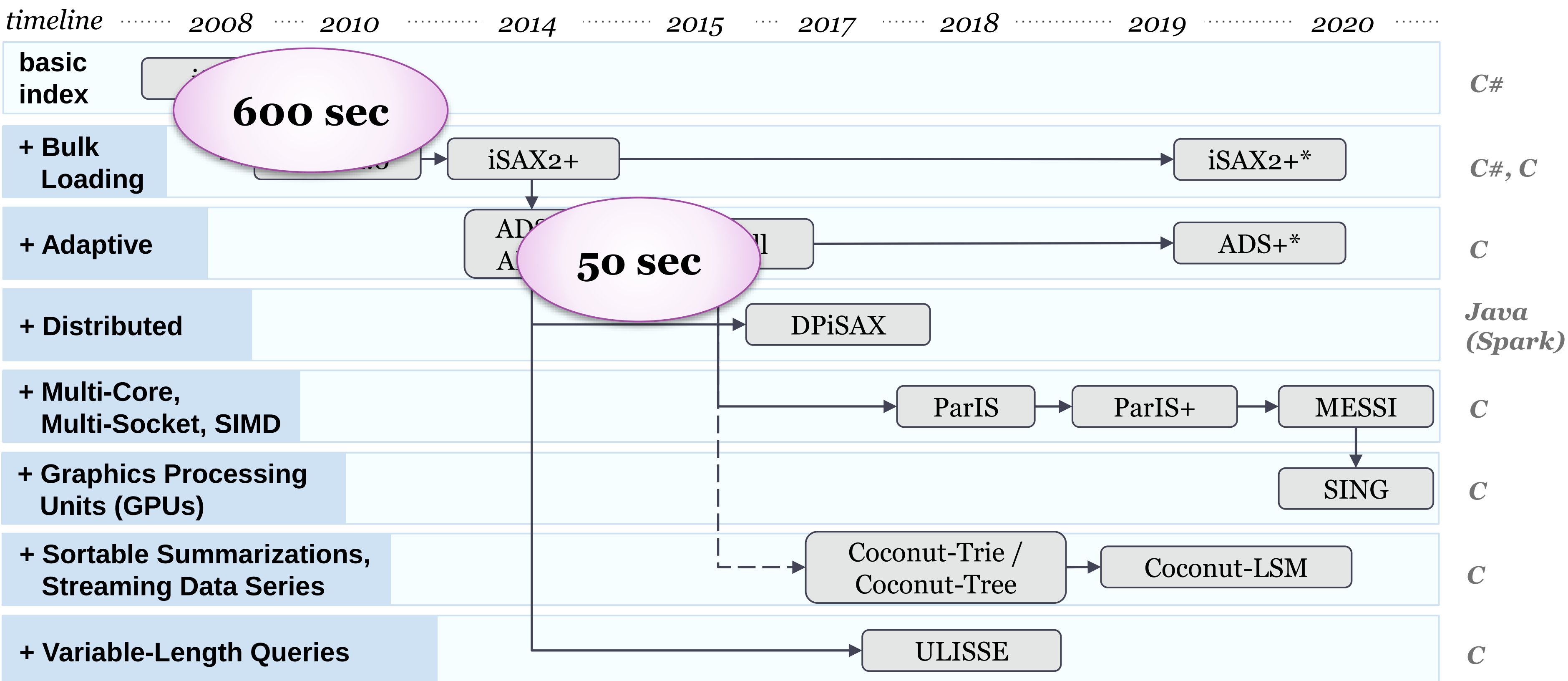
execution time for **1 similarity search query on a 100GB dataset *on disk***

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.

iSAX Index Family Lineage Tree

Publications

Palpanas-ISIP‘19



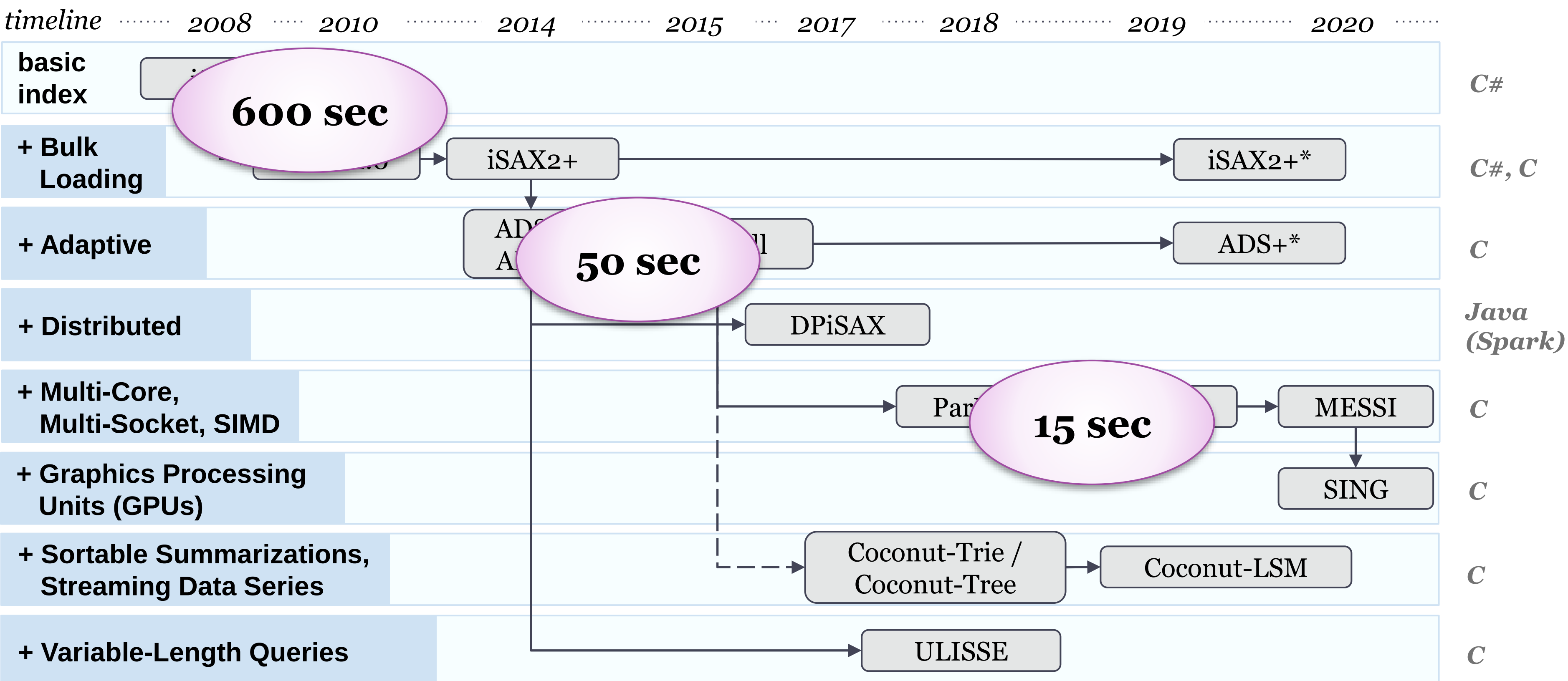
execution time for **1 similarity search query on a 100GB dataset *on disk***

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.

Publications

Palpanas-ISIP'19

iSAX Index Family Lineage Tree



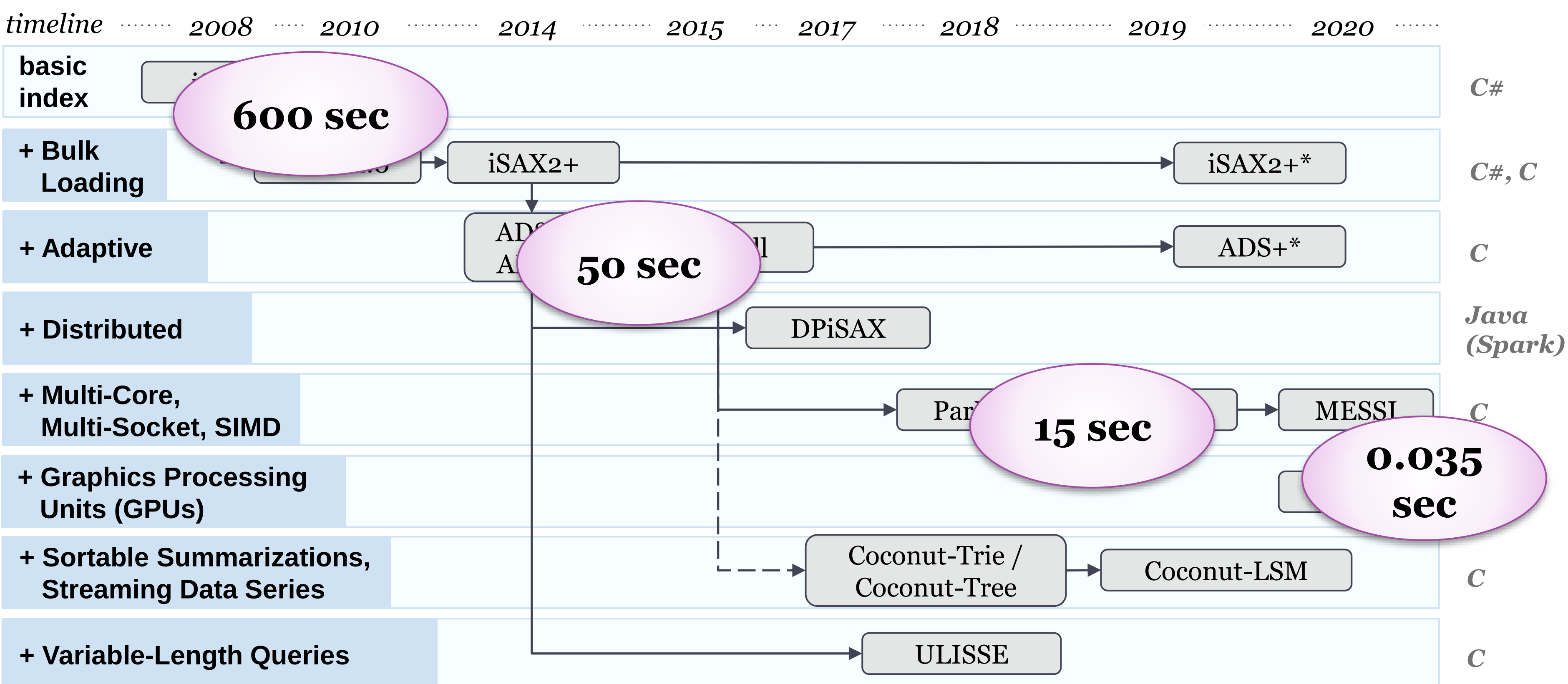
execution time for **1 similarity search query on a 100GB dataset *on disk***

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.

iSAX Index Family Lineage Tree

Publications

Palpanas-ISIP'19



execution time for **1 similarity search query on a 100GB dataset *in memory***

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.

Publications

Echihabi-PVLDB'22

Hercules
Parallel Indexing of Sequences

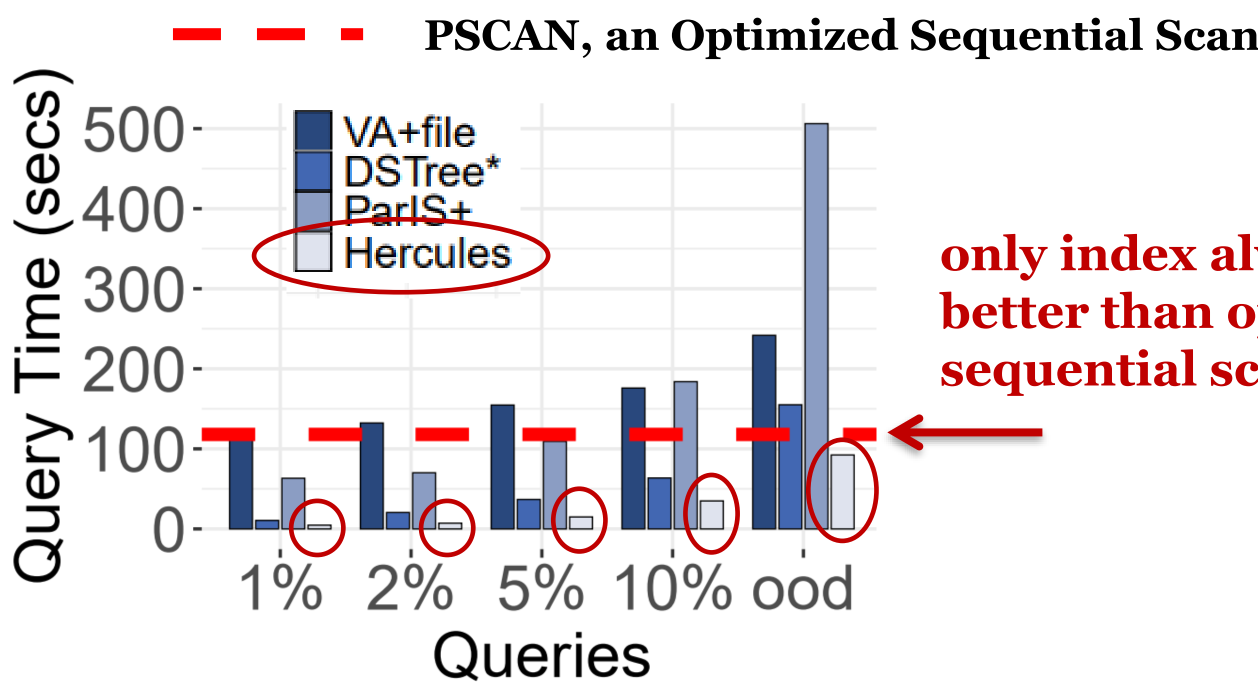
- Disk-based solution for SIMD, multi-core, multi-socket architectures
  - Exploits the benefits of two different summarization techniques (iSAX and EAPCA), and novel indexing and query answering algorithms
  - Leads to better query answering performance than all recent state-of-the-art approaches across all popular query workloads
    - only index that outperforms optimized scan on all scenarios (including hard query workloads on disk-based datasets)
  - Performs up to one order of magnitude faster than the best competitor (which is not always the same)

Publications

Echihabi-PVLDB'22

Hercules
Parallel Indexing of Sequences

- Disk-based solution for SIMD, multi-core, multi-socket architectures



only index always better than optimized sequential scan

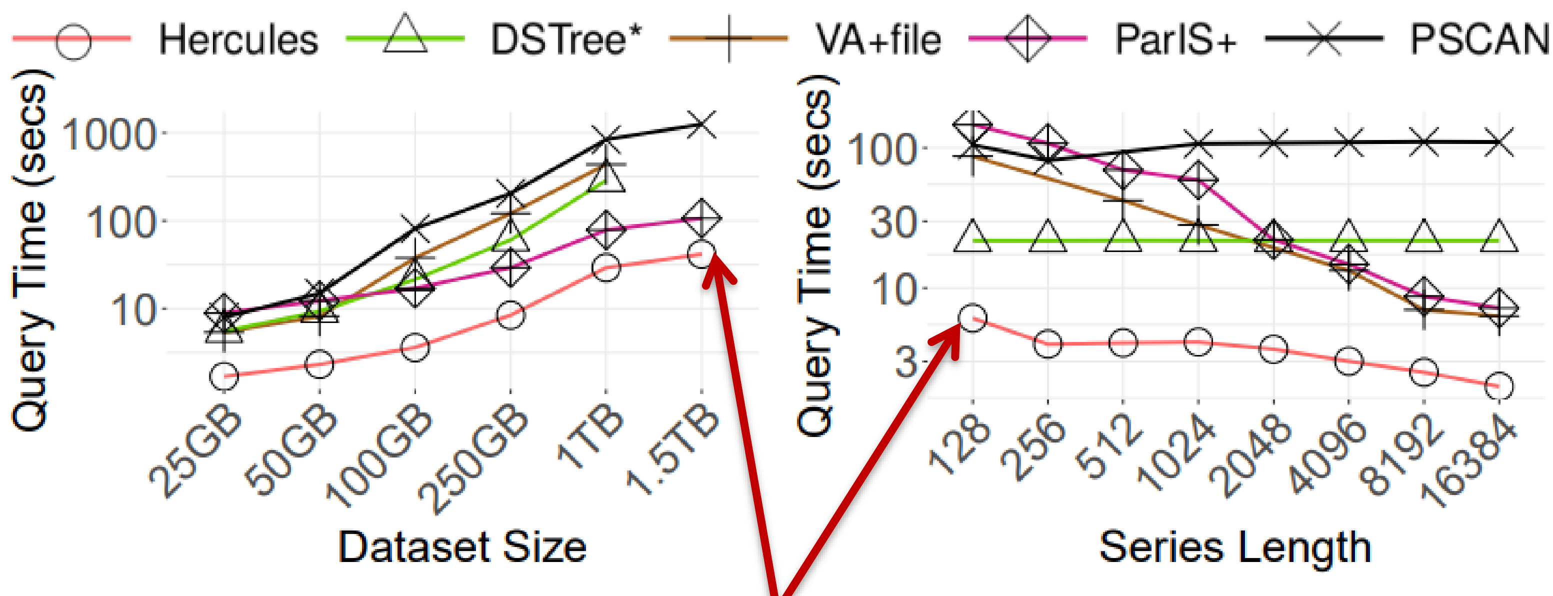
Query Performance with Increased Query Difficulty (Seismic100GB)

Publications

Echihabi-PVLDB'22

Hercules
Parallel Indexing of Sequences

- Disk-based solution for SIMD, multi-core, multi-socket architectures



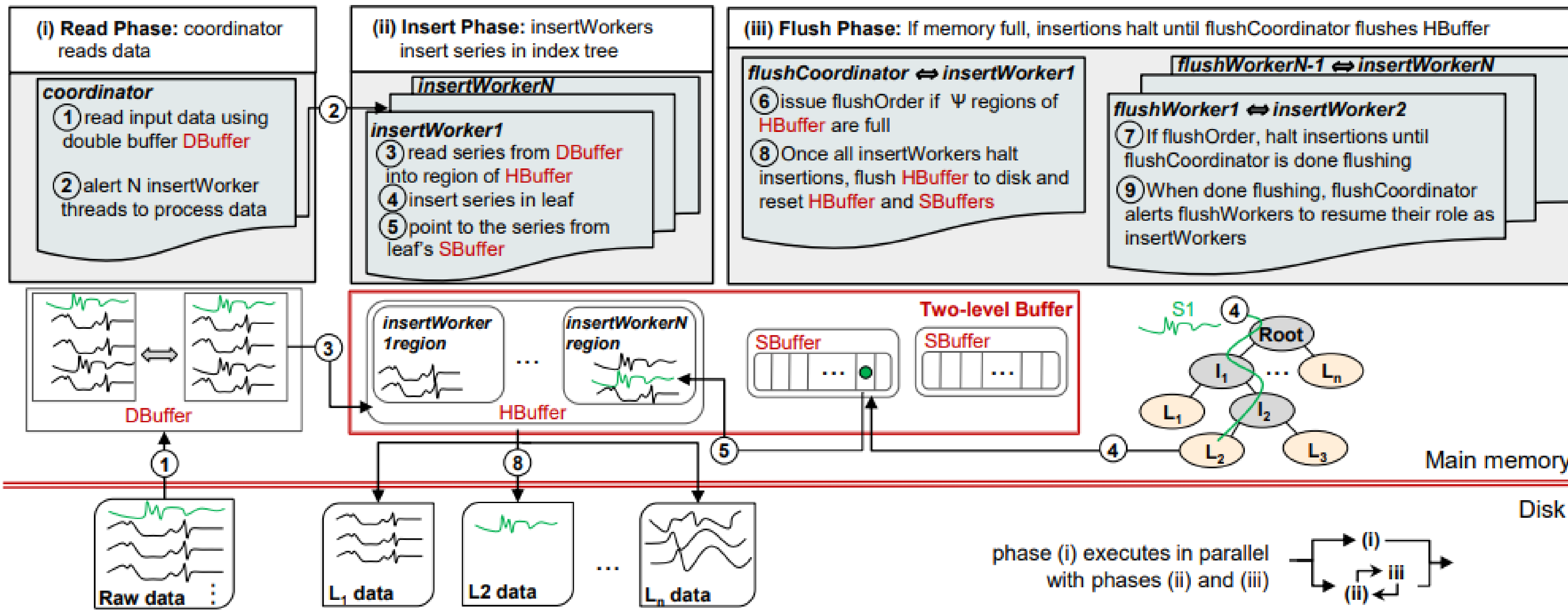
Hercules best overall

<u>Query Performance with Increased Dataset Size and Series Length (Synthetic)</u>

Hercules
Index Building

Publications

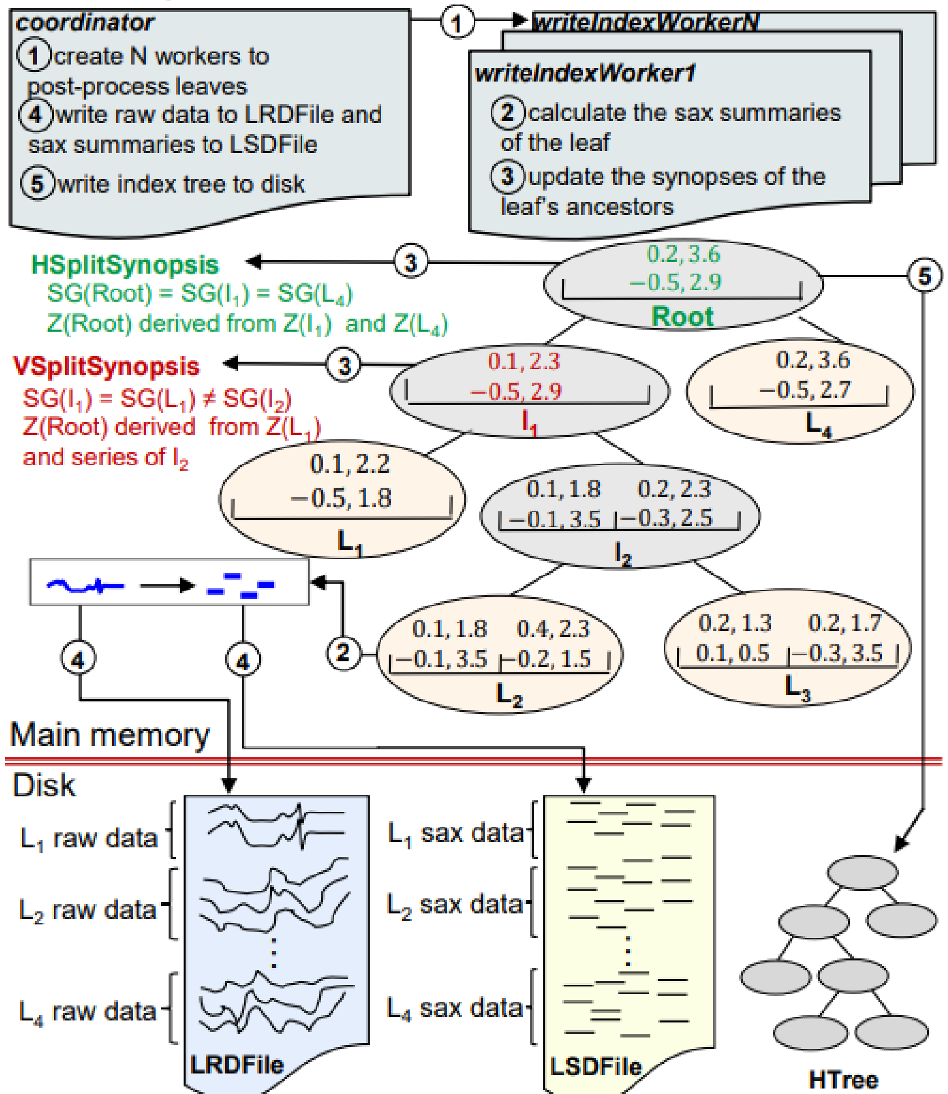
Echihabi-PVLDB'22



Hercules Index Building Workflow

Hercules
Index Writing

Publications
Echihabi-PVLDB'22

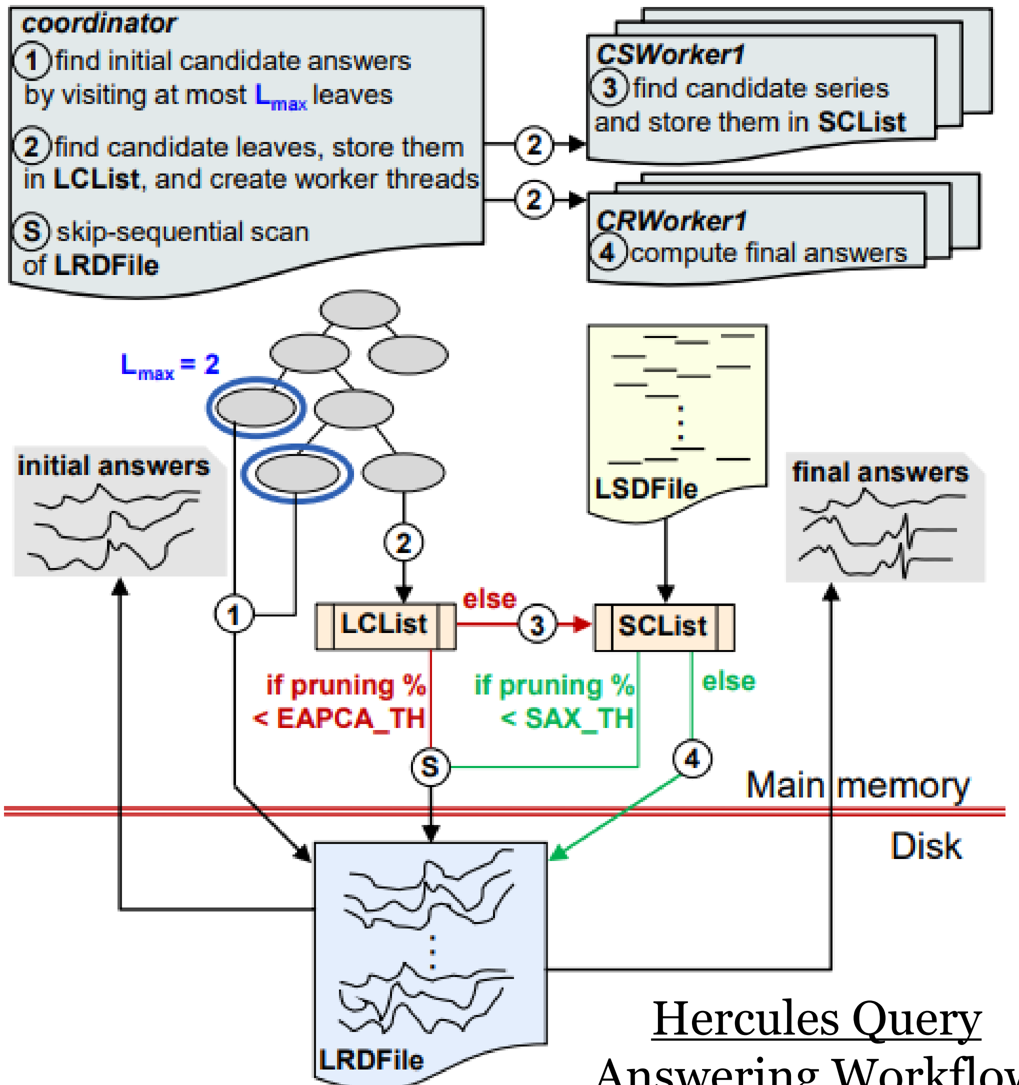


Hercules Index Writing Workflow

Hercules
Query Answering

Publications

Echihabi-PVLDB'22



Hercules Query Answering Workflow

TARDIS

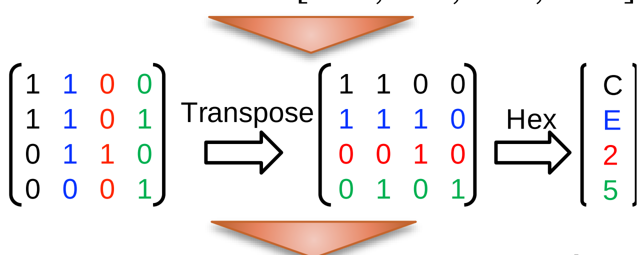
Publications

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

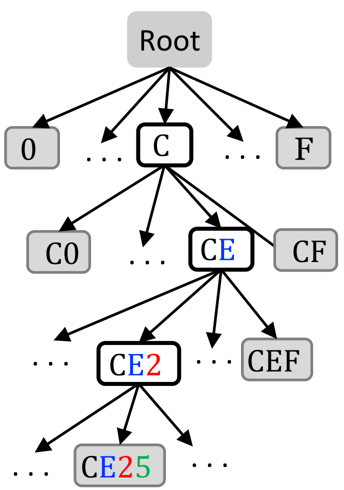
iSAX-T representation

time series SAX: [1100, 1101, 0110, 0001]



SAX						iSAX-T
(T,4,2)	=	{1,	1,	0,	0 }	= C
(T,4,4)	=	{11,	11,	01,	00 }	= CE
(T,4,8)	=	{110,	110,	011,	000 }	= CE2
(T,4,16)	=	{1100,	1101,	0110,	0001}	= CE25

sigTree index



TARDIS

Publications

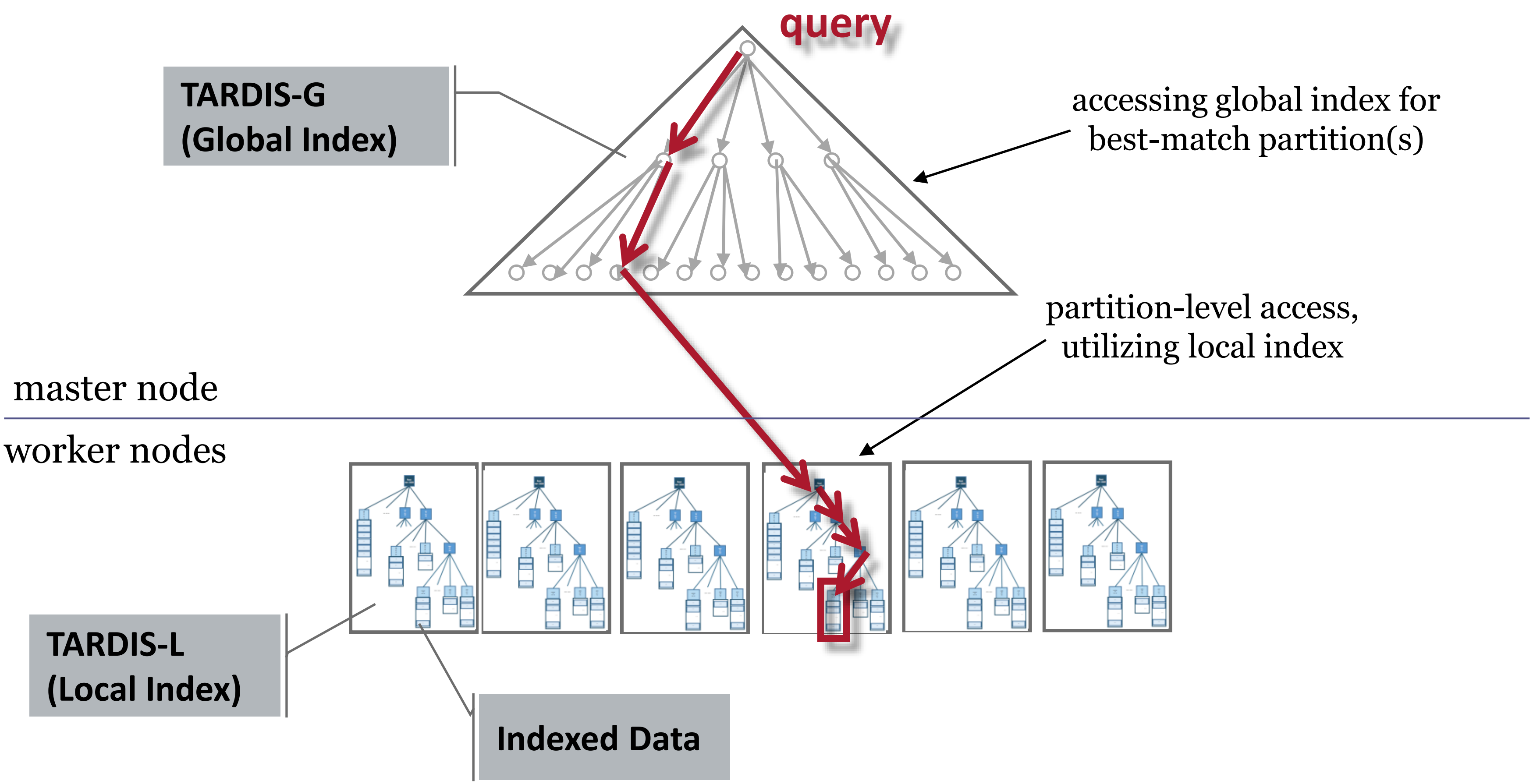
Zhang et al. ICDE'19

- 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



Publications

Zhang et al. ICDE'19

TARDIS

- 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)

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

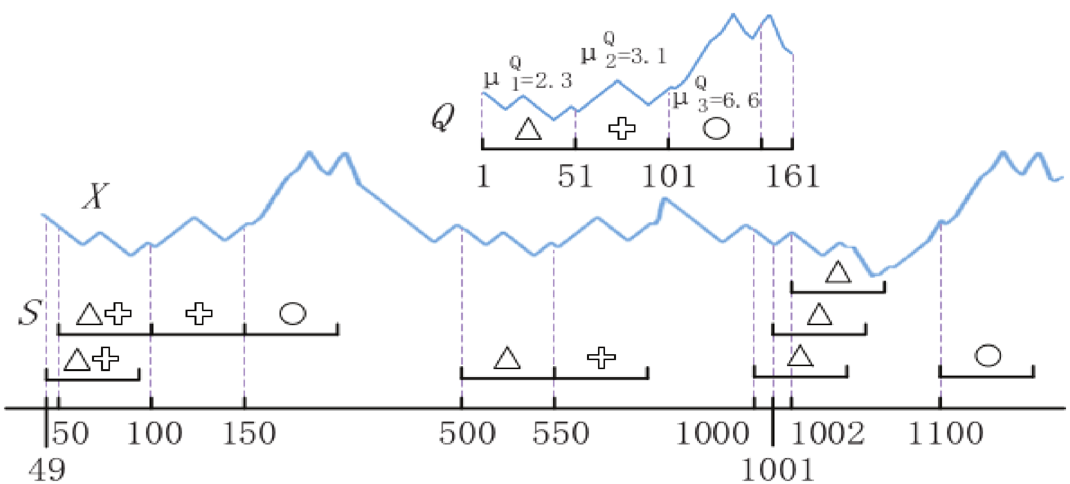
    - users control extent of amplitude scaling and offset shifting
    - normalized subsequence matching is a special case of cNSM

Publications

Wu et al. ICDE'19

KV-match

- 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

Publications

Wu et al. ICDE'19

KV-match

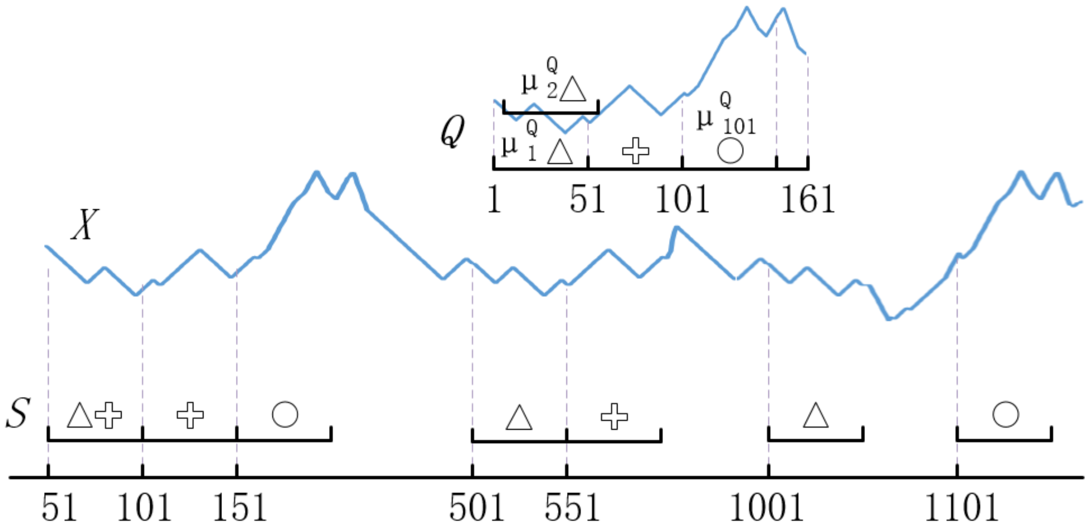
- solution for distributed (HDFS) subsequence similarity search
  - query answering
    - for query Q and corresponding subsequence S
      - segment Q into aligned length-w disjoint windows (requires having several indexes of different lengths)
    - for each window Q_i and S_i
      - filtering condition: 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

Publications

Feng et al.
IEEE Access'20

L-match

- 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

Questions?

Experimental Comparisons: Exact Query Answering

Experimental Framework

Publications

Echihabi-PVLDB'18

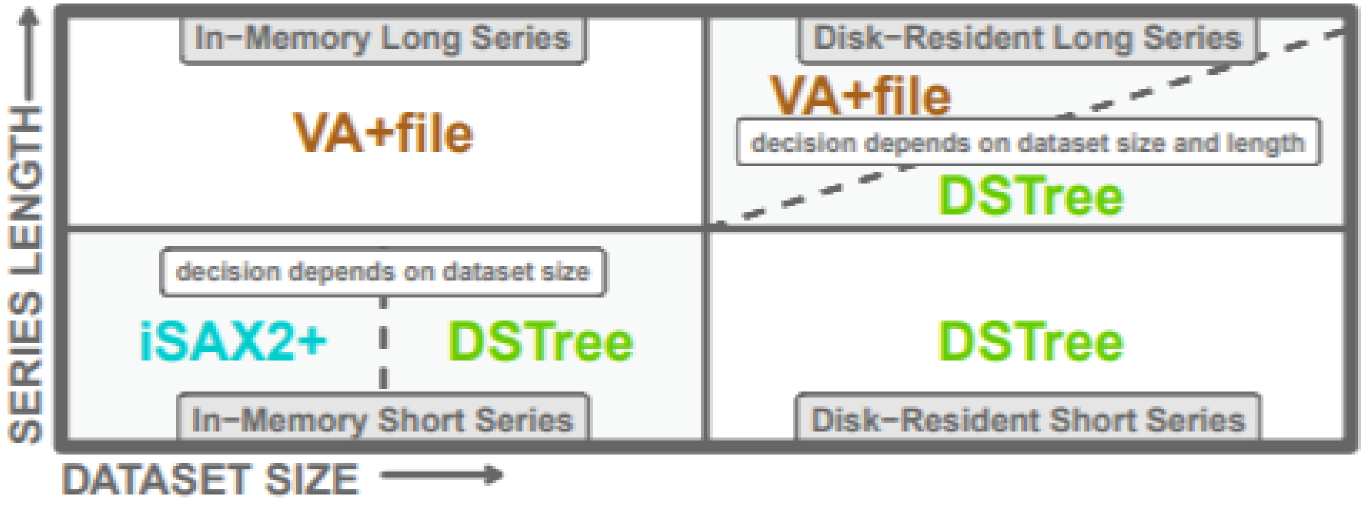
- Hardware
  - HDD and SSD
- Datasets
  - Synthetic (25GB to 1TB) and 4 real (100 GB)
- Exact Query Workloads
  - 100 – 10,000 queries
- Performance measures
  - Time, #disk accesses, footprint, pruning, Tightness of Lower Bound (TLB), etc.
- C/C++ methods (4 methods reimplemented from scratch)

- Procedure:
  - Step 1: Parametrization
  - Step 2: Evaluation of individual methods
  - Step 3: Comparison of best methods

Publications
Echihabi-PVLDB‘18

Recommendations

Scenario: Indexing and answering 10K exact queries on HDD



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
    - New 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

Insights

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

Experimental Comparisons: Approximate Query Answering

Experimental Framework

Publications

Echihabi-PVLDB'19

- Datasets
  - In-memory and disk-based datasets
  - Synthetic data modeling financial time series
  - Four real datasets from deep learning, computer vision, seismology, and neuroscience (25GB-250GB)
- Query Workloads
  - 100 – 10,000 kNN queries k in [1,100]
  - ng-approximate and δ-ε-approximate queries (exact queries used as yardstick)
- C/C++ methods (3 methods reimplemented from scratch)
- Performance measures
  - Efficiency: time, throughput, #disk accesses, % of data accessed
  - Accuracy: average recall, mean average precision, mean relative error
- Procedure:
  - Step 1: Parametrization
  - Step 2: Evaluation of indexing/query answering scalability in-memory
  - Step 3: Evaluation of indexing/query answering scalability on-disk
  - Step 4: Additional experiments with best-performing methods on disk

Approximate Methods Covered in Study

		Matching Accuracy				Representation		Implementation		
		exact	ng-appr.	ϵ-appr.	δ-ϵ-appr.	Raw	Reduced	Original	New	Disk-resident Data
Graphs	HNSW		[99]			✓		C++		
	NSG		[58]			✓		C++		

Approximate Methods Covered in Study

		Matching Accuracy				Representation		Implementation		
		exact	ng-appr.	ϵ-appr.	δ-ϵ-appr.	Raw	Reduced	Original	New	Disk-resident Data
Graphs	HNSW		[99]			✓		C++		
	NSG		[58]			✓		C++		
Inv. Indexes	IMI		[16, 60]				OPQ	C++		✓

Approximate Methods Covered in Study

		Matching Accuracy				Representation		Implementation		
		exact	ng-appr.	ϵ-appr.	δ-ϵ-appr.	Raw	Reduced	Original	New	Disk-resident Data
Graphs	HNSW		[99]			✓		C++		
	NSG		[58]			✓		C++		
Inv. Indexes	IMI		[16, 60]				OPQ	C++		✓
LSH	QALSH				[69]		Signatures	C++		
	SRS				[136]		Signatures	C++		

Approximate Methods Covered in Study

		Matching Accuracy				Representation		Implementation		
		exact	ng-appr.	ϵ-appr.	δ-ϵ-appr.	Raw	Reduced	Original	New	Disk-resident Data
Graphs	HNSW		[99]			✓		C++		
	NSG		[58]			✓		C++		
Inv. Indexes	IMI		[16, 60]				OPQ	C++		✓
LSH	QALSH				[69]		Signatures	C++		
	SRS				[136]		Signatures	C++		
Scans	VA+file	[55]	•	•	•		DFT	MATLAB	C	✓

- **Our extensions**

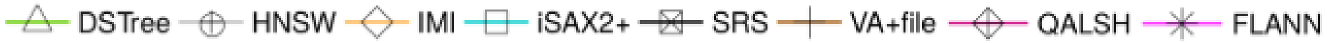
Approximate Methods Covered in Study

		Matching Accuracy				Representation		Implementation		
		exact	ng-appr.	ϵ-appr.	δ-ϵ-appr.	Raw	Reduced	Original	New	Disk-resident Data
Graphs	HNSW		[99]			✓		C++		
	NSG		[58]			✓		C++		
Inv. Indexes	IMI		[16, 60]				OPQ	C++		✓
LSH	QALSH				[69]		Signatures	C++		
	SRS				[136]		Signatures	C++		
Scans	VA+file	[55]	•	•	•		DFT	MATLAB	C	✓
Trees	Flann		[107]			✓		C++		
	DSTree	[146]	[146]	•	•		EAPCA	Java	C	✓
	HD-index		[11]				Hilbert keys	C++		✓
	iSAX2+	[30]	[30]	•	•		iSAX	C#	C	✓

- **Our extensions**

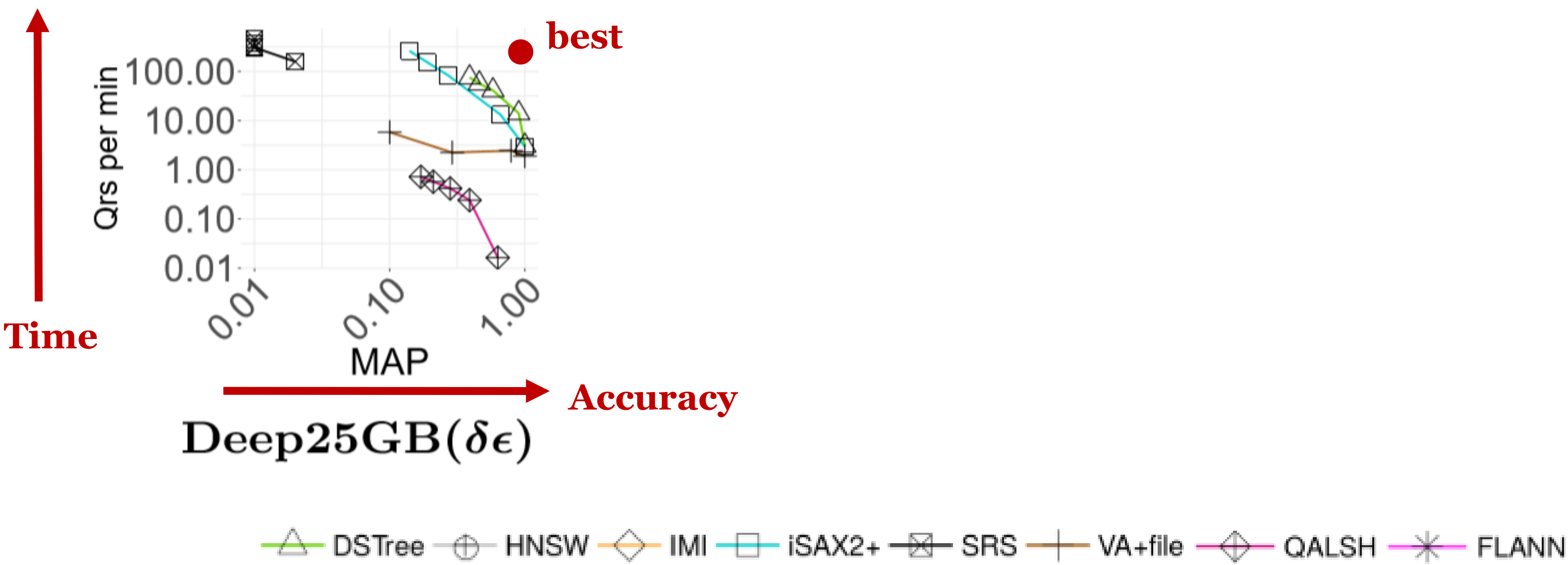
Unexpected Results

- New data series extensions are the overall winners even for general high-d vectors
  - perform the best for approximate queries with probabilistic guarantees (δ-ε-approximate search)



Unexpected Results

- New data series extensions are the overall winners even for general high-d vectors
  - perform the best for approximate queries with probabilistic guarantees (δ-ε-approximate search)



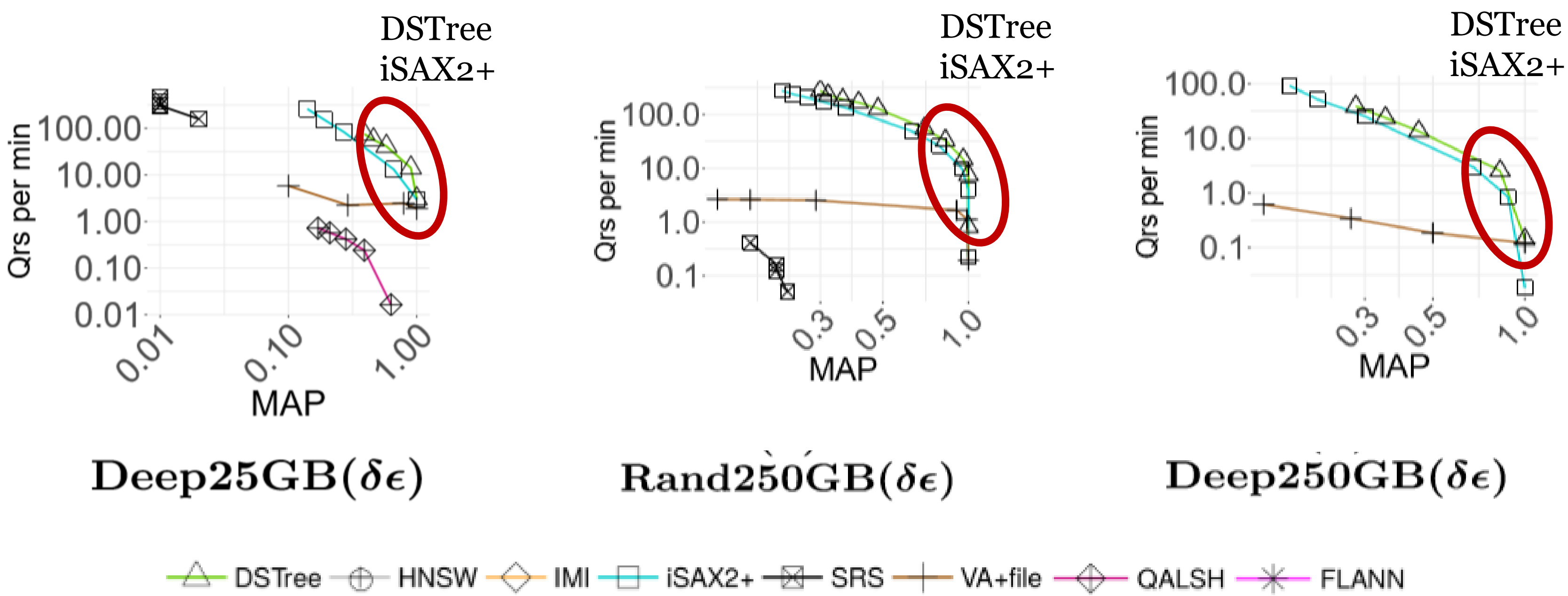
Unexpected Results

- New data series extensions are the overall winners even for general high-d vectors
  - perform the best for approximate queries with probabilistic guarantees (δ-ε-approximate search), in-memory

Deep25GB(δϵ)

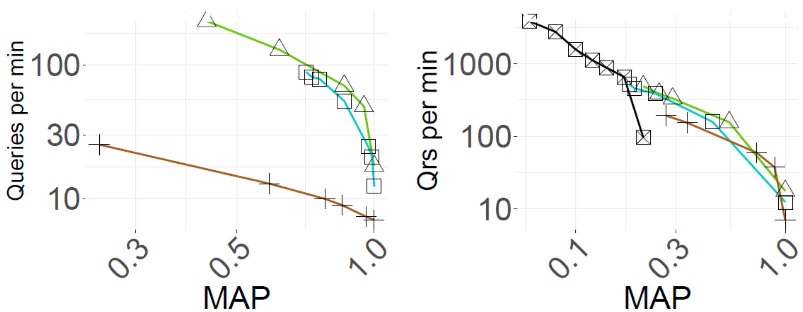
Unexpected Results

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

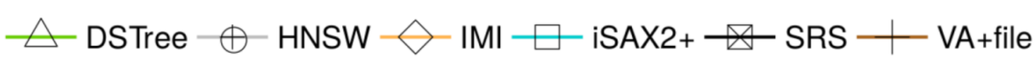


Unexpected Results

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

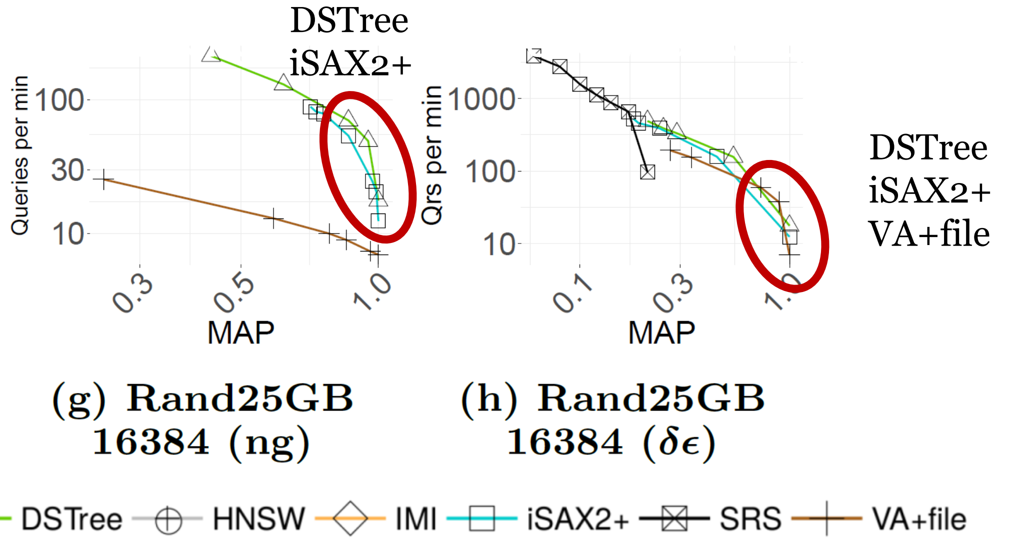


(g) Rand25GB 16384 (ng) **(h) Rand25GB 16384 ($\delta\epsilon$)**



Unexpected Results

- New data series extensions are the overall winners even for general high-d vectors
  - 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

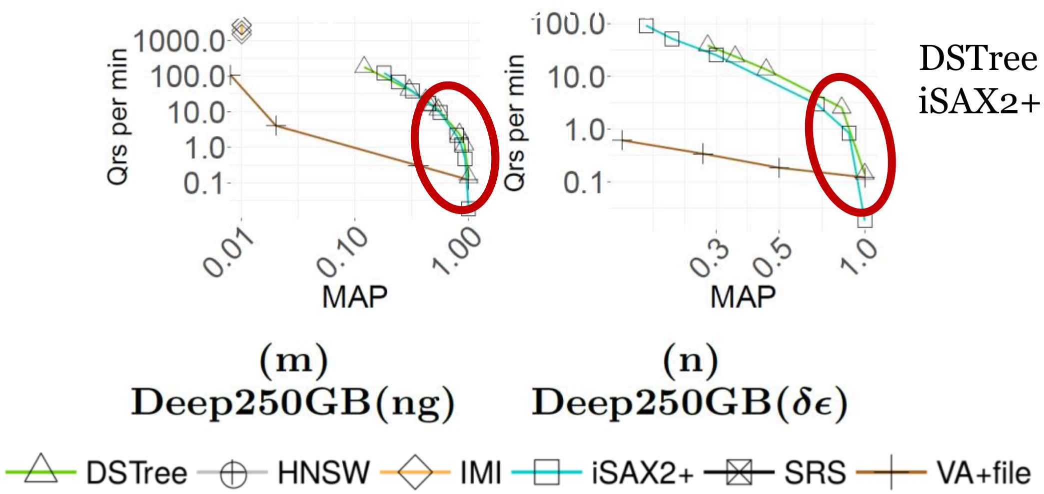


(g) Rand25GB 16384 (ng)

(h) Rand25GB 16384 ($\delta\epsilon$)

Unexpected Results

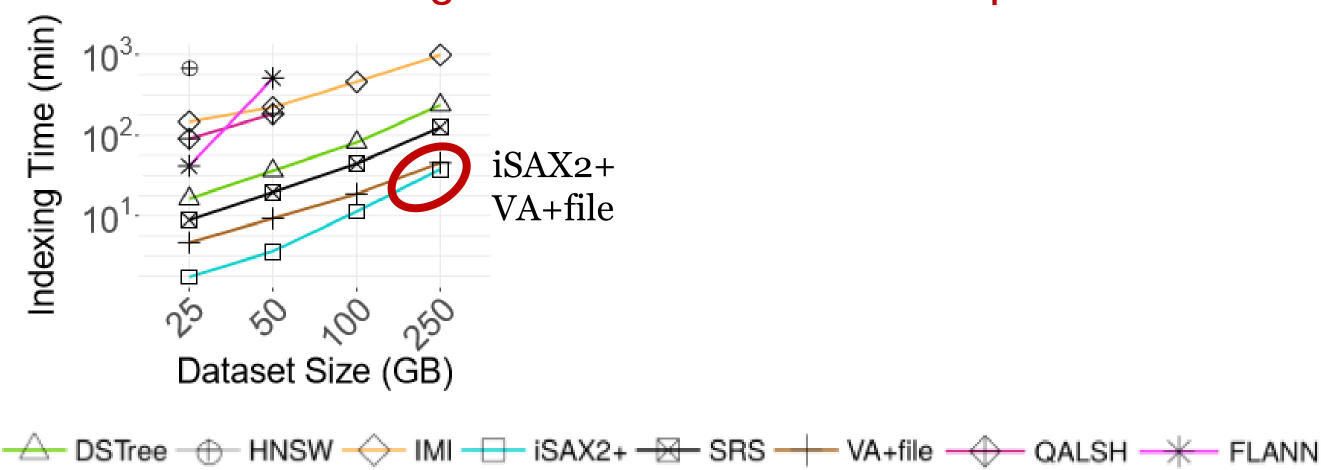
- New data series extensions are the overall winners even for general high-d vectors
  - perform the best for approximate queries with probabilistic guarantees (δ-ε-approximate search), in-memory and on-disk
  - perform the best for long vectors, in-memory and on-disk
  - perform the best for disk-resident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



(m) Deep250GB(ng) (n) Deep250GB(δϵ)

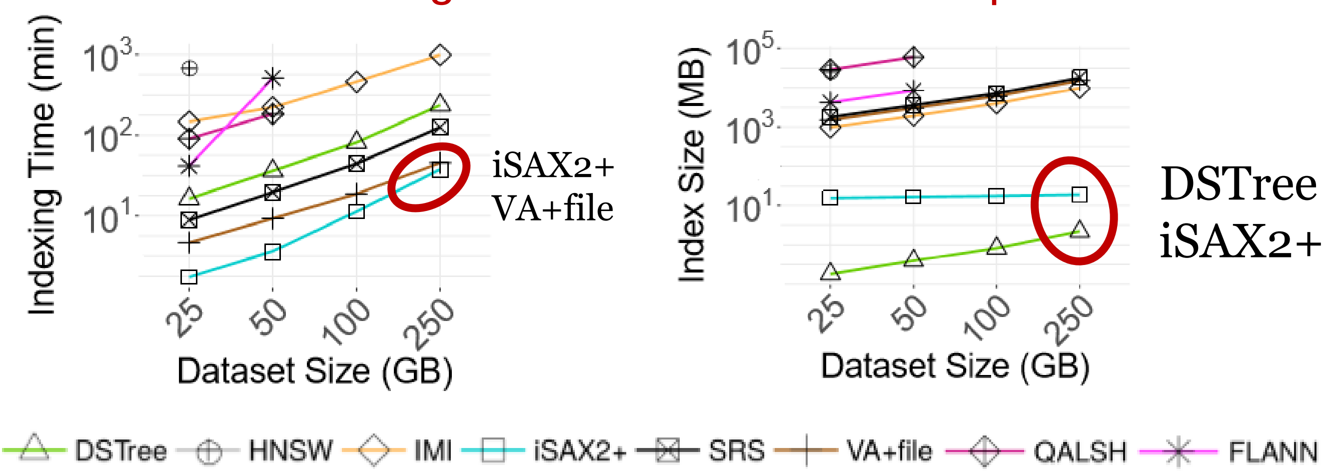
Unexpected Results

- New data series extensions are the overall winners even for general high-d vectors
  - 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
  - are fastest at indexing and have the lowest footprint



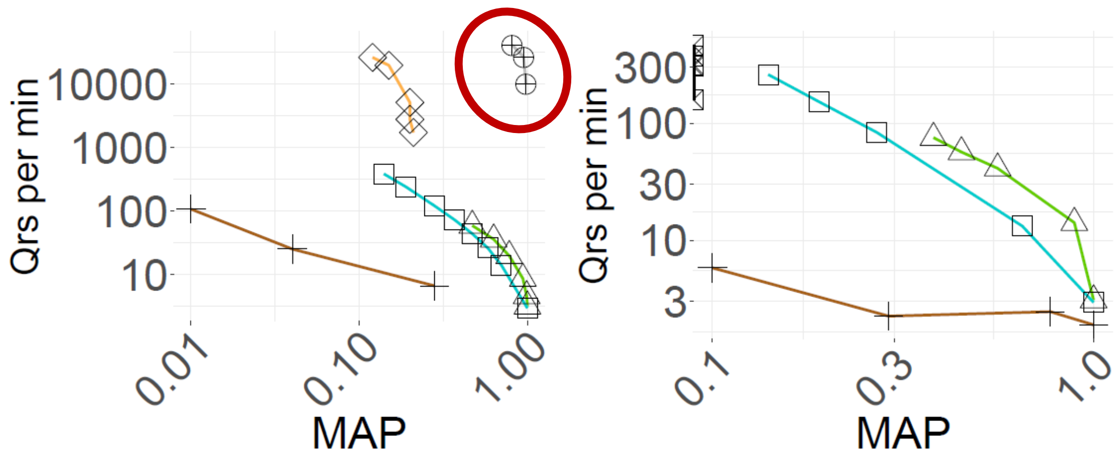
Unexpected Results

- New data series extensions are the overall winners even for general high-d vectors
  - 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
  - are fastest at indexing and have the lowest footprint



Unexpected Results

- New data series extensions are the overall winners even for general high-d vectors
  - 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
  - are fastest at indexing and have the lowest footprint



(s) **Deep25GB(ng)** (t) **Deep25GB($\delta\epsilon$)**

Only exception is HNSW winning on in-memory data, with a prebuilt index (no guarantees for the answers)

Insights

Exciting research direction for approximate similarity search in high-d spaces:

Insights

Exciting research direction for approximate similarity search in high-d spaces:

Currently two main groups of solutions exist:

approximate search solutions
without guarantees
relatively efficient

Insights

Exciting research direction for approximate similarity search in high-d spaces:

Currently two main groups of solutions exist:

approximate search solutions
without guarantees
relatively efficient

approximate search solutions
with guarantees
relatively slow

Insights

Exciting research direction for approximate similarity search in high-d spaces:

Currently two main groups of solutions exist:

approximate search solutions	approximate search solutions
without guarantees	with guarantees
relatively efficient	relatively slow

We show that it is possible to have efficient approximate algorithms with guarantees

Insights

Approximate state-of-the-art techniques for high-d vectors are not practical:

Insights

Approximate state-of-the-art techniques for high-d vectors are not practical:

- LSH-based techniques
  - slow, high-footprint, low accuracy (recall/MAP)

Insights

Approximate state-of-the-art techniques for high-d vectors are not practical:

- LSH-based techniques
  - slow, high-footprint, low accuracy (recall/MAP)
- kNNG-based techniques
  - slow indexing, difficult to tune, in-memory, no guarantees

Insights

Approximate state-of-the-art techniques for high-d vectors are not practical:

- LSH-based techniques
  - slow, high-footprint, low accuracy (recall/MAP)
- kNNG-based techniques
  - slow indexing, difficult to tune, in-memory, no guarantees
- Quantization-based techniques
  - slow indexing, difficult to tune, no guarantees

Insights

Approximate state-of-the-art techniques for high-d vectors are not practical:

- LSH-based techniques
  - slow, high-footprint, low accuracy (recall/MAP)
- kNNG-based techniques
  - slow indexing, difficult to tune, in-memory, no guarantees
- Quantization-based techniques
  - slow indexing, difficult to tune, no guarantees

All suffer a serious limitation:
accuracy determined during index-building & query answering

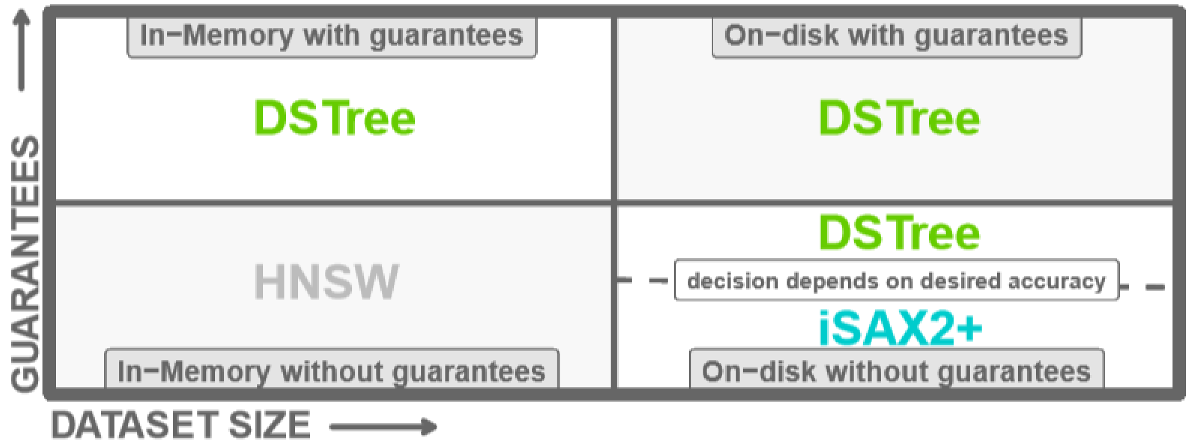
Recommendations for approx. techniques

Data series approaches are the overall winners!

The only exception is HNSW for in-memory ng-approximate queries using an existing index

Recommendations

Scenario: Answering a query workload using an existing index



Questions?

AI and Similarity Search

AI and Similarity Search

- Representation Learning
  - Learned summarizations for data series
- Search and Indexing
  - Learned indexes
  - Similarity search on deep network embeddings

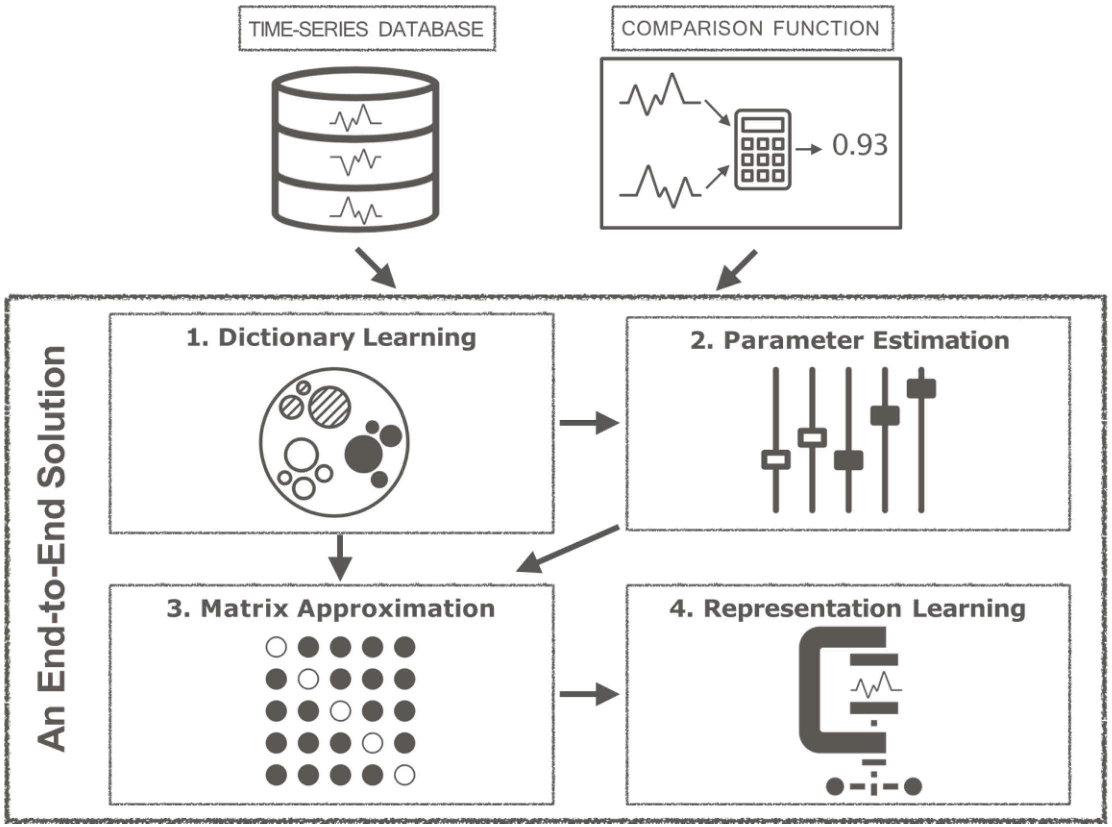
AI and Similarity Search

Representation Learning for Sequences

Publications

Paparrizos - PVLDB'19

- GRAIL
  - learns representations that preserve a user-defined comparison function
  - for a given comparison function:
    - extracts landmark series using clustering
    - optimizes parameters
    - exploits approximations for kernel methods to construct representations by expressing each series as a combination of the landmark series



Slide by J. Paparrizos

AI and Similarity Search

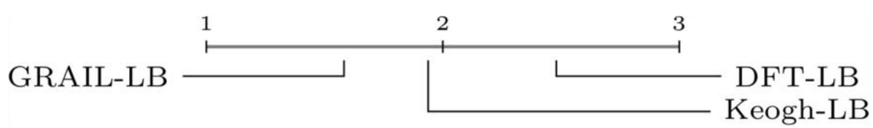
Representation Learning for Sequences

Publications

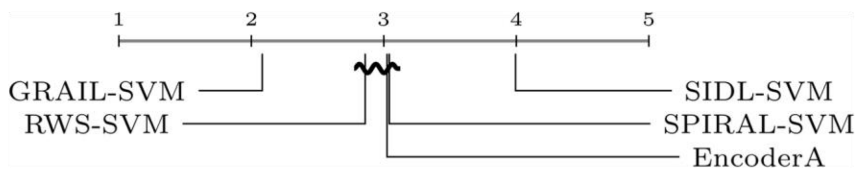
Paparrizos - PVLDB'19

- GRAIL
  - uses the learned representations for querying, classification, clustering, …

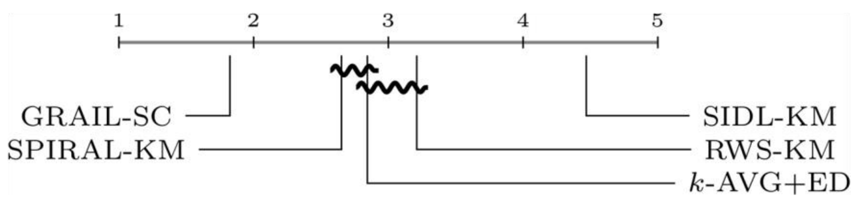
QUERYING: GRAIL Lower Bound vs. Lower Bounds for DFT & DTW



CLASSIFICATION: GRAIL with SVM vs. other Learned Representations



CLUSTERING: GRAIL with Spectral Clustering vs. other Learned Representations



Slide by J. Paparrizos

AI and Similarity Search

Representation Learning for Sequences

Publications

Wang - KDD'21

- Series Approximation Network (SEAnet)
  - novel autoencoder architecture
  - learns deep embedding approximations
  - uses those for similarity search

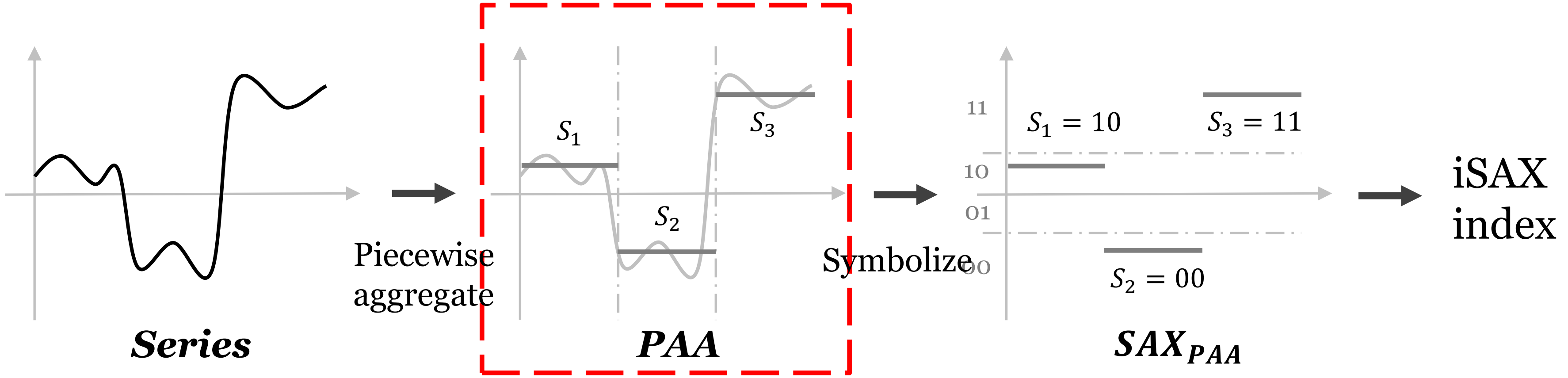
AI and Similarity Search

Representation Learning for Sequences

Publications

Wang - KDD'21

- Series Approximation Network (SEAnet)
  - novel autoencoder architecture
  - learns deep embedding approximations
  - uses those for similarity search

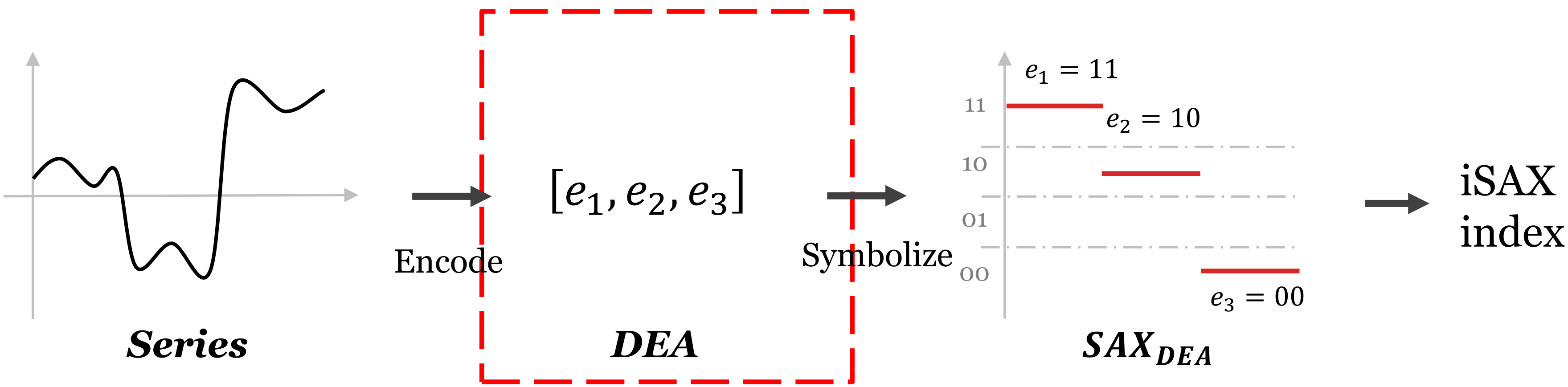


AI and Similarity Search
Representation Learning for Sequences

Publications

Wang - KDD’21

- Series Approximation Network (SEAnet)
  - novel autoencoder architecture
  - learns deep embedding approximations
  - uses those for similarity search



AI and Similarity Search

Representation Learning for Sequences

Publications

Wang - KDD'21

- Series Approximation Network (SEAnet)
  - is an exponentially dilated ResNet architecture + Sum of Squares regularization
  - minimizes
    - reconstruction error
    - difference between distance of two vectors in embedded space and distance in original space

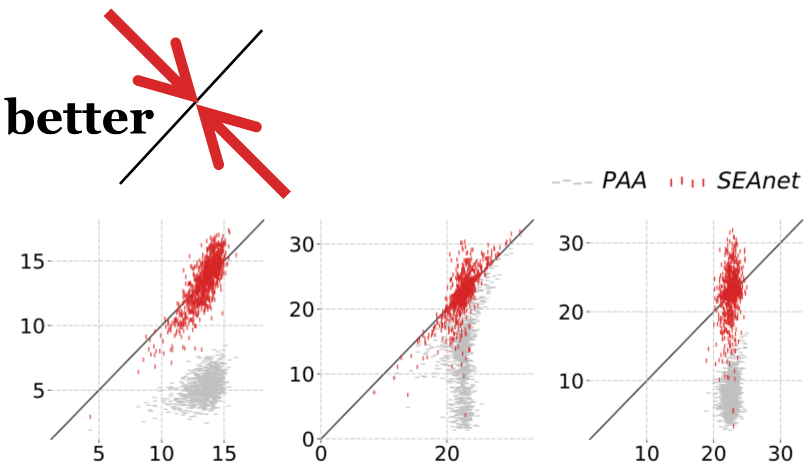
AI and Similarity Search

Representation Learning for Sequences

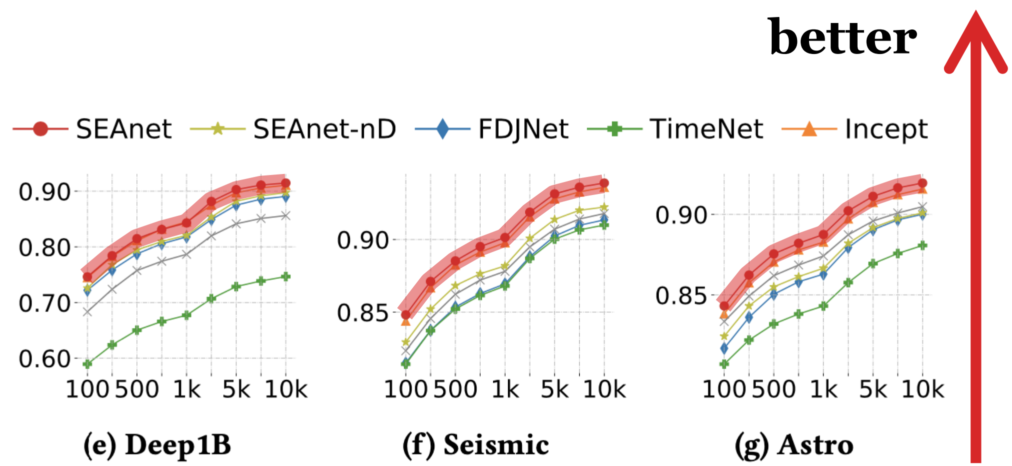
Publications

Wang - KDD'21

- Series Approximation Network (SEAnet)
  - is an exponentially dilated ResNet architecture + Sum of Squares regularization
  - minimizes
    - reconstruction error
    - difference between distance of two vectors in embedded space and distance in original space



(e) Deep1B (f) Seismic (g) Astro



(e) Deep1B (f) Seismic (g) Astro

AI and Similarity Search
Search and Indexing

- Search and Indexing
  - Problem:
    - Sequence similarity search is hard
    - Massive datasets and high dimensionality in 100s-1000s
    - Sophisticated indexing structures and search algorithms
  - Solutions:
    - Learned Indexes
    - Improve search efficiency using deep learning
    - Indexing for learned embeddings

AI and Similarity Search
Search and Indexing

Publications

Kraska-SIGMOD'18

Al-Mamun-SIGSPATIAL'20

- Learned Indexes:
  - Main idea: replace an index with a learned model
    - One-dimensional learned indexes
      - Seminal work: The Case for Learned Indexes
    - Multi-dimensional indexes
      - Exhaustive tutorial on this topic at SIGSPATIAL'20: https://www.cs.purdue.edu/homes/aref/learned-indexes-tutorial.html
    - Some initial attempts for similarity search
  - Main challenges for multi-dimensional indexes:
    - How to sort the data?
    - How to correct prediction errors?
    - Which ML model to choose?
    - How to store the data?
    - How to learn indexes specifically for (the high-d) sequences?

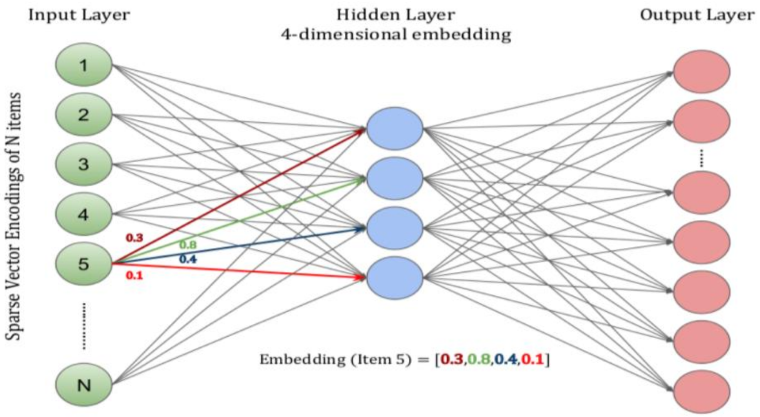
AI and Similarity Search
Search and Indexing

Publications

Echihabi-PVLDB'19

- Indexing Deep Network Embeddings (DNE)

sequences
text
images
video
graphs
…



deep embeddings
high-d vectors learned using a DNN

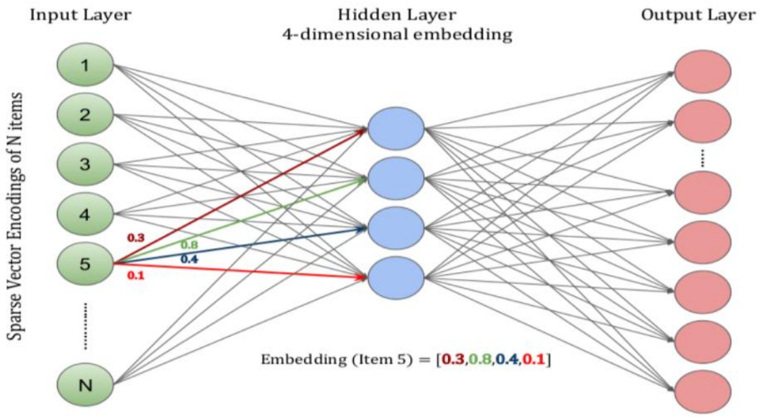
AI and Similarity Search
Search and Indexing

Publications

Echihabi-PVLDB'19

- Indexing Deep Network Embeddings (DNE)

sequences
text
images
video
graphs
...



deep embeddings
high-d vectors learned using a DNN

  - Data series techniques provide effective/scalable similarity search over DNE
  - They outperform hashing-based, quantization-based inverted indexes and kNN graphs on many scenarios

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

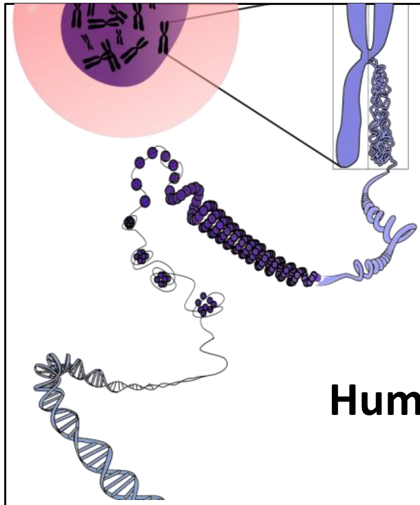
Palpanas-SIGREC'19

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

facebook

Challenges and Open Problems Outline

- sequence management system
- benchmarking
- interactive analytics
- 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

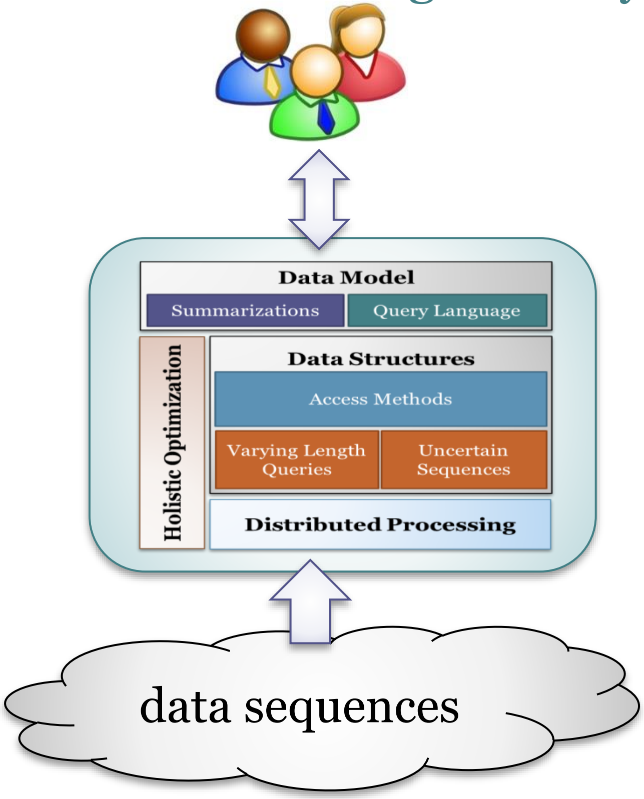
Zoumbatianos ICDE'18

Palpanas-HPCS'17

Palpanas-SIGREC'15

Management System

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



Publications

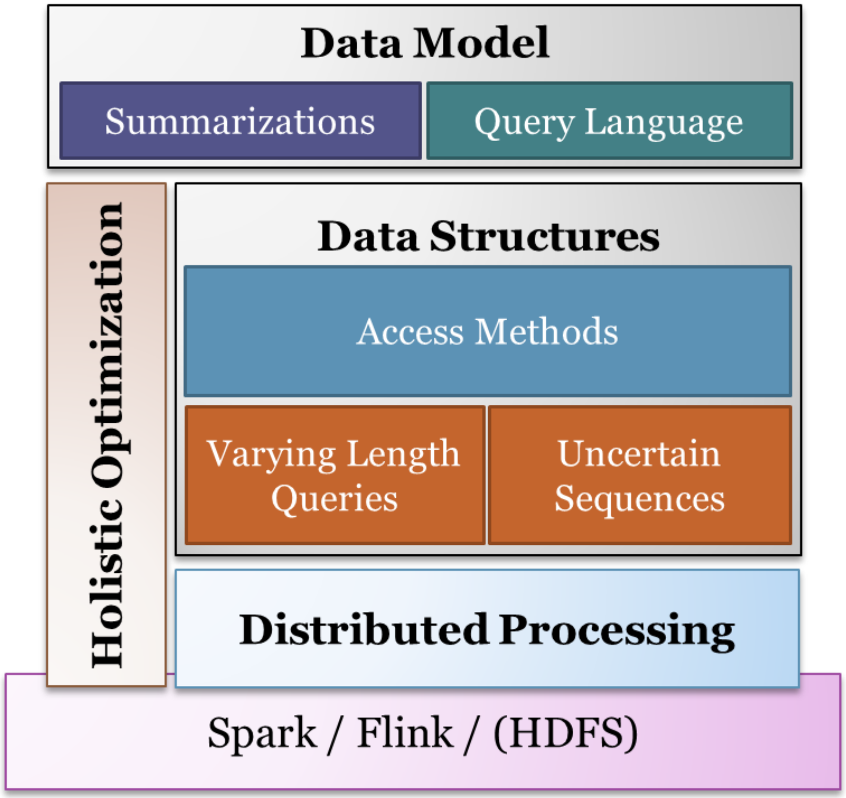
Zoumbatianos ICDE’18

Palpanas-HPCS’17

Palpanas-SIGREC’15

Management System

- Big Sequence Management System

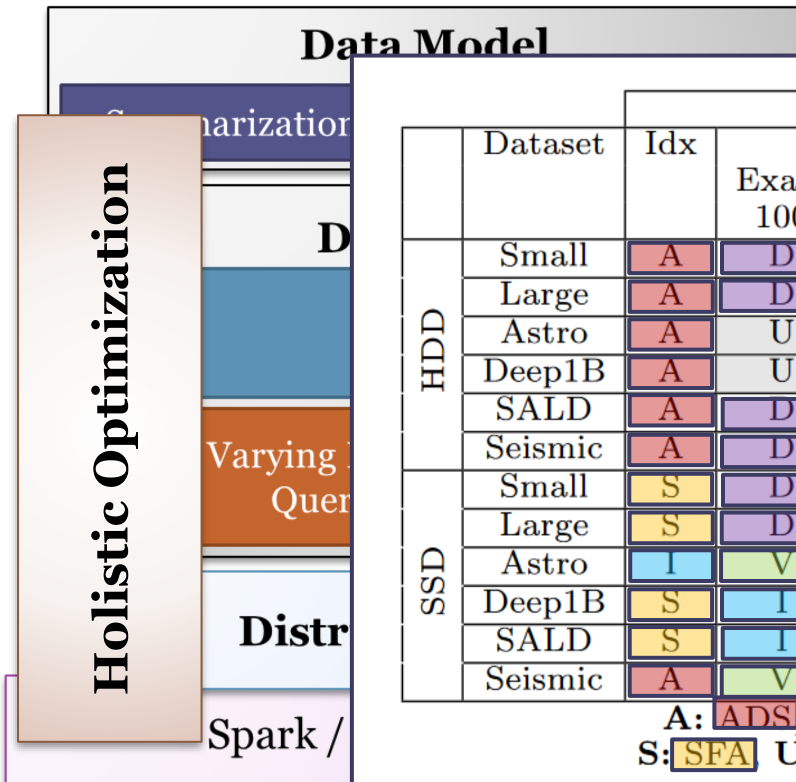


Publications

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

Management System

- Big Sequence Management System



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

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

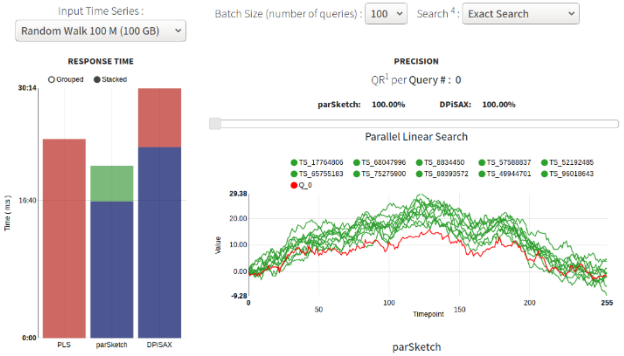
BestNeighbor: Choosing Indexing Method for Given Dataset

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/



- how do these results extend to
  - other data characteristics?
  - more indexing methods?
  - take hardware specifications into consideration?
  - …

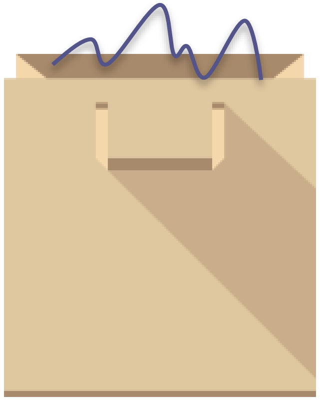
Challenges and Open Problems Outline

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

Previous Studies

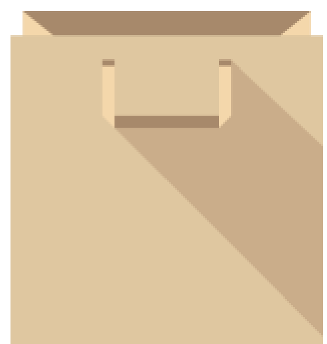
evaluate **performance** of **indexing methods** using **random queries**

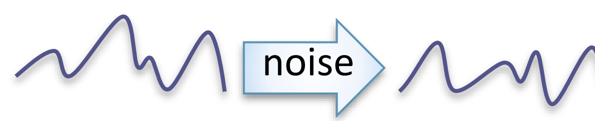
- chosen from the data (with/without noise)



Previous Studies

With or without noise





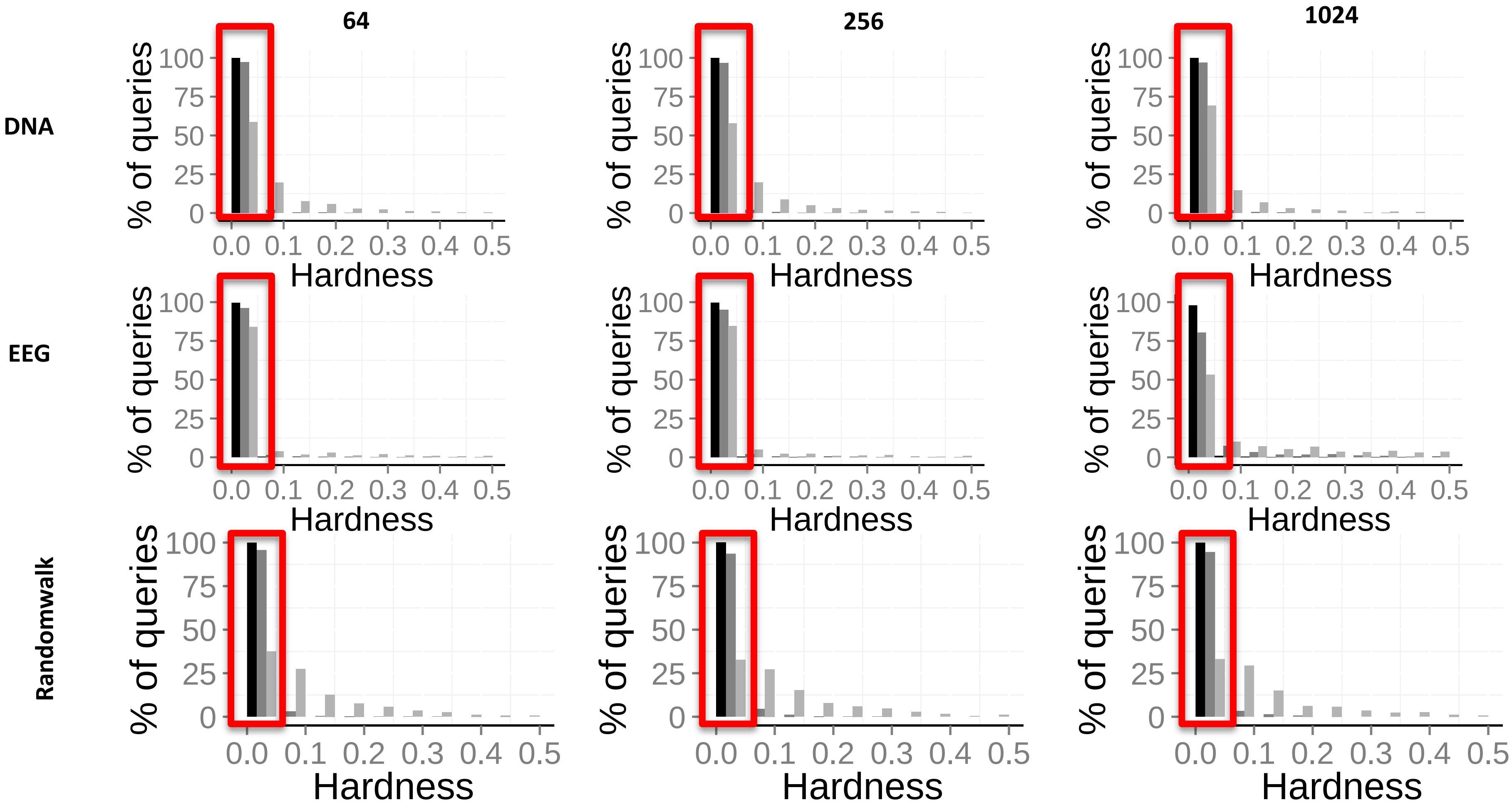
Publications

Zoumbatianos KDD '15

Zoumbatianos TKDE '18

Previous Workloads

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



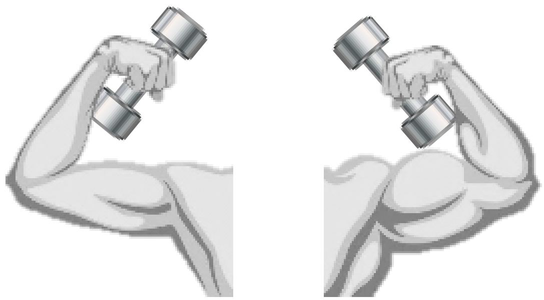
Publications

Zoumbatianos KDD ‘15

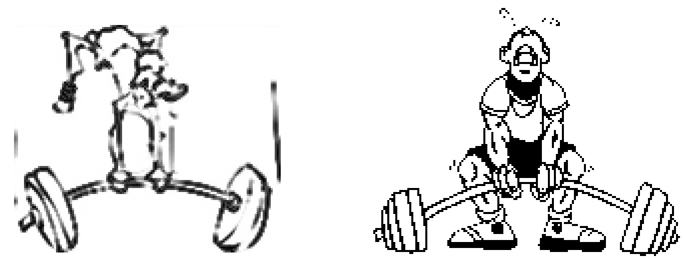
Zoumbatianos TKDE ‘18

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**

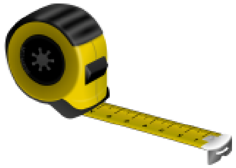
Publications

Zoumbatianos KDD '15

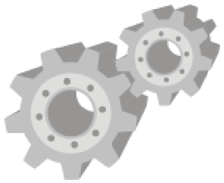
Zoumbatianos TKDE '18

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?

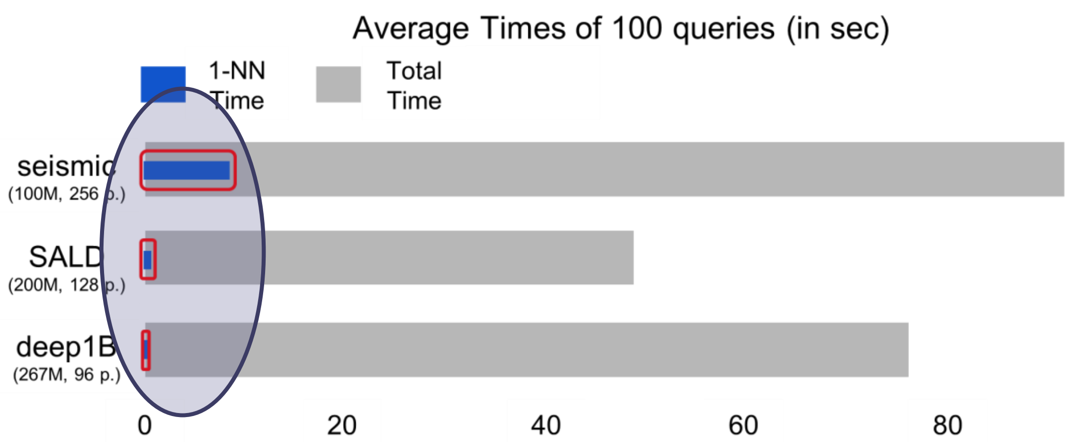
- data series analytics is computationally expensive
  - very high inherent complexity
- may not always be possible to remove delays
  - but could try to hide them!

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

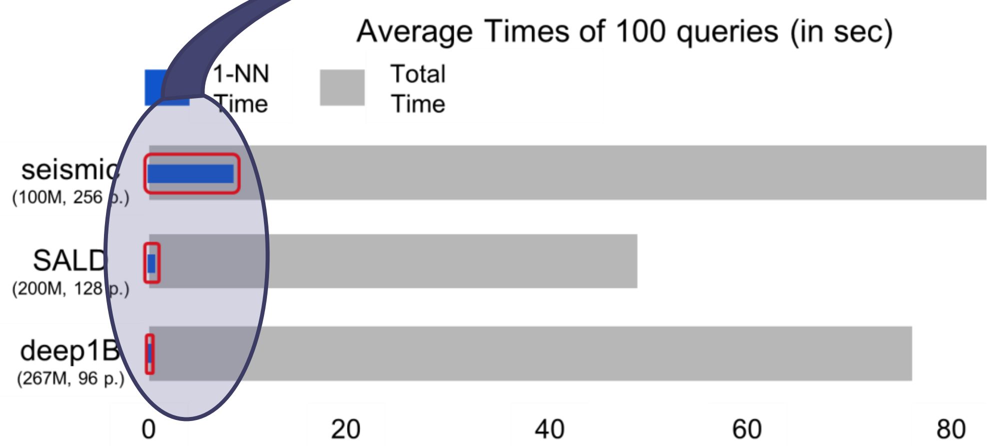


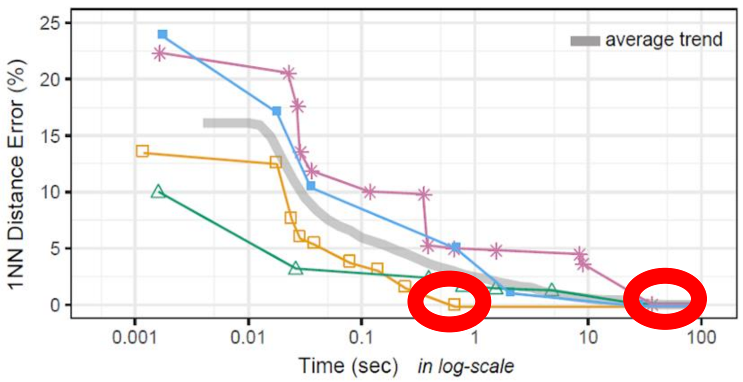
Need for Interactive Analytics

Publications

Gogolou-BigVis'19

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



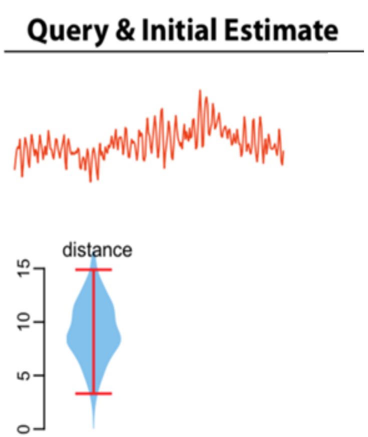


Need for Interactive Analytics

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



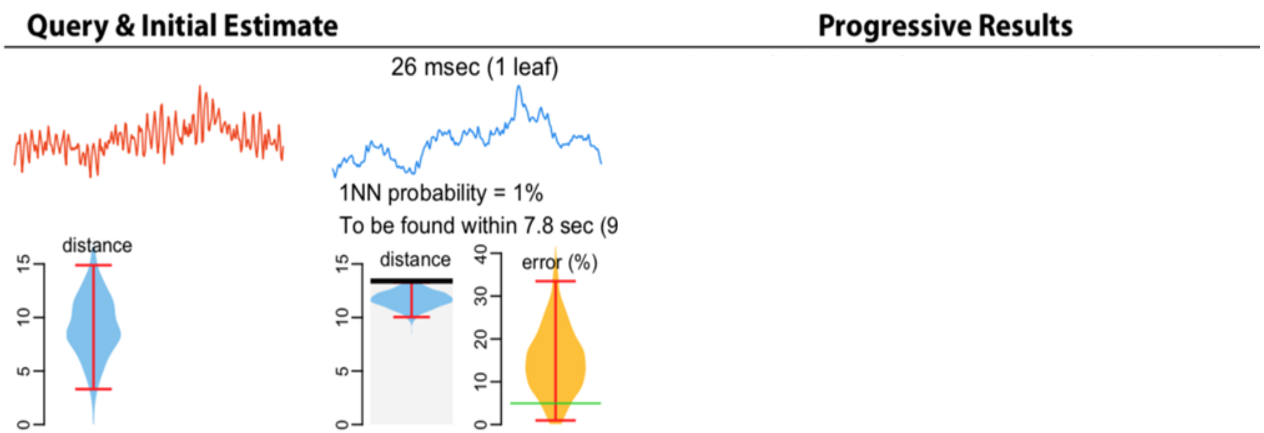
Need for Interactive Analytics

Publications

Gogolou-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



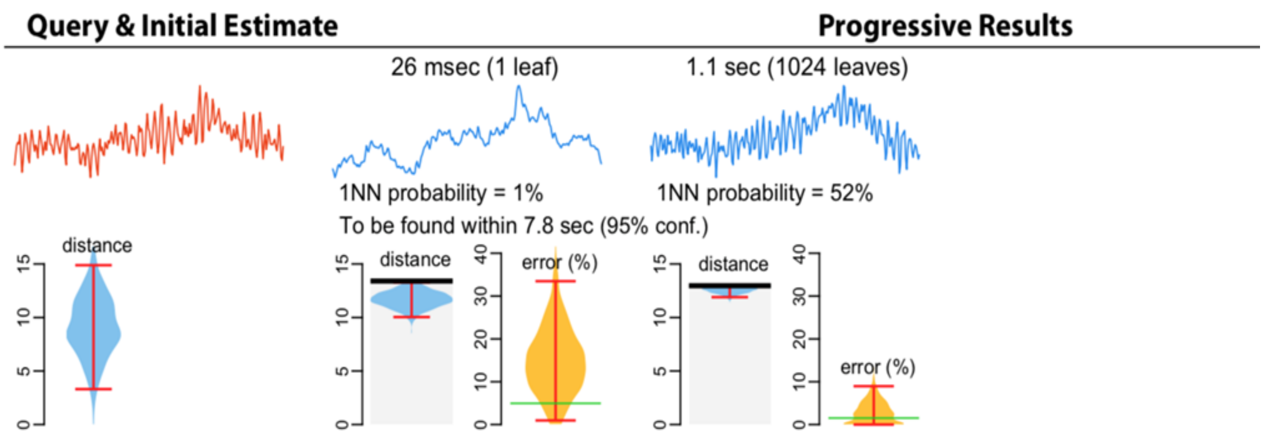
Need for Interactive Analytics

Publications

Gogolou-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



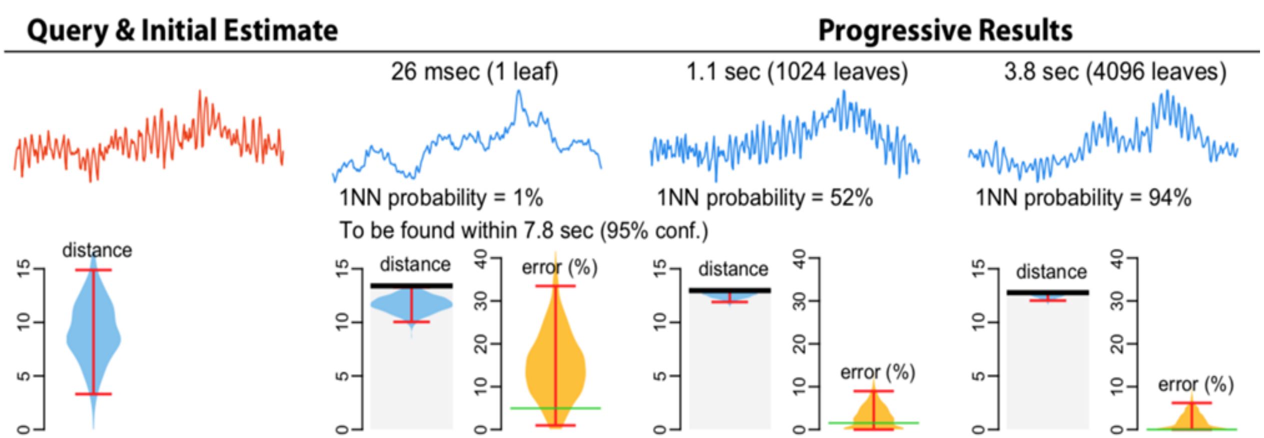
Need for Interactive Analytics

Publications

Gogolou-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

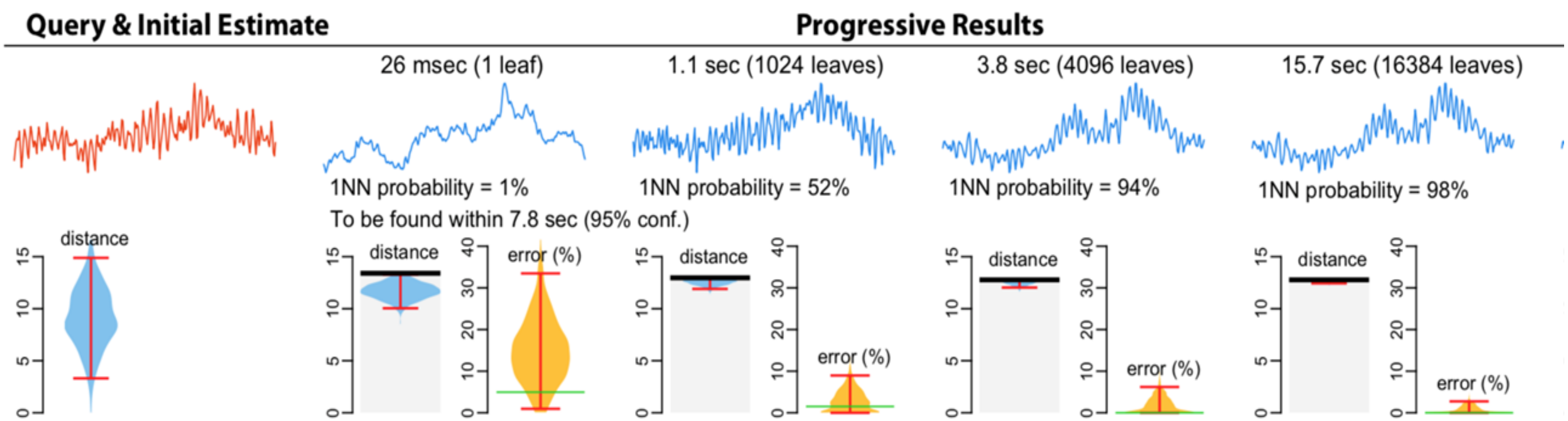


Need for Interactive Analytics

Publications

- Gogolou-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



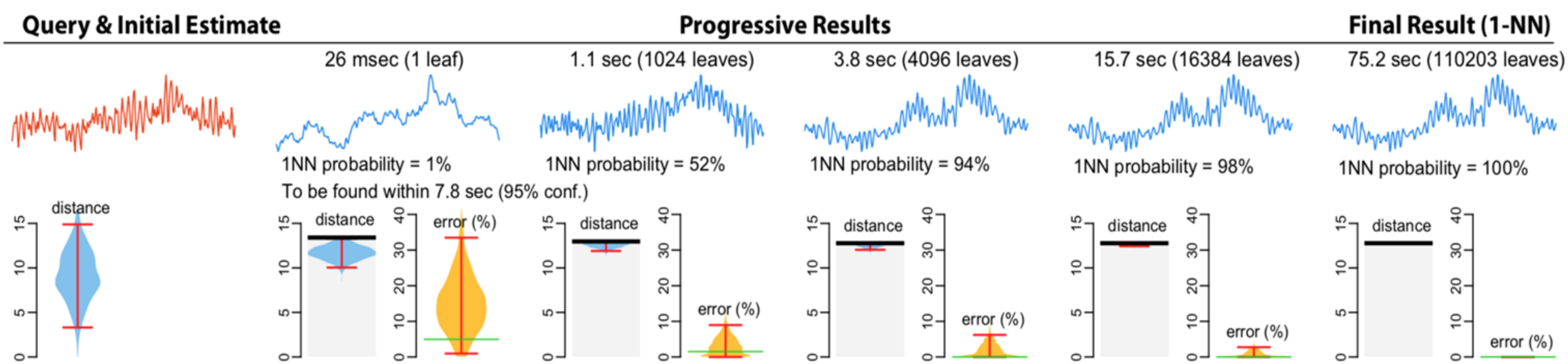
Need for Interactive Analytics

Publications

Gogolou-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
- general high-dimensional vectors
- deep learning

Publications

Echihabi-WIMS'20

Data Series vs. high-d Vectors

- two sides of the same(?) coin
  - data series as multidimensional points
  - for a specific ordering of the dimensions

- **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!
  - design efficient techniques for ng-approximate search
  - devise efficient stopping conditions for δε-approximate search

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**
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
- 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.

References (chronological order)

- 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.
- Katsiaryna Mirylenka, Michele Dallachiesa, Themis Palpanas: Correlation-Aware Distance Measures for Data Series. EDBT 2017: 502-505

References (chronological order)

- Katsiaryna Mirylenka, Michele Dallachiesa, Themis Palpanas: Data Series Similarity Using Correlation-Aware Measures. SSDBM 2017: 11:1-11:12
- G. Chatzigeorgakidis, D. Skoutas, K. Patroumpas, S. Athanasiou, and S. Skiadopoulos. Indexing geolocated time series data. In SIGSPATIAL, pages 19:1–19:10, 2017.
- G. Chatzigeorgakidis, D. Skoutas, K. Patroumpas, S. Athanasiou, S. Skiadopoulos, Map-based visual exploration of geolocated time series, in: Proceedings of the Workshops of the EDBT/ICDT 2018 Joint Conference (EDBT/ICDT 2018), Vienna, Austria, March 26, 2018., 2018, pp. 92–99.
- 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)

References (chronological order)

- Tim Kraska, Alex Beutel, Ed H. Chi, Jeffrey Dean, Neoklis_Polyzotis: The Case for Learned Index Structures. SIGMOD Conference 2018: 489-504
- Botao Peng, Panagiota Fatourou, Themis Palpanas: ParIS: The Next Destination for Fast Data Series Indexing and Query Answering. BigData 2018: 791-800
- 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.

References (chronological order)

- 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
- 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
- G. Chatzigeorgakidis, K. Patroumpas, D. Skoutas, S. Athanasiou, and S. Skiadopoulos. Visual exploration of geolocated time series with hybrid indexing. Big Data Research, 15:12–28, 2019.
- Chatzigeorgakidis, G., Patroumpas, K., Skoutas, D. and Athanasiou, S. A Visual Explorer for Geolocated Time Series. In Proceedings of the 28th International Conference on Advances in Geographic Information Systems. 2020
- 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)

References (chronological order)

- 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)
- Abdullah Al-Mamun, Hao Wu, Walid G. Aref: A Tutorial on Learned Multi-dimensional Indexes. SIGSPATIAL/GIS 2020: 1-4
- 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
- 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

References (chronological order)

- 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
- Qitong Wang, Themis Palpanas: Deep Learning Embeddings for Data Series Similarity Search. KDD 2021
- G. Chatzigeorgakidis, D. Skoutas, K. Patroumpas, T. Palpanas, S. Athanasiou and S. Skiadopoulos: Efficient Range and kNN Twin Subsequence Search in Time Series. TKDE 2022
- Karima Echihabi, Panagiota Fatourou, Kostas Zoumpatianos, Themis Palpanas, Houda Benbrahim: Hercules Against Data Series Similarity Search. PVLDB 2022

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/