High-Dimensional Vector Similarity Search

Themis Palpanas

Université Paris Cité
French University Institute

LIPADE
Laboratoire d'Informatique PAris DEscartes

diiP Summer School on Data Science (dSDS) – Paris (France), June 2024

diNo

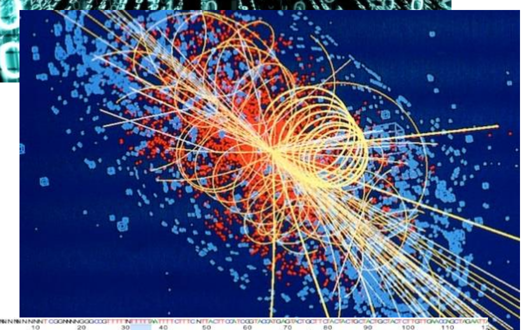
Acknowledgements

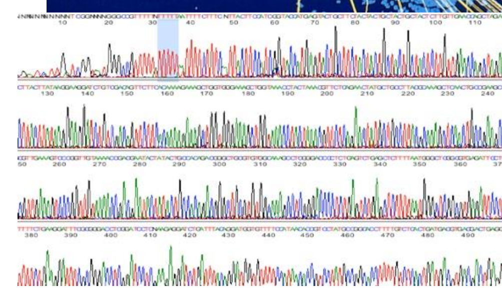
- thanks to all my collaborators from:
  - Harvard University
  - Cornell University/Microsoft
  - University of Chicago
  - University of Toronto
  - Ohio State University
  - Inria
  - University of California at Riverside
  - University of Crete/FORTH
  - University of Trento

  - EDF
  - CEA
  - Huawei

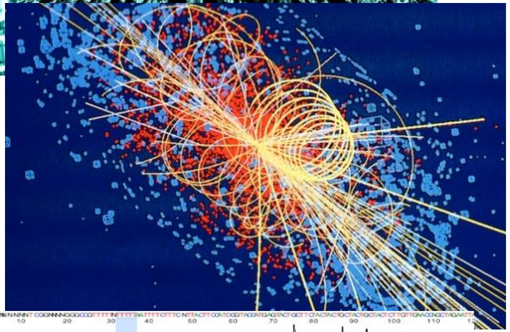
Executive Summary

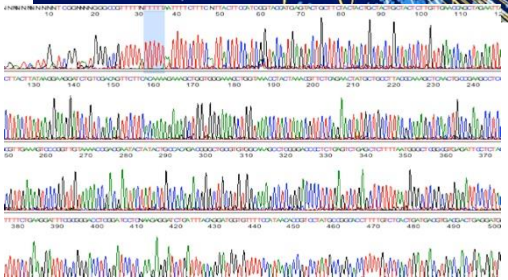
- data collected at unprecedented rates
- they enable data-driven scientific discovery
- lots of these data are high-d vectors
  - takes **days-weeks** to analyze big high-d vector collections





Executive Summary





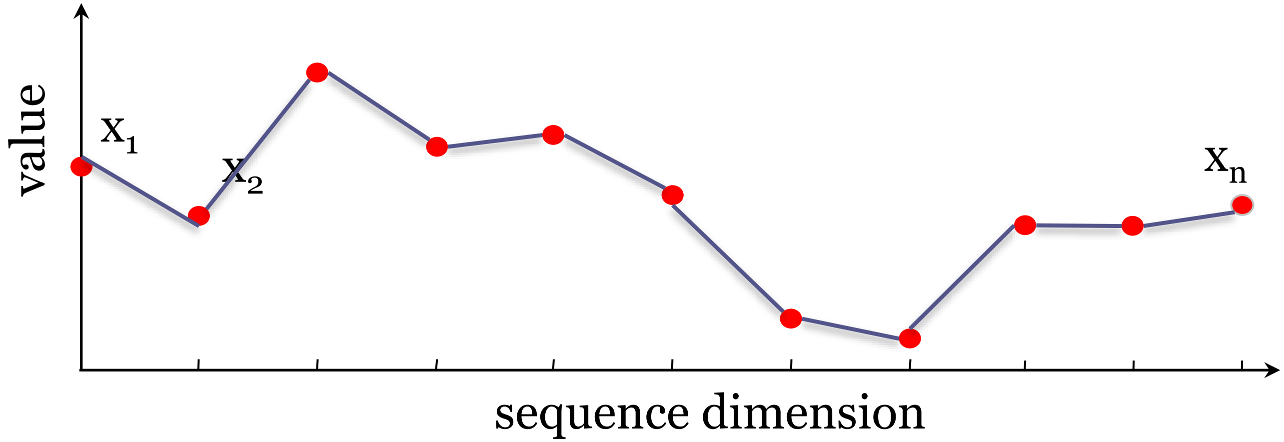
- data collected at unprecedented rates
- they enable data-driven scientific discovery
- lots of these data are high-d vectors
  - takes **days-weeks** to analyze big high-d vector collections

goal: analyze big high-d vectors in **seconds**

Data Series

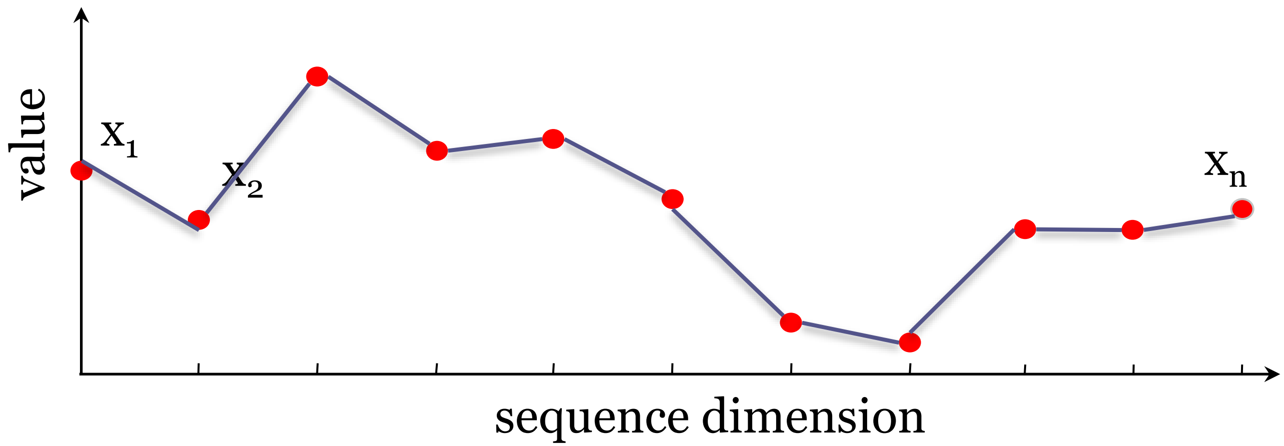
Data Series

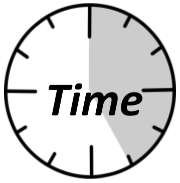
- Sequence of points ordered along some dimension



Data Series

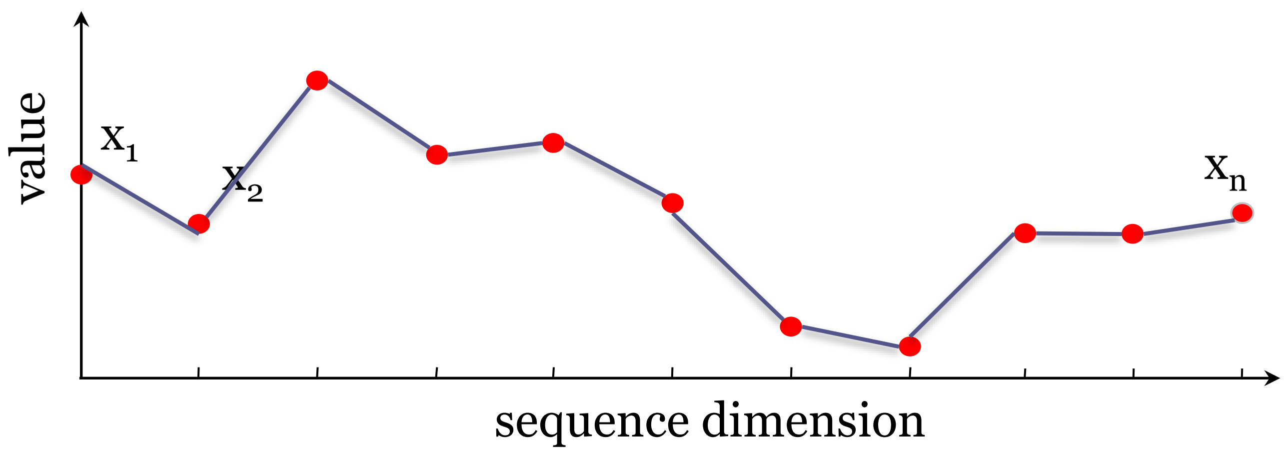
- Sequence of points ordered along some dimension

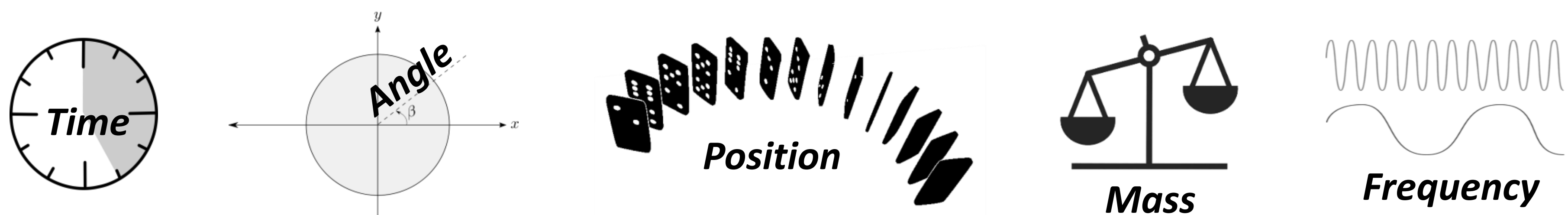




Data Series

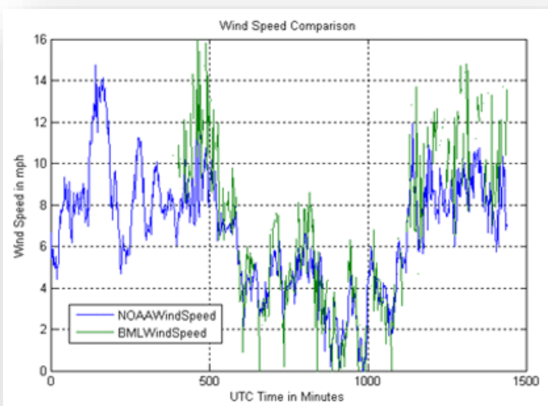
- Sequence of points ordered along some dimension





Scientific Monitoring

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



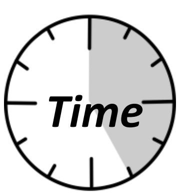
Wind speed

From ocean observing node project

http://bml.ucdavis.edu/boon/wind.html

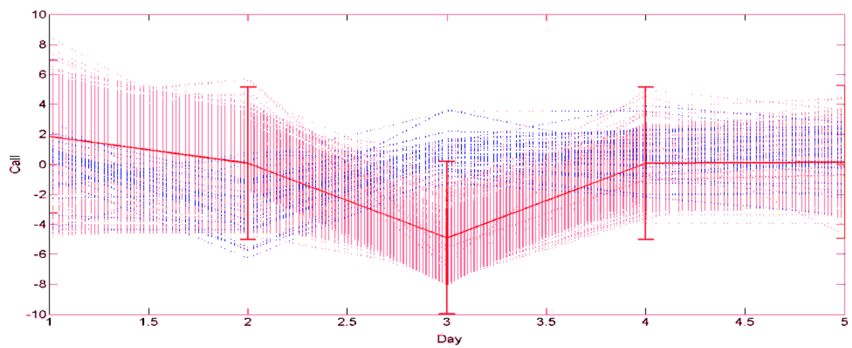
Historical stock quotes

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

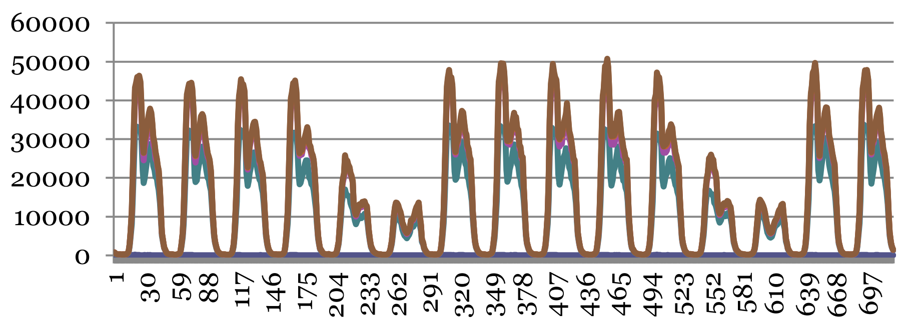


Telecommunications

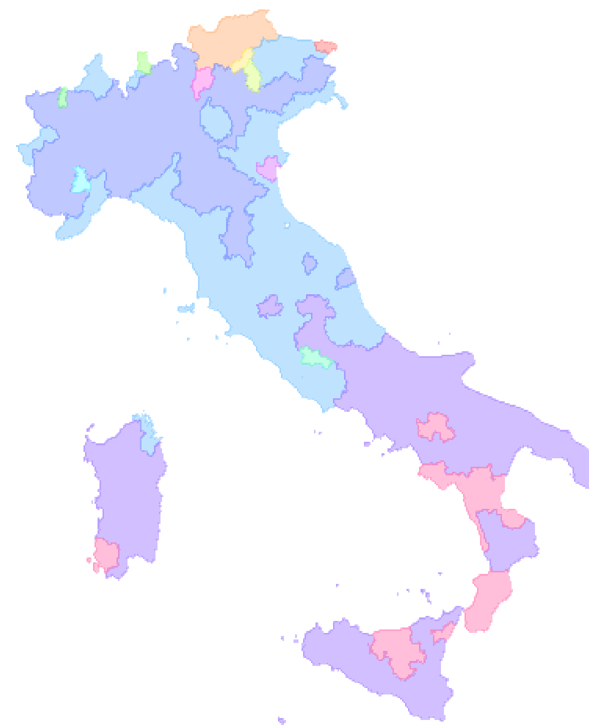
- analysis of call activity patterns
  - Telecom Italia



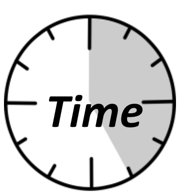
call activity for Easter Monday



average number of calls for 5 smallest clusters

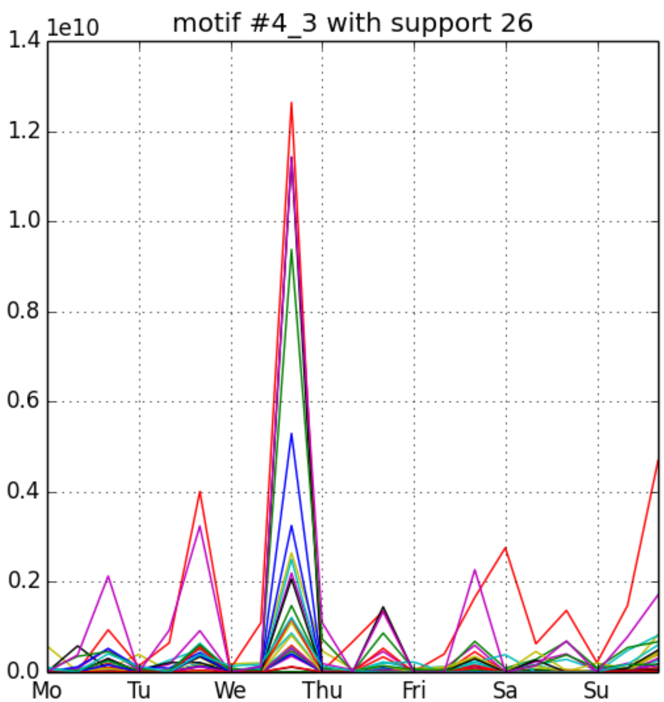


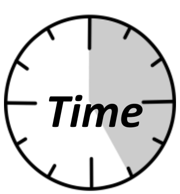
clustermap of incoming calls time series



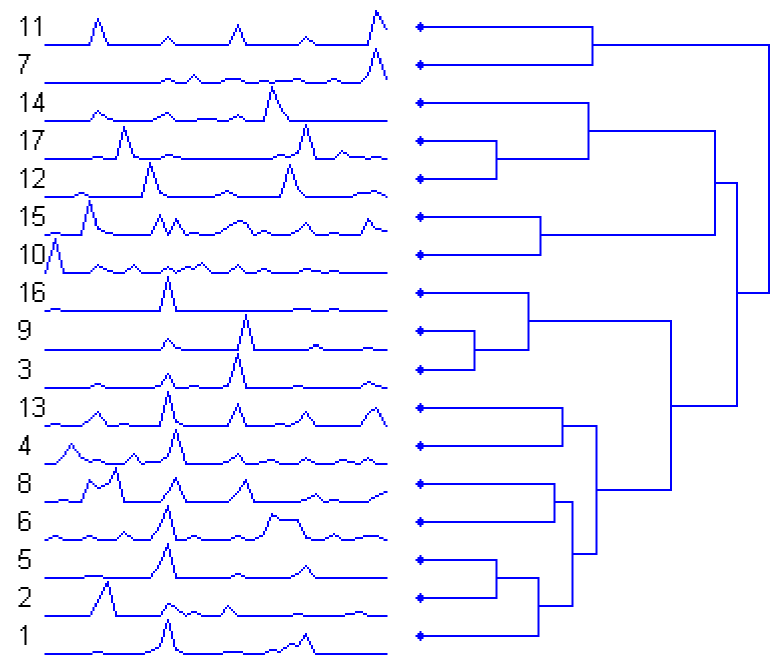
Home Networks

- temporal usage behavior analysis of home networks
  - Portugal Telecom





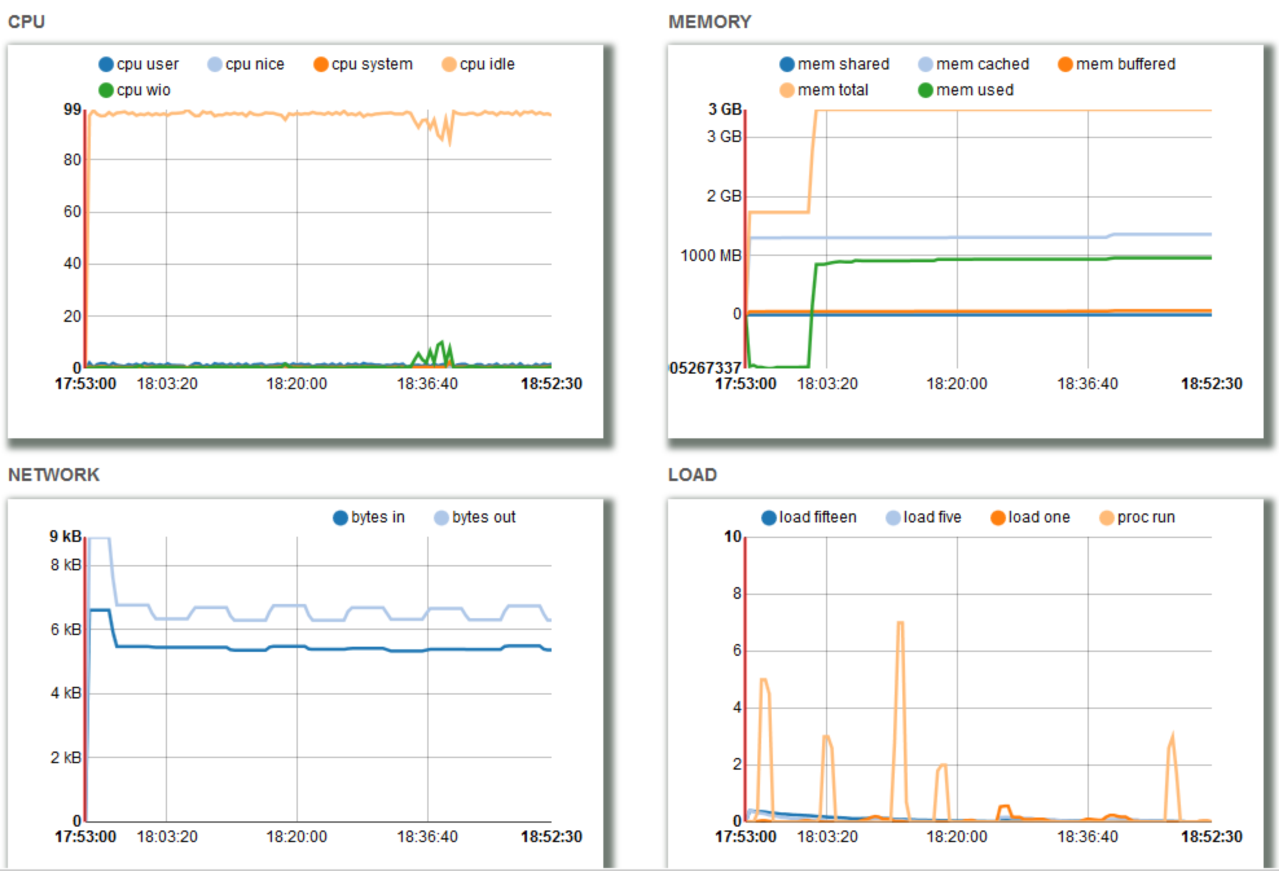
(previously unknown) frequent behavior pattern

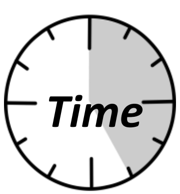


clustering based on user activity patterns

Data Centers

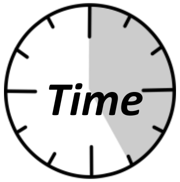
- cloud utilization/operation/health monitoring





Neuroscience

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

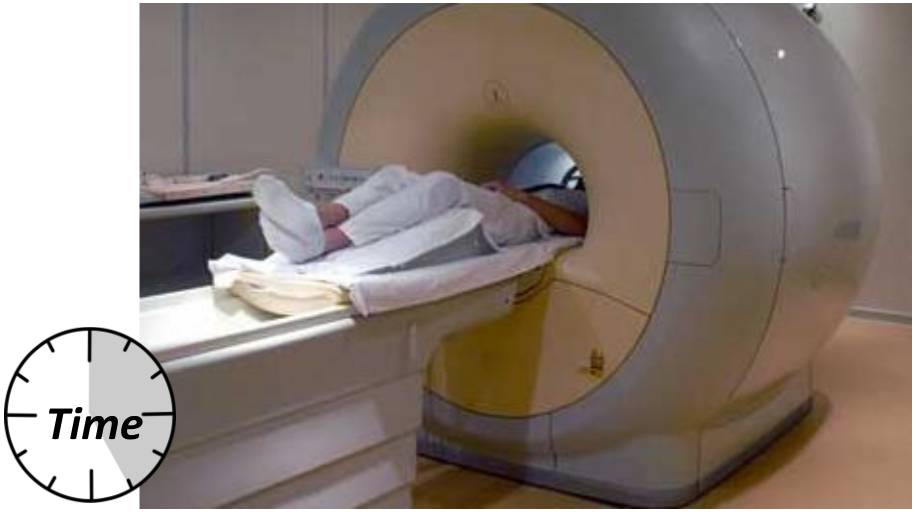


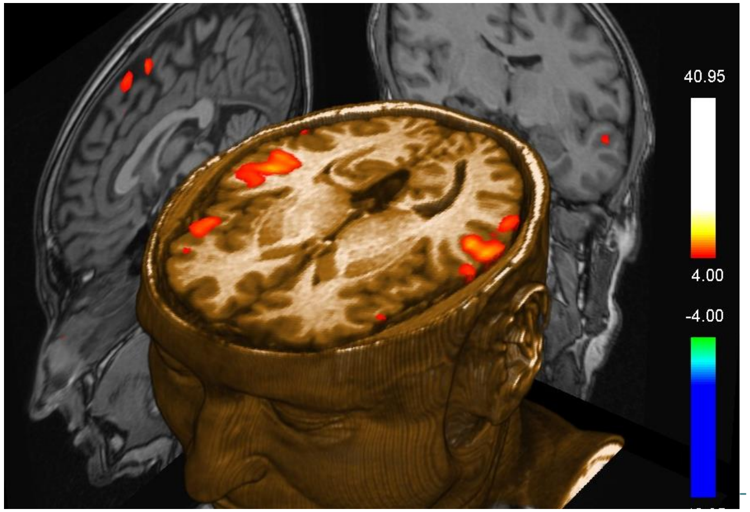
Neuroscience

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

Neuroscience

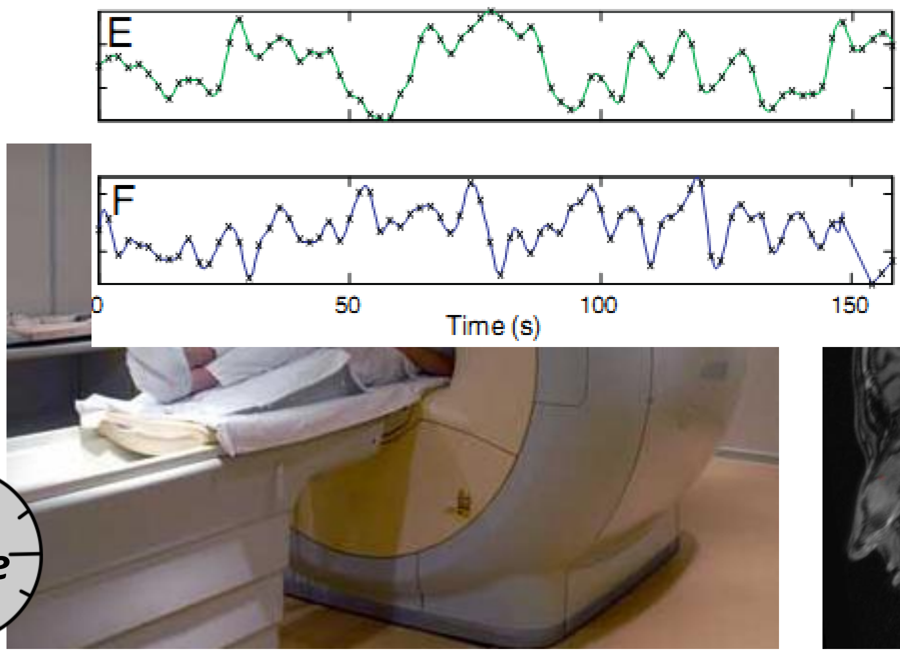
- functional Magnetic Resonance Imaging (fMRI) data
  - primary experimental tool of neuroscientists
  - reveal how different parts of brain respond to stimuli

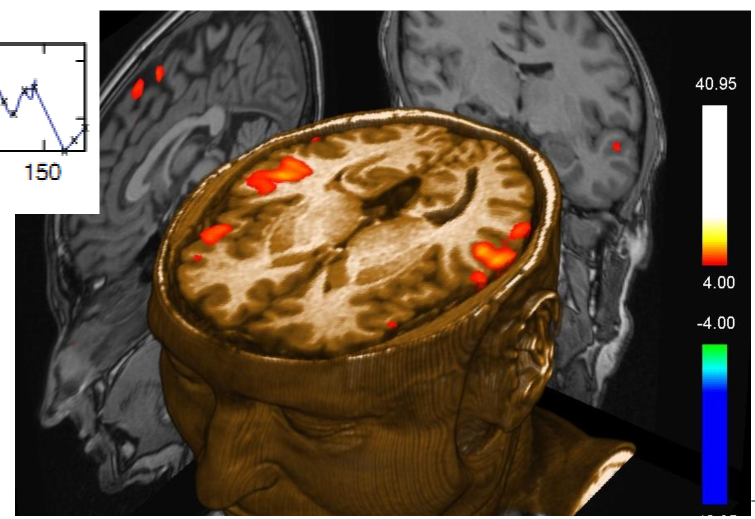


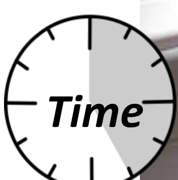


Neuroscience

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

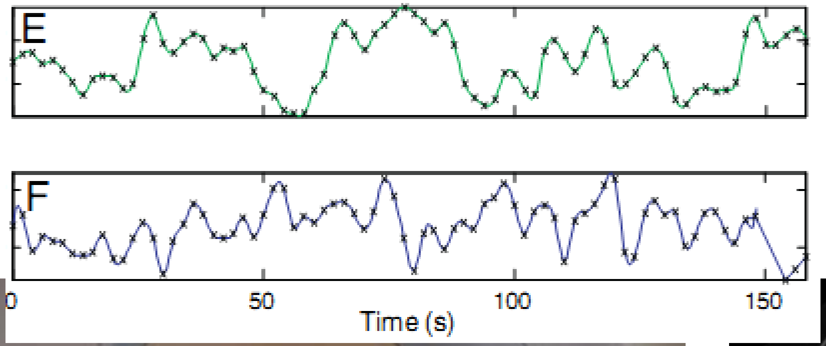


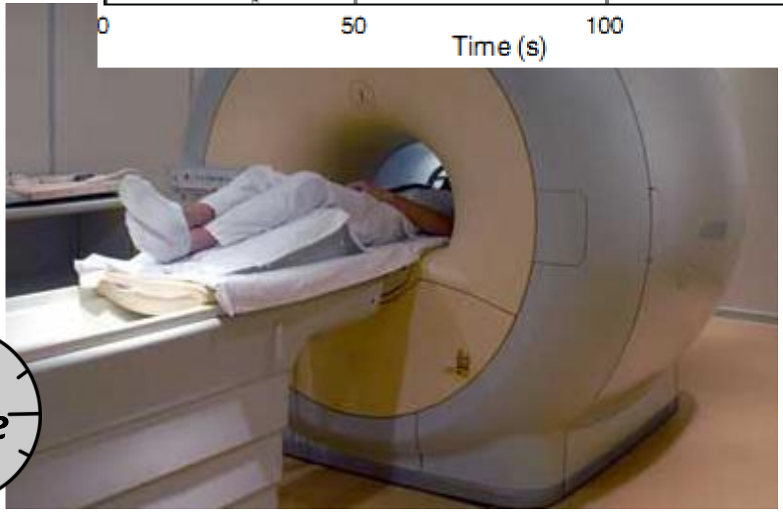


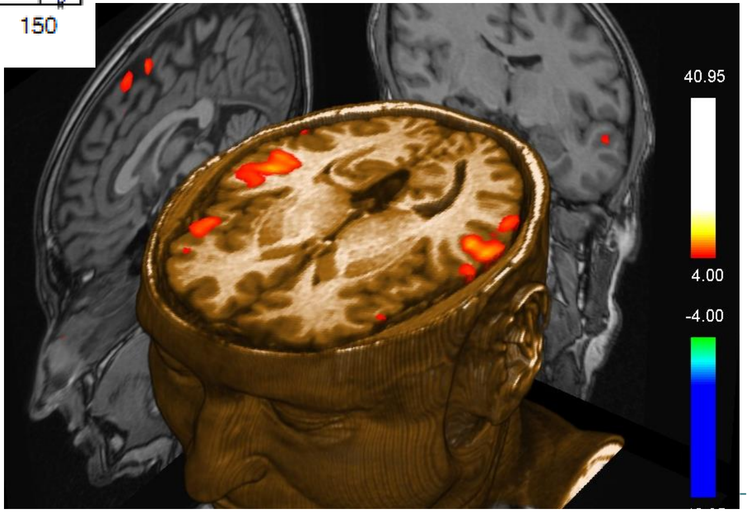


Neuroscience

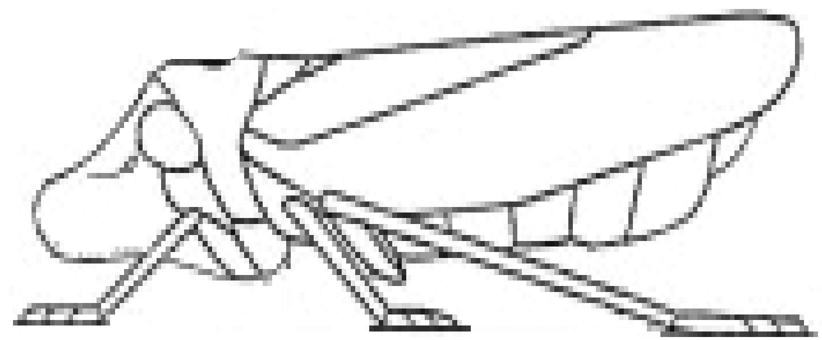
- functional Magnetic Resonance Imaging (fMRI) data
  - primary experimental tool of neuroscientists
  - reveal how different parts of brain respond to stimuli

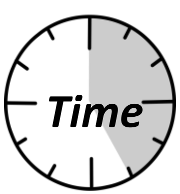




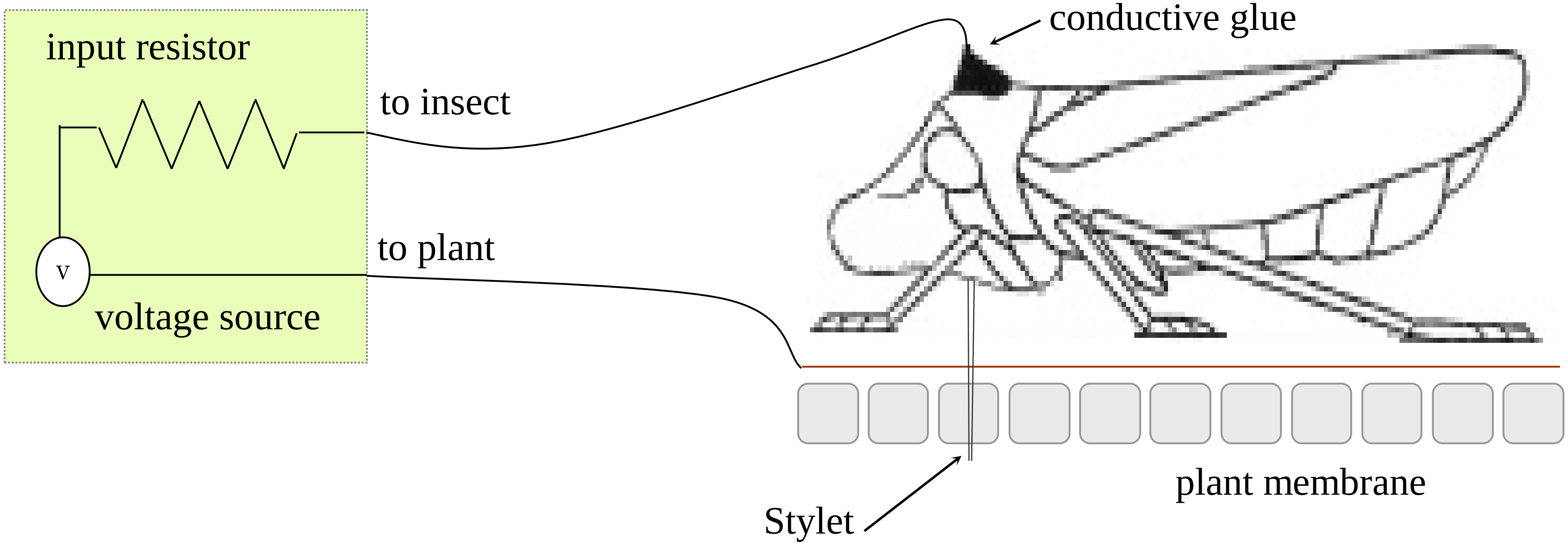


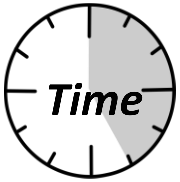
Entomology



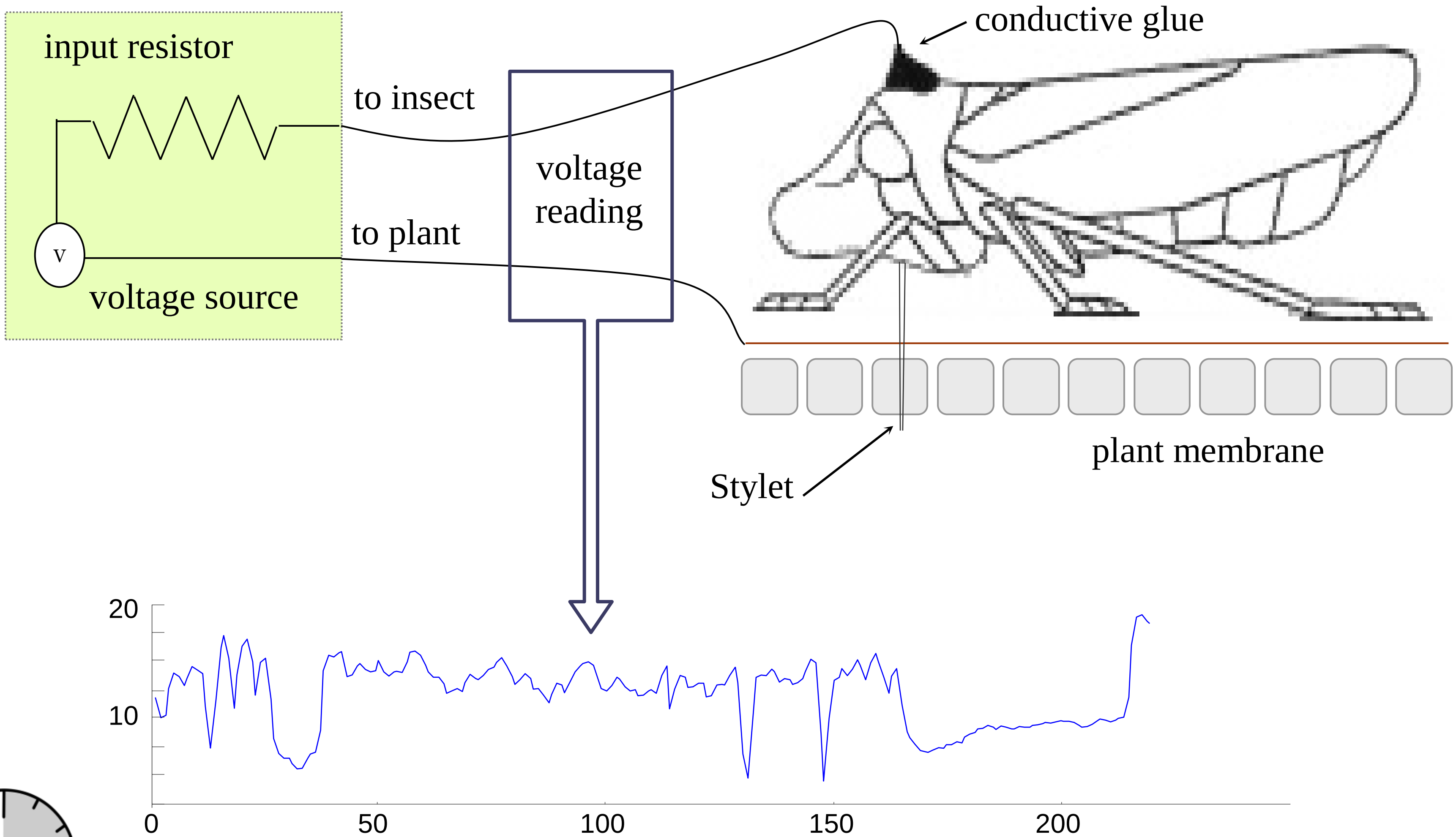


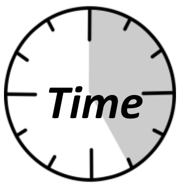
Entomology


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


Time

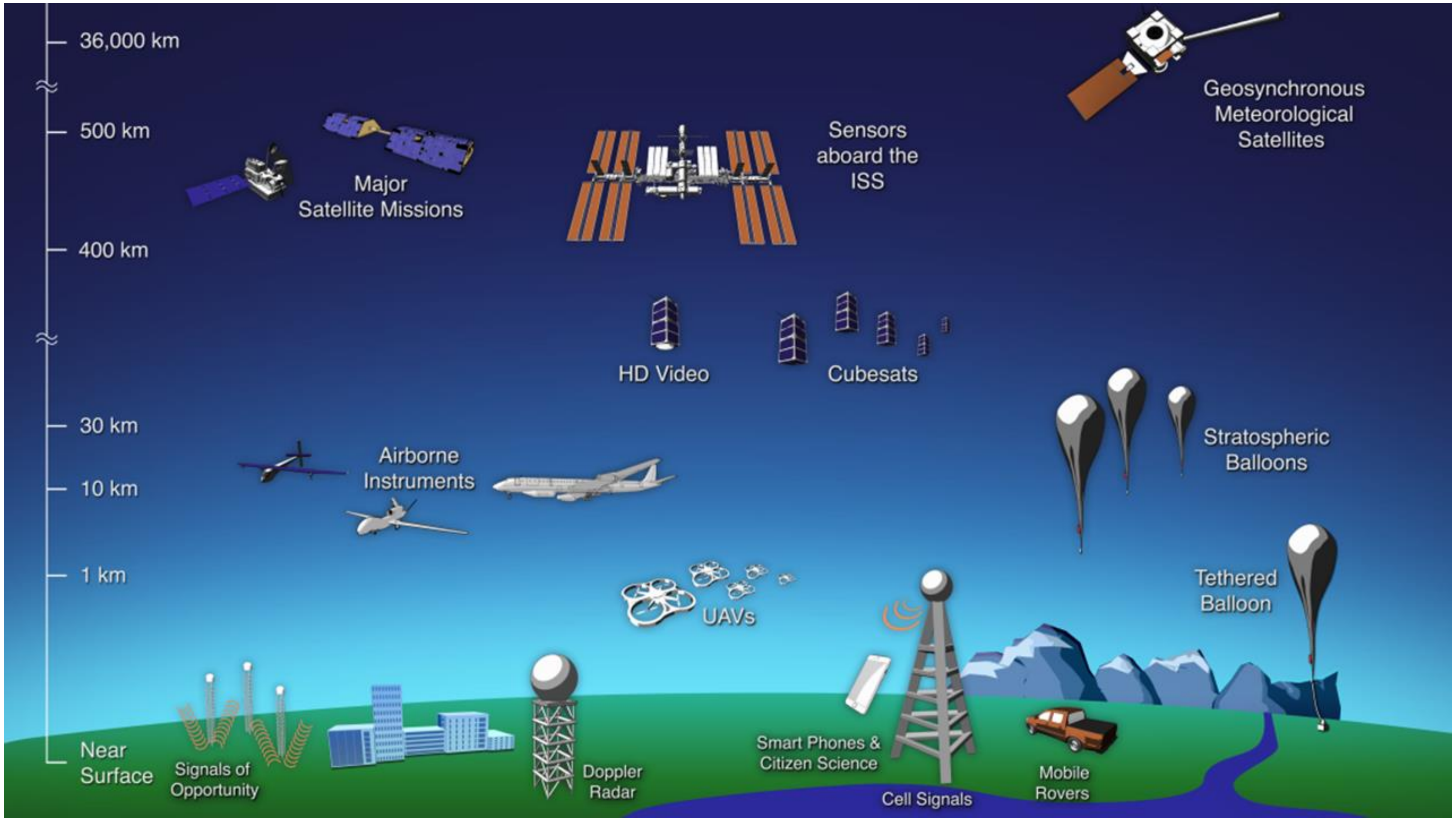
Entomology

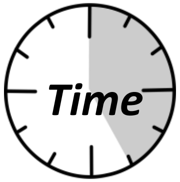

input resistor
to insect
to plant
v
voltage source
voltage reading
conductive glue
Stylet
plant membrane
20
10
0
50
100
150
200


Time

Remote Sensing

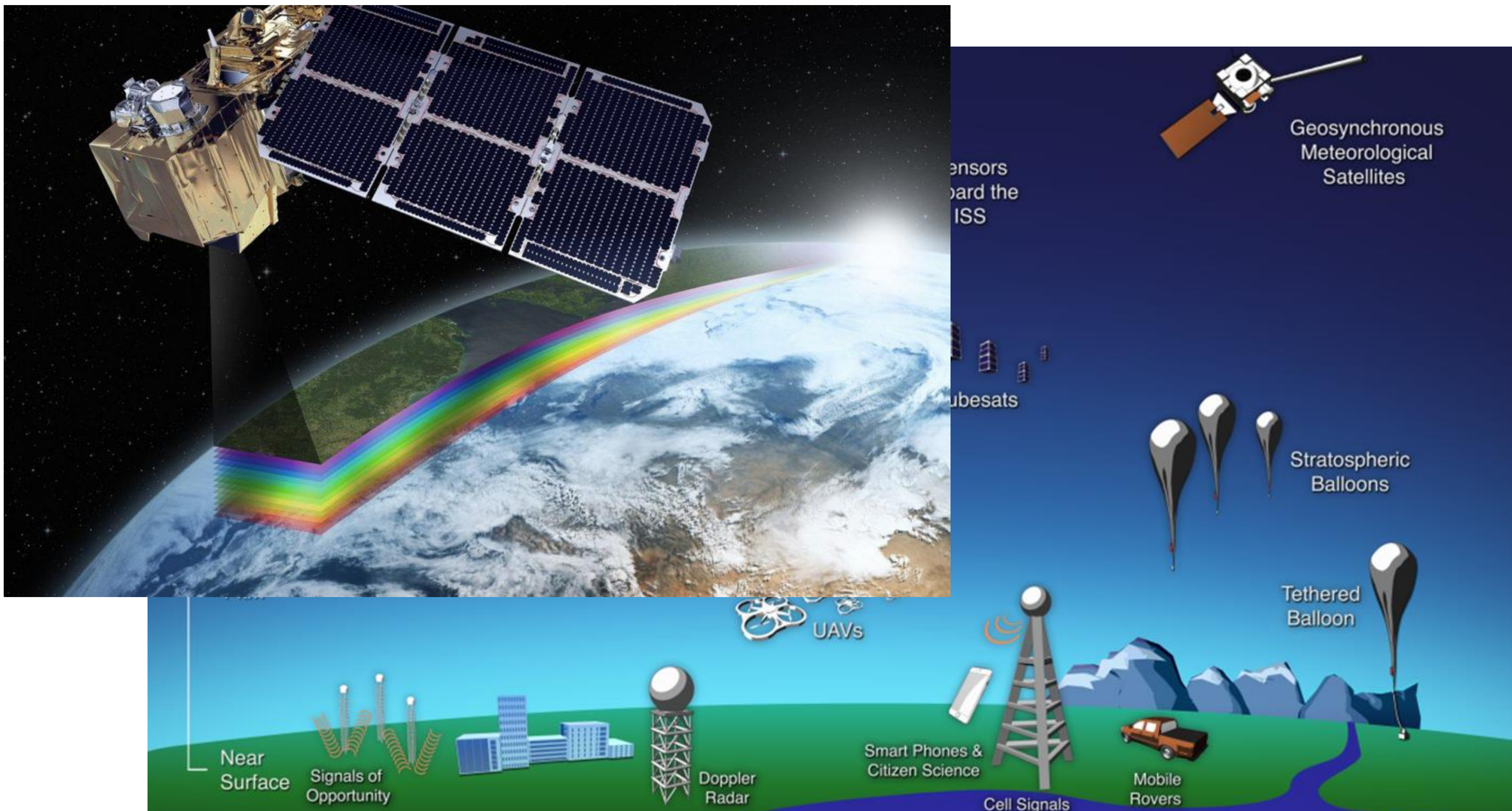
- Earth monitoring

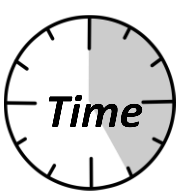




Remote Sensing

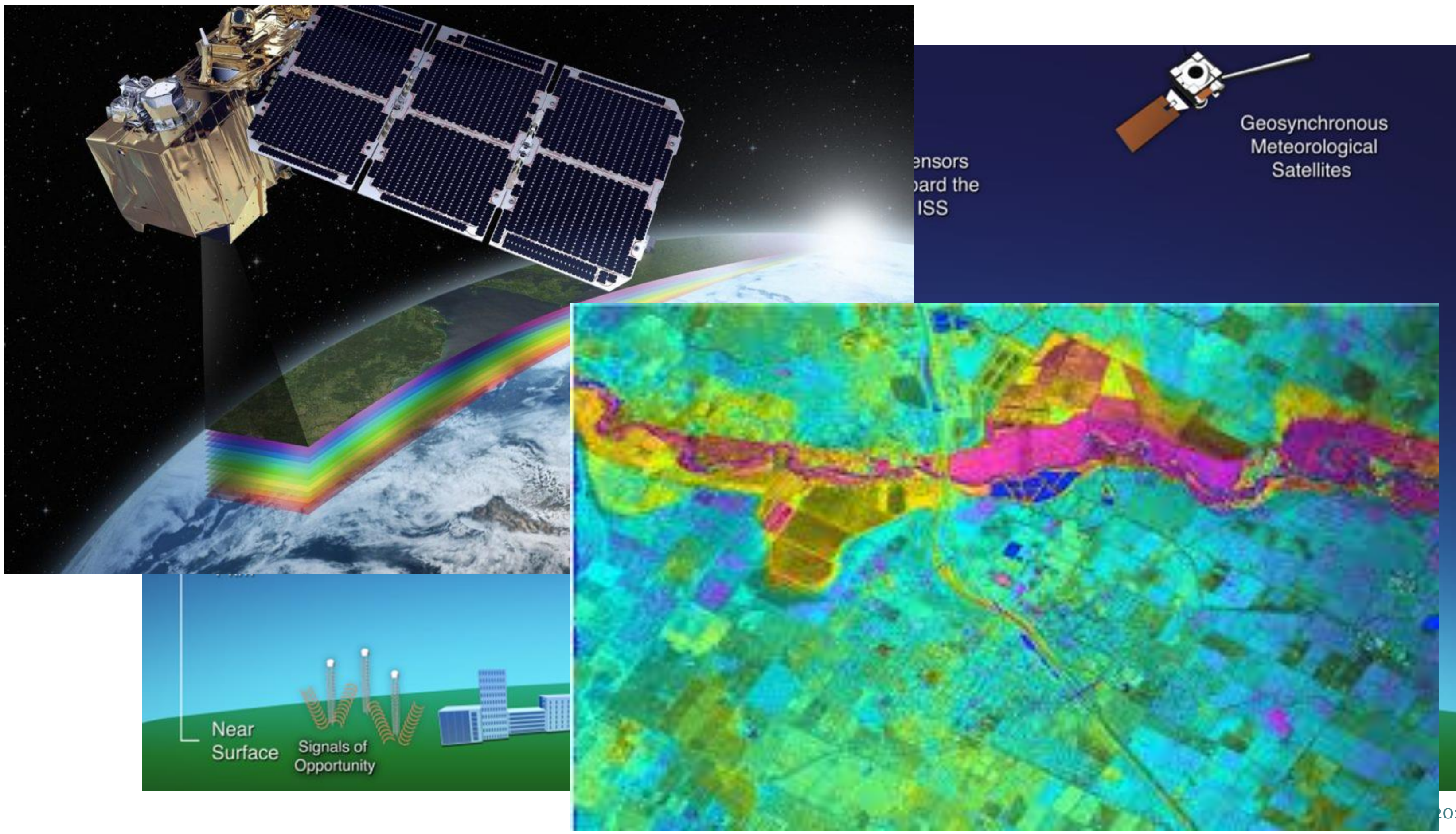
- Earth monitoring

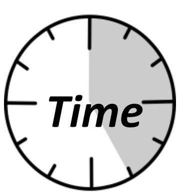




Remote Sensing

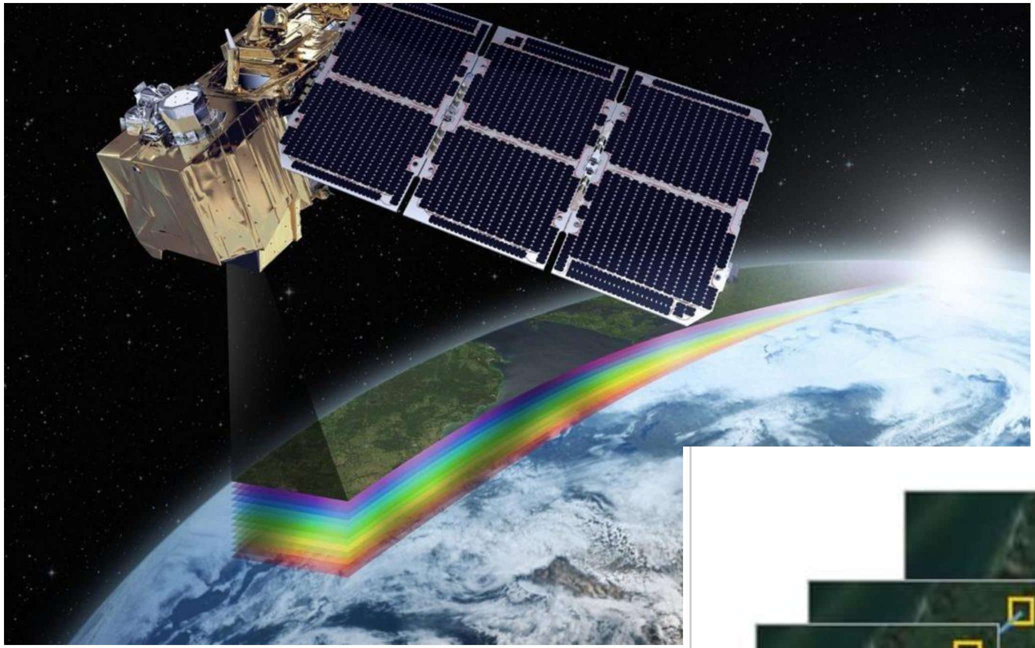
- Earth monitoring

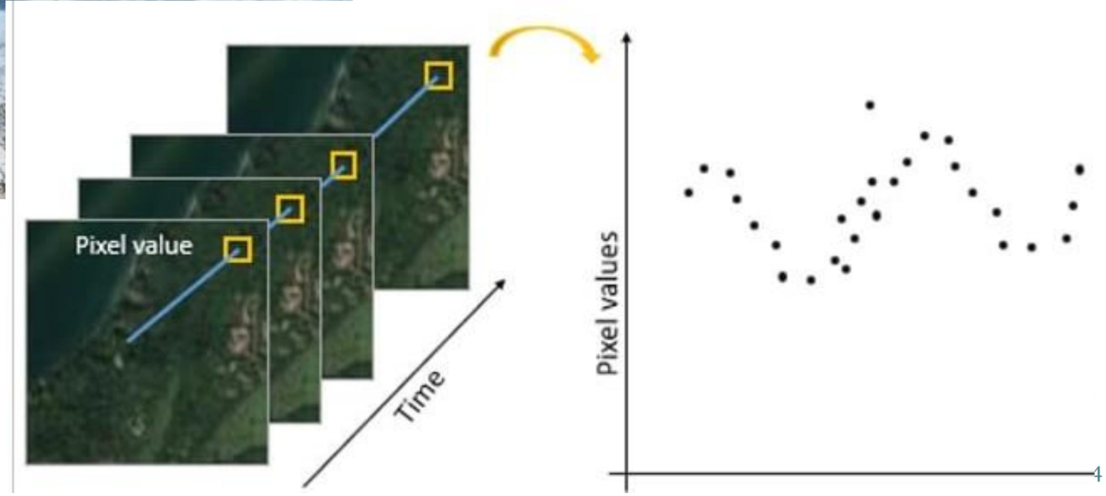


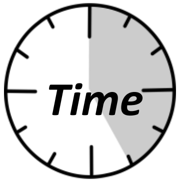


Remote Sensing

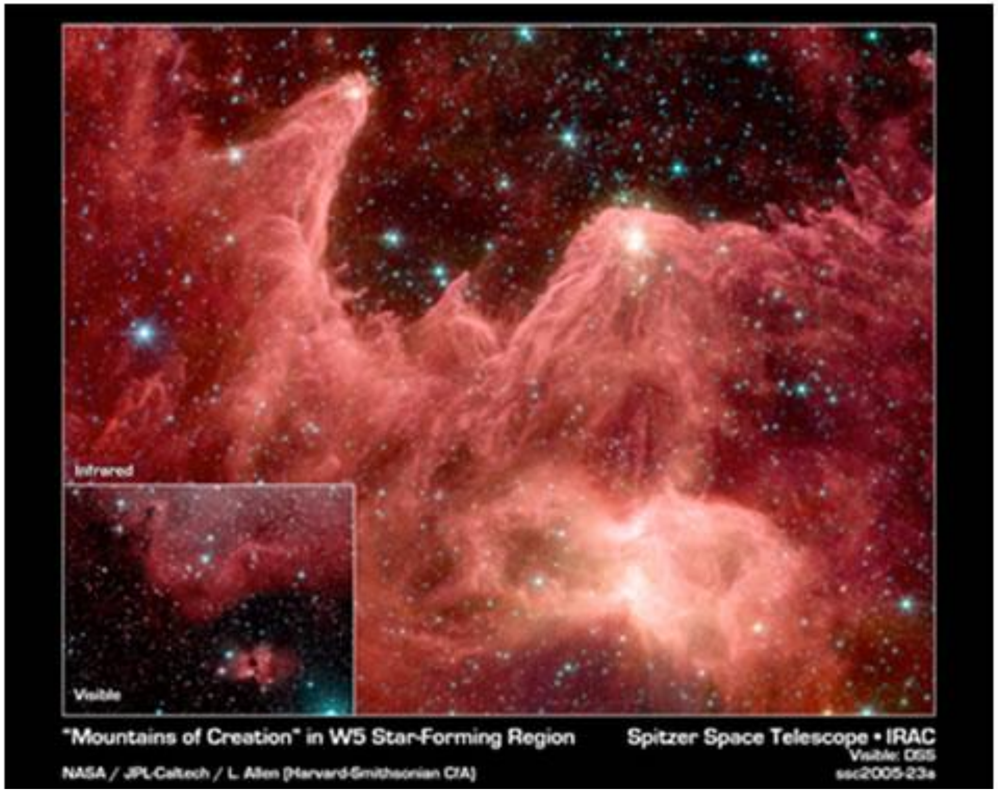
- Earth monitoring

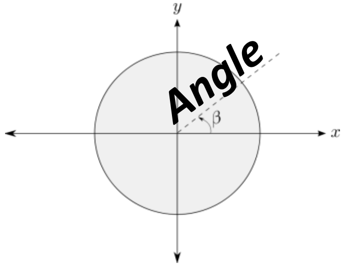




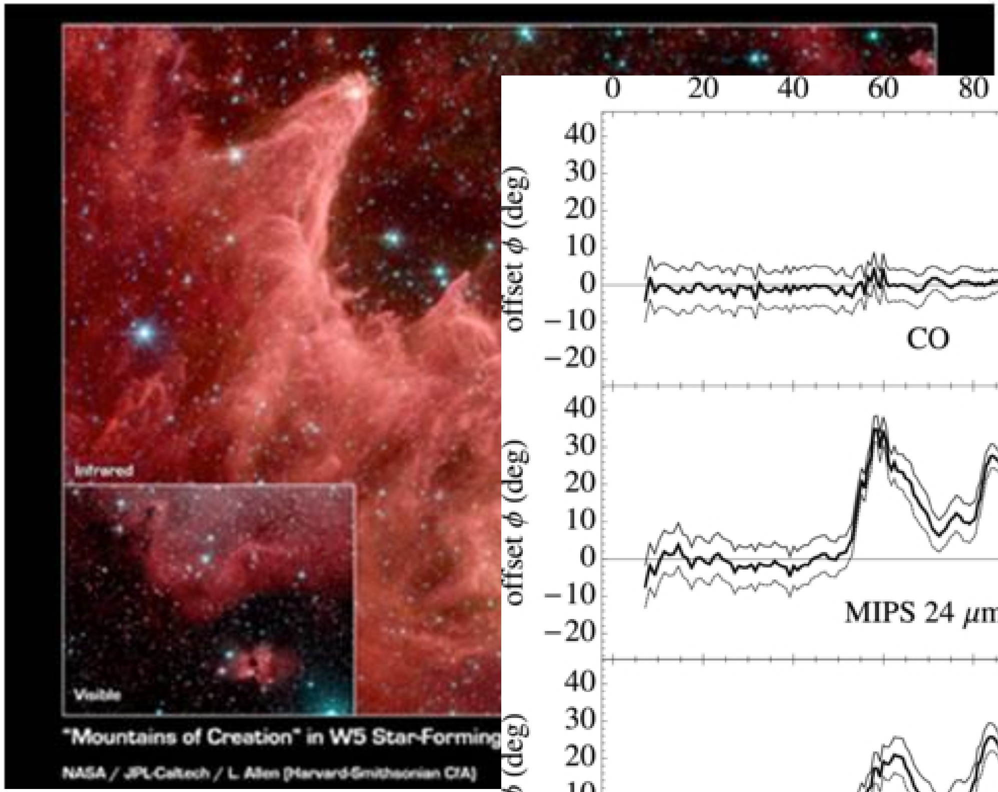


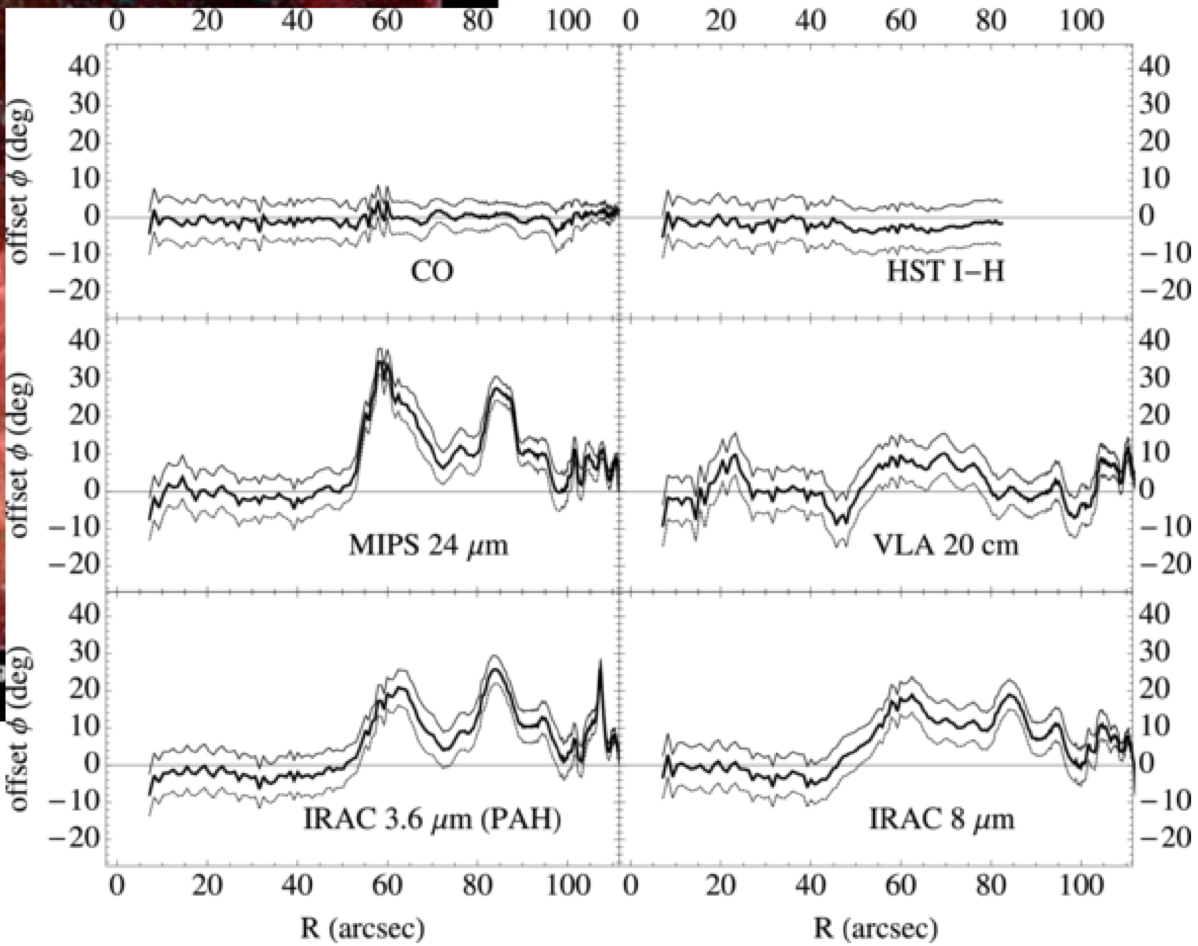
Astrophysics

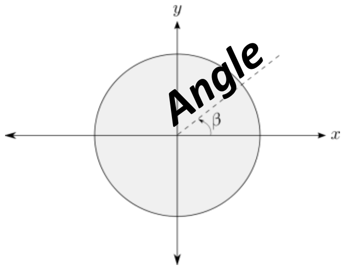




Astrophysics

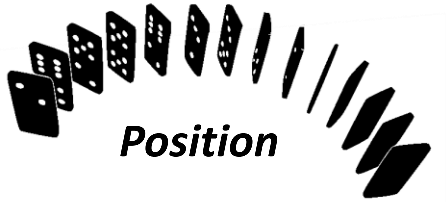




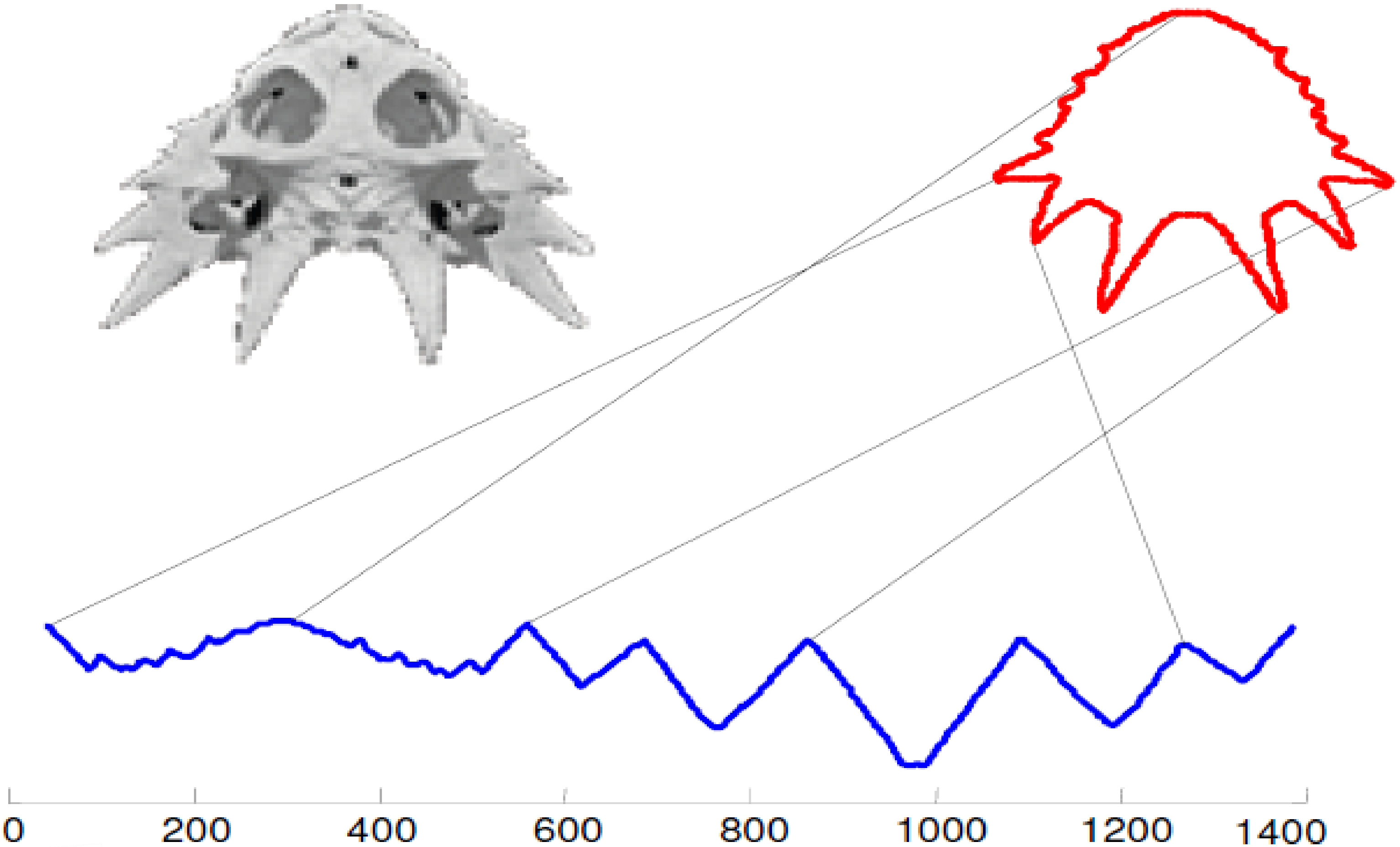


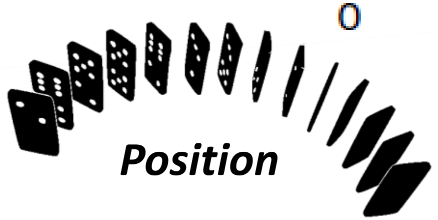
Schinnerer et al.

Paleontology



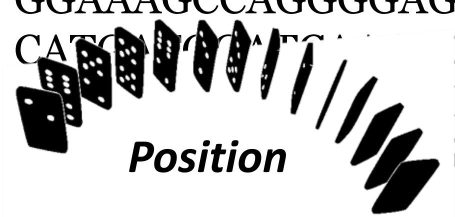
Paleontology





Biology

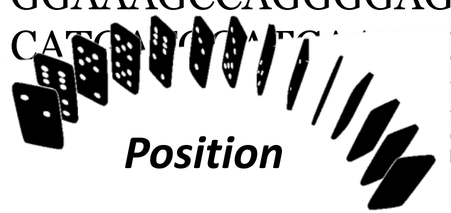
GTCAATGGCCAGGATATTAGAACAGTACTCTGTGAACCCTATTTATGGTGGCACCCCTTAGACTAA
GATAACACAGGGAGCAAGAGGTTGACAGGAAAGCCAGGGGAGCAGGGAAGCCTCCTGTAAAGAG
AGAAGTGCTAAGTCTCCTTTCTAAGGCACATGATGGATTCAAGGGAAAGCCACATTTGACTAAAGC
CCAAGGATTGTTGCTTCTAATCCGATTTCTTGGCAGAAGATATTACAAACTAAGAGTCAGATTAA
TATGTGGGTGCCAAAATAAATAAACAAATAATTGAATAATCCCTGGAGGTTTAAGTGAGGAGAAA
CTCCTCCACAGCTTGCTACCGAGGCAGAACCGGTTGAAACTGAAATGCATCCGCCGCCAGAGGATC
TGTAAAAGAGAGGTTGTTACGAAACTGGCAACTGCCAACCAAAGTCCACCAATGGACAAGCAAAA
AAGAGCACTCATCTCATGCTCCCAAGGATCAACCTTCCCAGAGTTTTCACTTAAGTGGCCACCAAG
CCAGTTGTCAATCCAGGGCTTTGGACTGAAATCTAGGGCTTCATCCGCTACCTCAGAGTGTCTTCT
ATTTCTTCCAGCCAGTGACAAATACAACAAACATCTGAGATGTTTTAGCTATAAATCCTTTACAATT
GTTATTTATGTCTTAACTTTTGTTATACCTGGAAAAGTAGGGGAAACAATAAGAACATACTGTCTT
GGCCAAGCATCCAAGGTTAAATGAGTTATGGAAATTCATTTGGGAGCCAAGACATTGCACGTGGT
TATTTATTAGTCACCCAAGCATGTATTTTGCATGTCCATCAGTTGTTCTTGGCCAAAAGAGCAGAAT
CAATGAGCCGCTGCAGATGCAGACATAGCAGCCCCTTGCAGGGACAAGTCTGCAAGATGAGCATT
GAAGAGGATGCACAAGCCCGGTAGCCCGGGAAATGGCAGGCACTTACAAGAGCCCAGGTTGTTGC
CATGTTTGTTTTTGCAACTTGTCTATTTAAAGAGATTTGGGCAATGGCCAGGATATTAGAACAGTA
CTCTGTGAACCCTATTTATGGTAGCACCCCTTAGACTAAGATAACACAGGGAGCAAGAGGTTGACA
GGAAAGCCAGGGGAGCAGGGAAGCCTCCTGTAAAGAGAGAAGTGCTAAGTCTCCTTTCTAAGGCA
CA GAAAGTCACATTTGACTAAAGCCCAAGGATTGTTGCTTCTAATCCGATTC
GCAAACTAAGAGTCAGATTAATATGTGGGTGCCAAAATAAATAAACAAATA
AGGTTTAAGTGAGGAGAAACTCCTCCACACTTGCTACCGAGGCAGAACCG



Biology

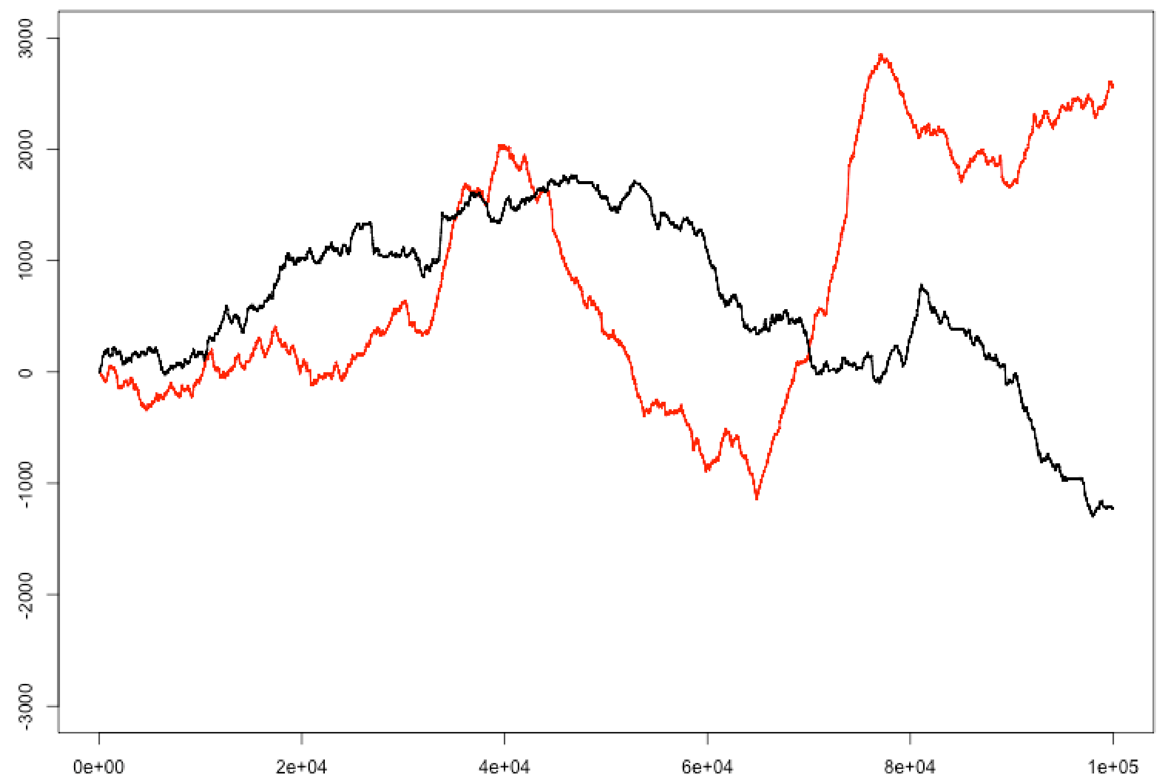
GTCAATGGCCAGGATATTAGAACAGTACTCTGTGAACCCTATTTATGGTGGCACCCCTTAGACTAA
GATAACACAGG CCTGTAAAGAG
AGAAGTGCTAA TTGACTAAAGC
CCAAGGATTG AGTCAGATTAA
TATGTGGGTGC TGAGGAGAAA
CTCCTCCACAG GCCAGAGGATC
TGTAAAAGAGA ACAAGCAAAA
AAGAGCACTCA GGCCACCAAG
CCAGTTGTCAAT GAGTGTCTTCT
ATTTCTTCCAGC CCTTTACAATT
GTTATTTATGTC CATACTGTCTT
GGCCAAGCATC TGCACGTGGT
TATTTATTAGTC AAGAGCAGAAT
CAATGAGCCGC AGATGAGCATT
GAAGAGGATGC CAGGTTGTTGC
CATGTTTGTTTT TTAGAACAGTA
CTCTGTGAACC AGAGGTTGACA

GGAAAGCCAGGGGAGCAGGGAAGCCTCCTGTAAAGAGAGAAGTGCTAAGTCTCCTTTCTAAGGCA
GAAAGTCACATTTGACTAAAGCCCAAGGATTGTTGCTTCTAATCCGATTC
GCAAACTAAGAGTCAGATTAATATGTGGGTGCCAAAATAAATAACAAATA
AGGTTTAAGTGAGGAGAAACTCCTCCACACTTGCTACCGAGGCAGAACCG



Biology

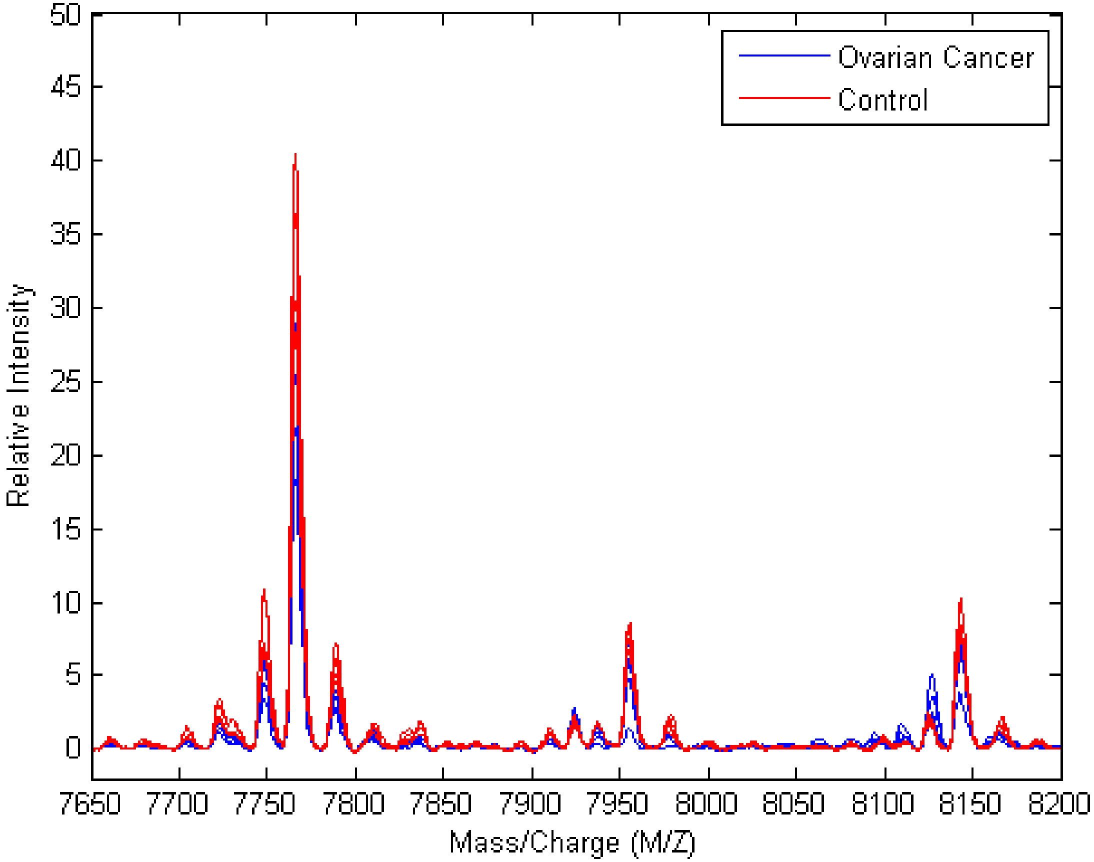
GTCAATGGCCAGGATATTAGAACAGTACTCTGTGAACCCTATTTATGGTGGCACCCCTTAGACTAA



GATAACACAGG CCTGTAAAGAG
AGAAGTGCTAA TTGACTAAAGC
CCAAGGATTG AGTCAGATTAA
TATGTGGGTGC TGAGGAGAAA
CTCCTCCACAG GCCAGAGGATC
TGTAAAAGAGA ACAAGCAAAA
AAGAGCACTCA GGCCACCAAG
CCAGTTGTCAAT GAGTGTCTTCT
ATTTCTTCCAGC CCTTTACAATT
GTTATTTATGTC CATACTGTCTT
GGCCAAGCATC TGCACGTGGT
TATTTATTAGTC AAGAGCAGAAT
CAATGAGCCGC AGATGAGCATT
GAAGAGGATGC CAGGTTGTTGC
CATGTTTGTTTT TTAGAACAGTA
CTCTGTGAACC AGAGGTTGACA
GGAAAGCCAGGGGAGCAGGGAAGCCTCCTGTAAAGAGAGAAGTGCTAAGTCTCCTTTCTAAGGCA

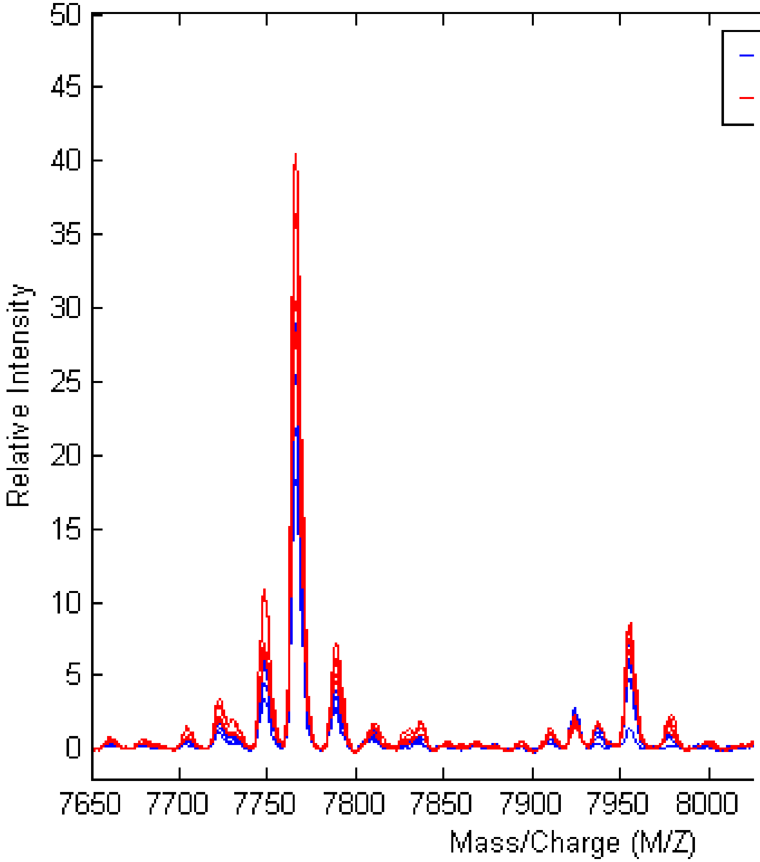
GAAAGTCACATTTGACTAAAGCCCAAGGATTGTTGCTTCTAATCCGATTC
GCAAACTAAGAGTCAGATTAATATGTGGGTGCCAAAATAAATAAACAAATA
AGGTTTAAGTGAGGAGAAACTCCTCCACACTTGCTACCGAGGCAGAACCG

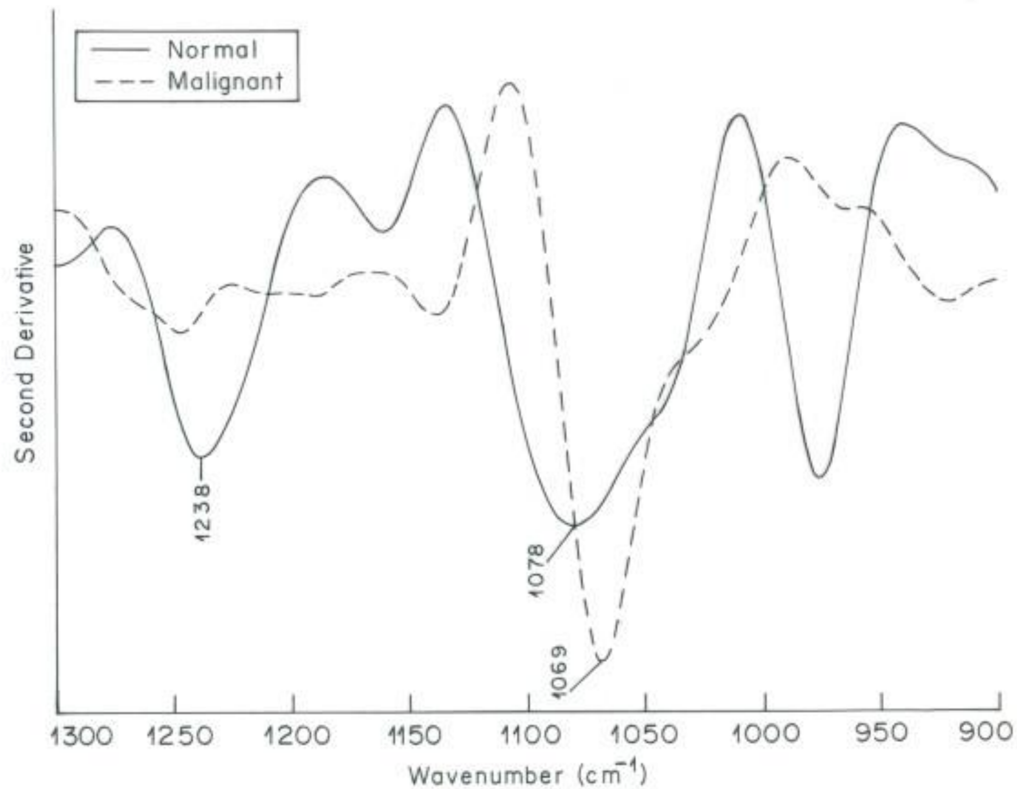
Medicine



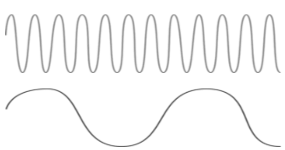
Mass

Medicine





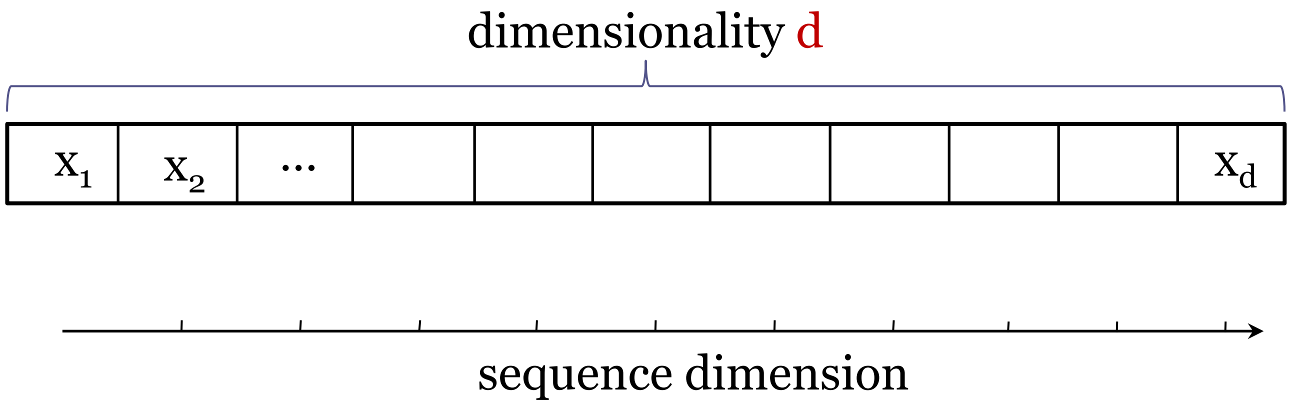
Mass



Frequency

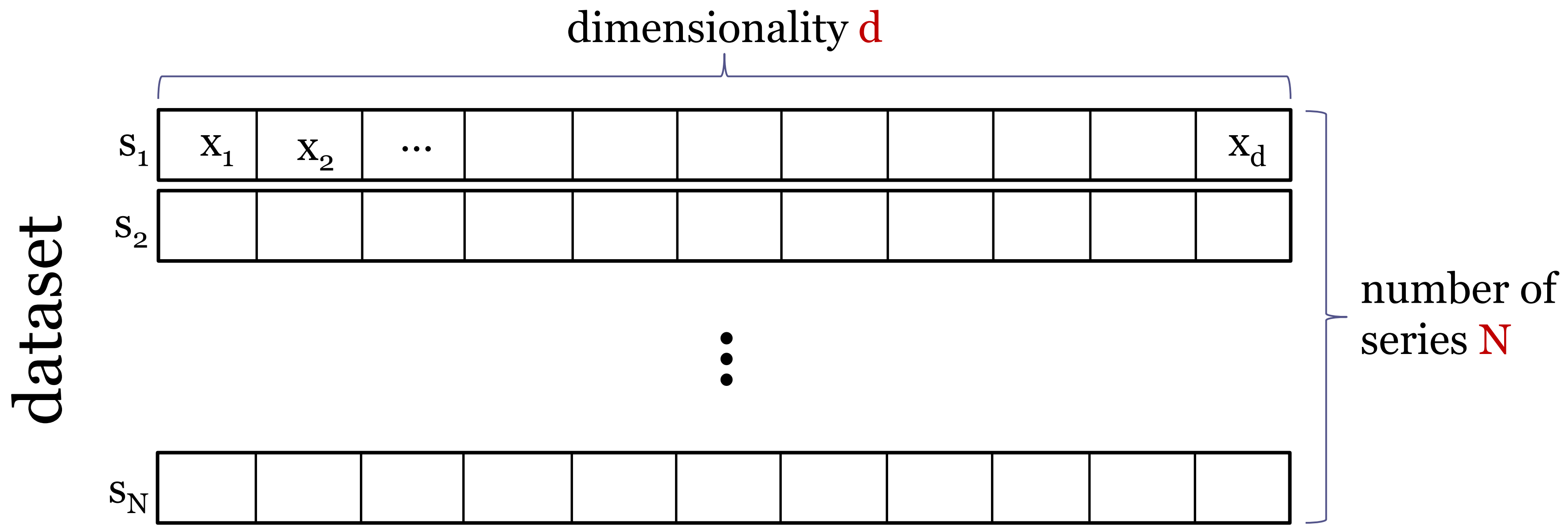
Data Series

- represented as *d*-dimensional vector



Data Series Collections

- represented as *N d*-dimensional vectors



What do we want to do with them?
- simple query answering

select values in time interval

select values in some range

select some data series

combinations of those

What do we want to do with them? - simple query answering

- a solved(?) problem
  - your favorite DBMS
  - …
  - InfluxData
  - kx
  - Riak TS
  - OpenTSDB
  - Gorilla/Beringei
  - TimescaleDB
  - KairosDB
  - Druid
  - …

What do we want to do with them? - complex analytics

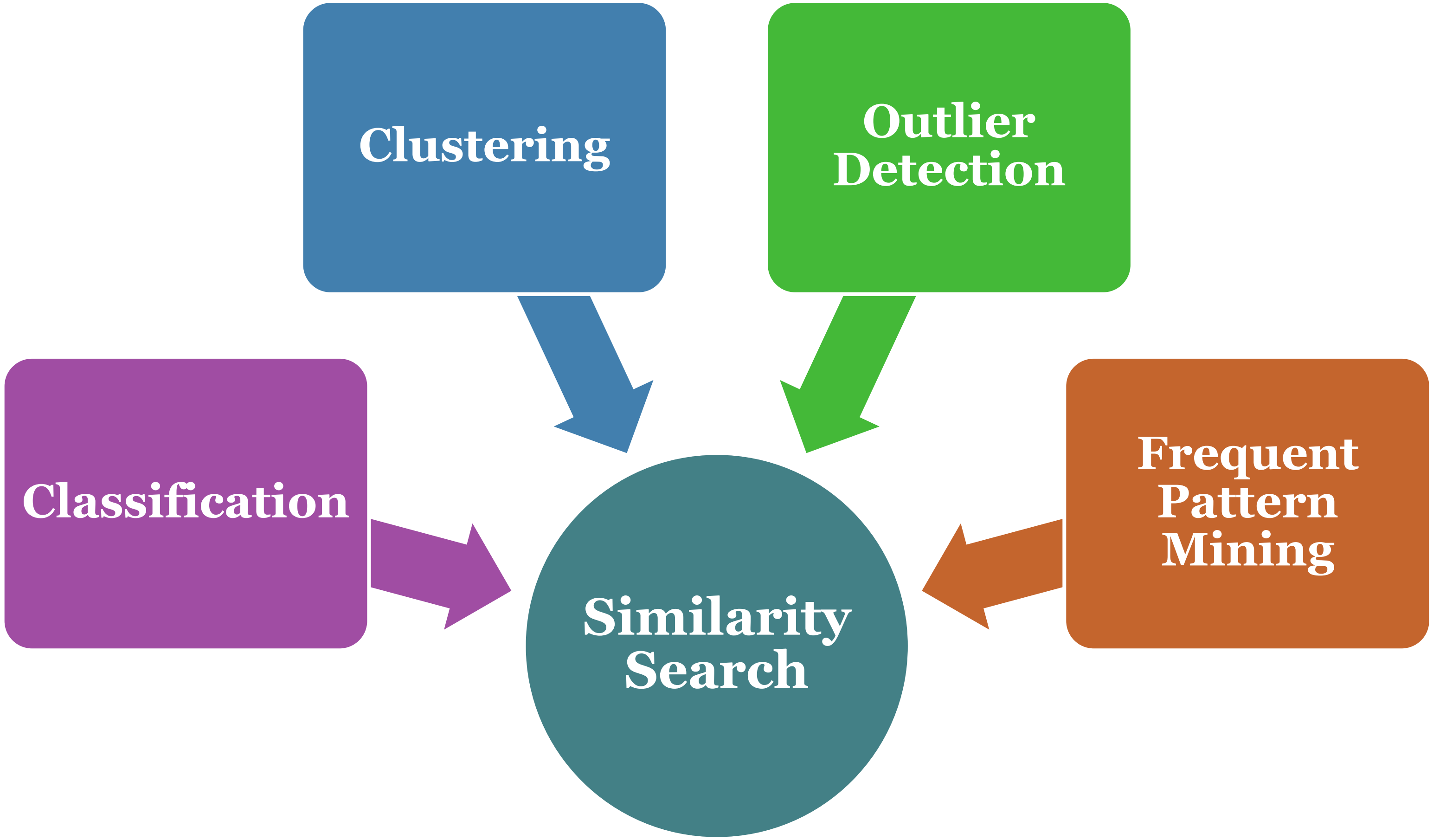
Clustering

Outlier Detection

Classification

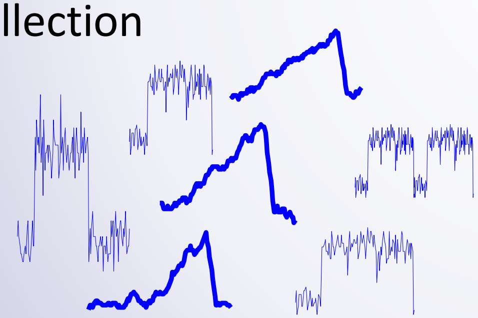
Frequent Pattern Mining

What do we want to do with them? - complex analytics

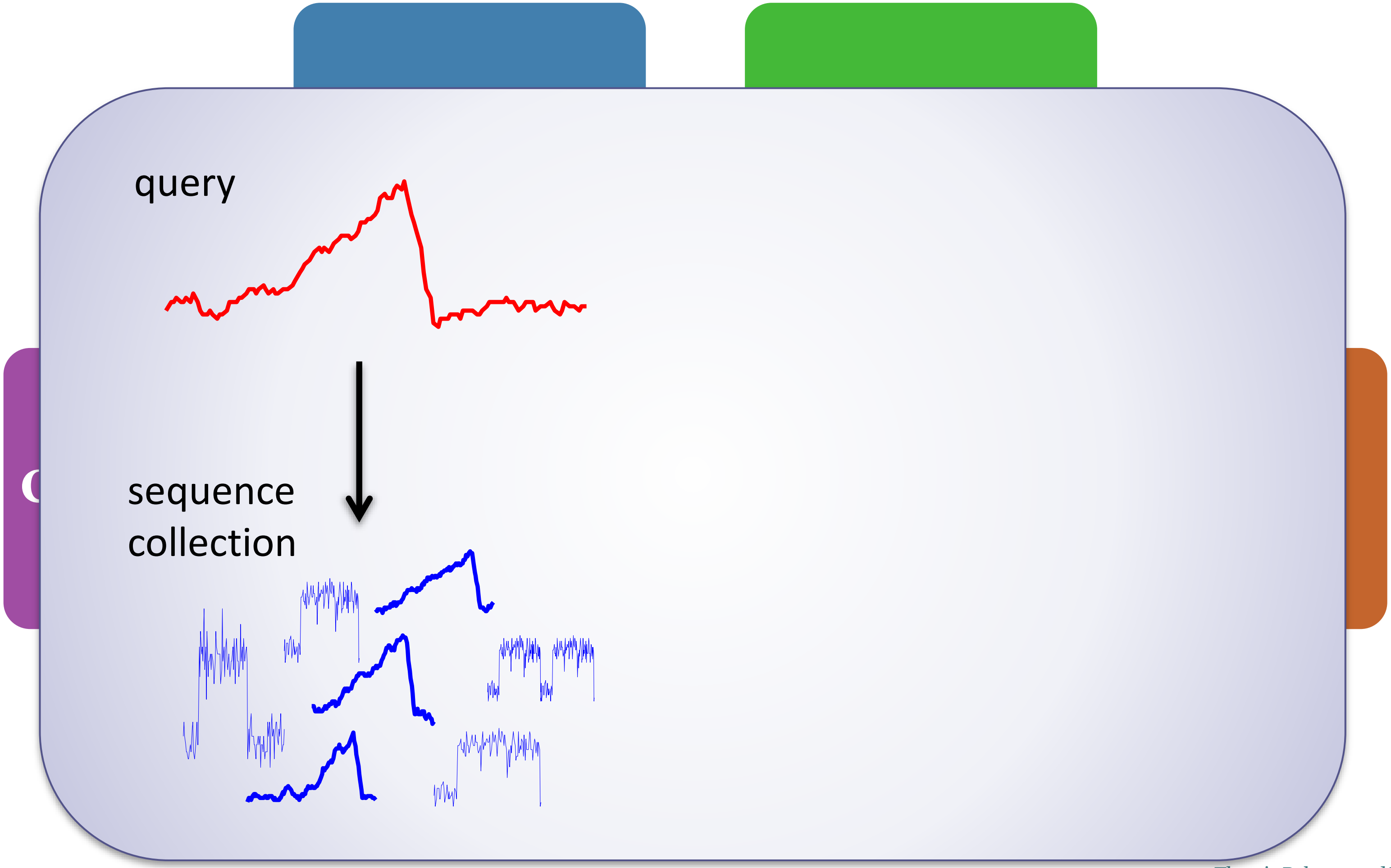


What do we want to do with them? - complex analytics

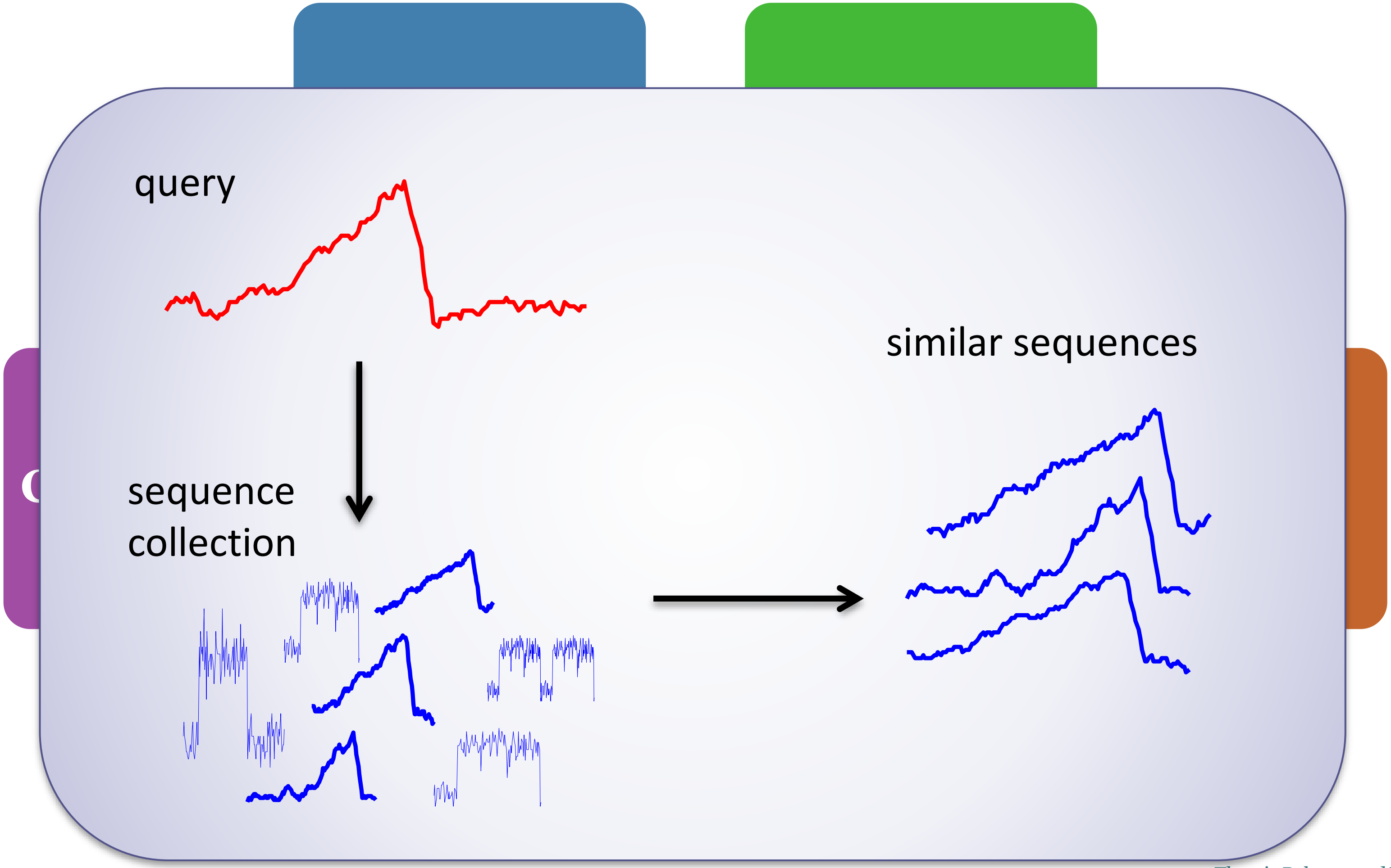
sequence collection



What do we want to do with them? - complex analytics



What do we want to do with them? - complex analytics



What do we want to do with them? - complex analytics

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

What do we want to do with them? - complex analytics

Euclidean

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

Dynamic Time Warping (DTW)

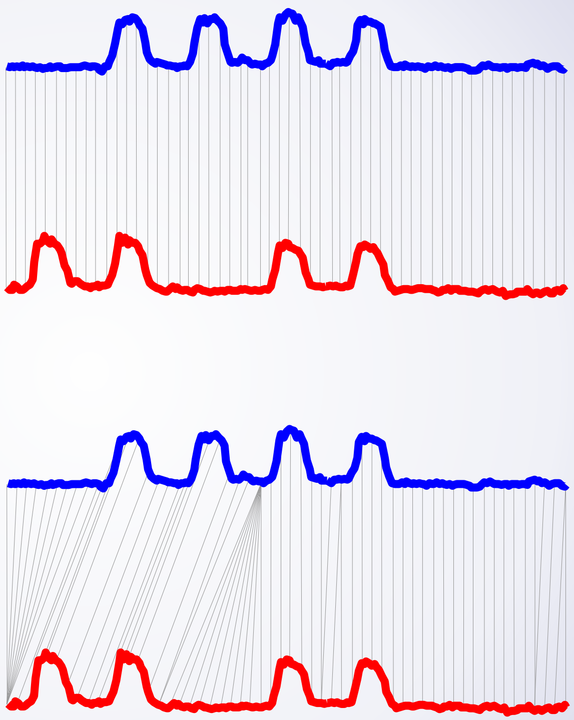
$$D_{dtw}(X,Y) = f(n,m)$$

$$f(i,j) = \|x_i - y_j\| + \min \begin{cases} f(i, j-1) \\ f(i-1, j) \\ f(i-1, j-1) \end{cases}$$

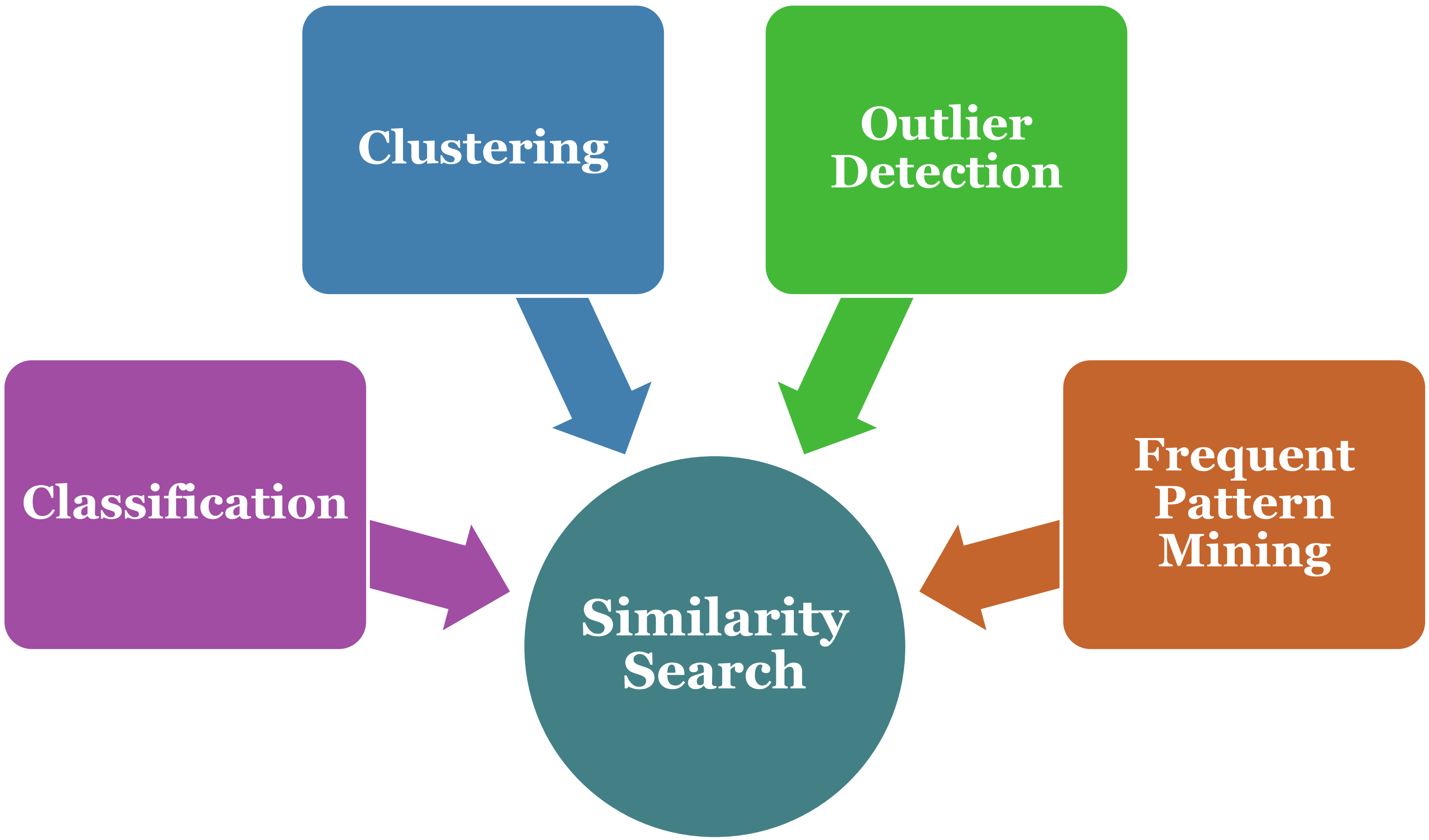
What do we want to do with them? - complex analytics

Euclidean

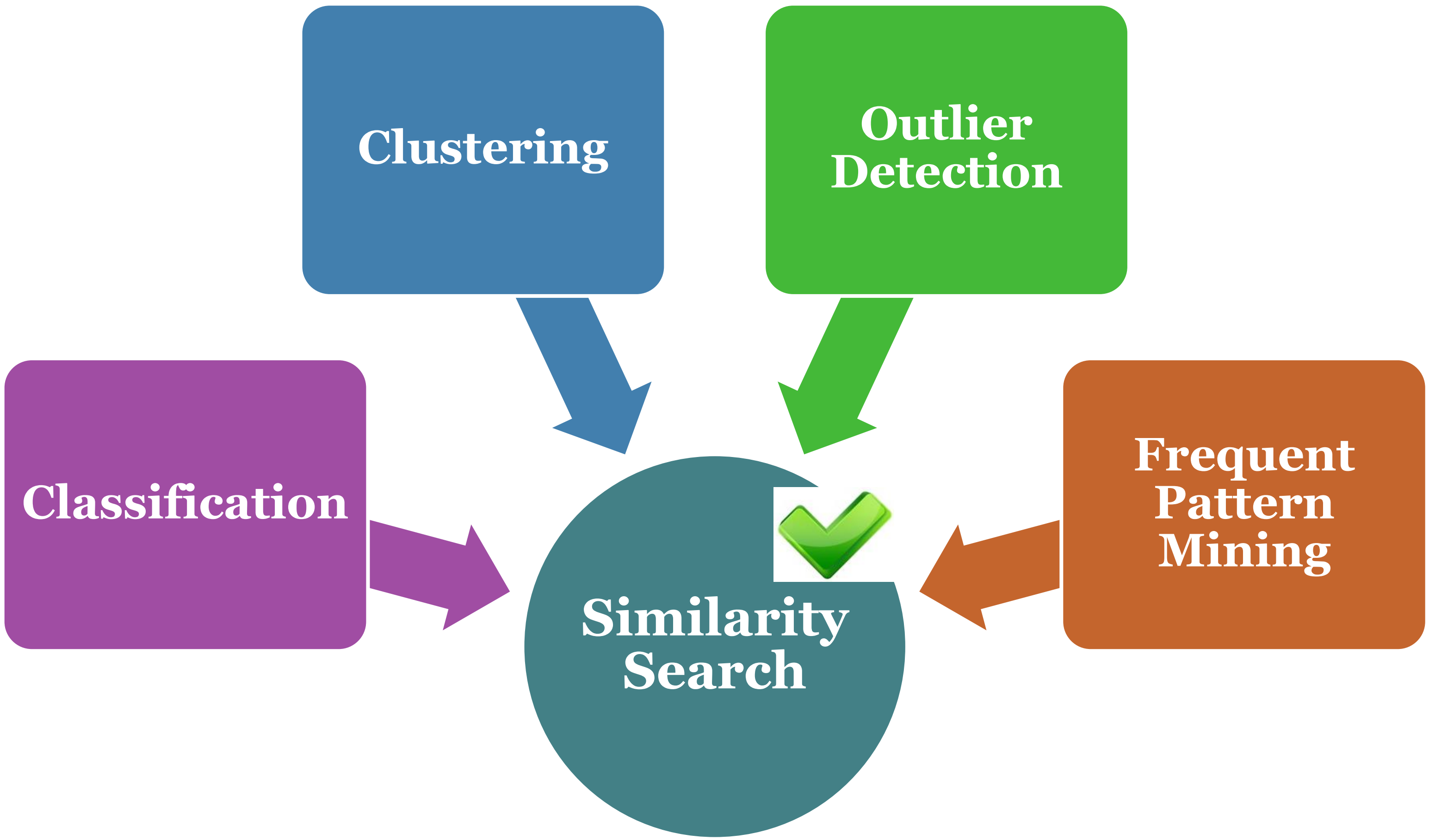
Dynamic Time Warping (DTW)



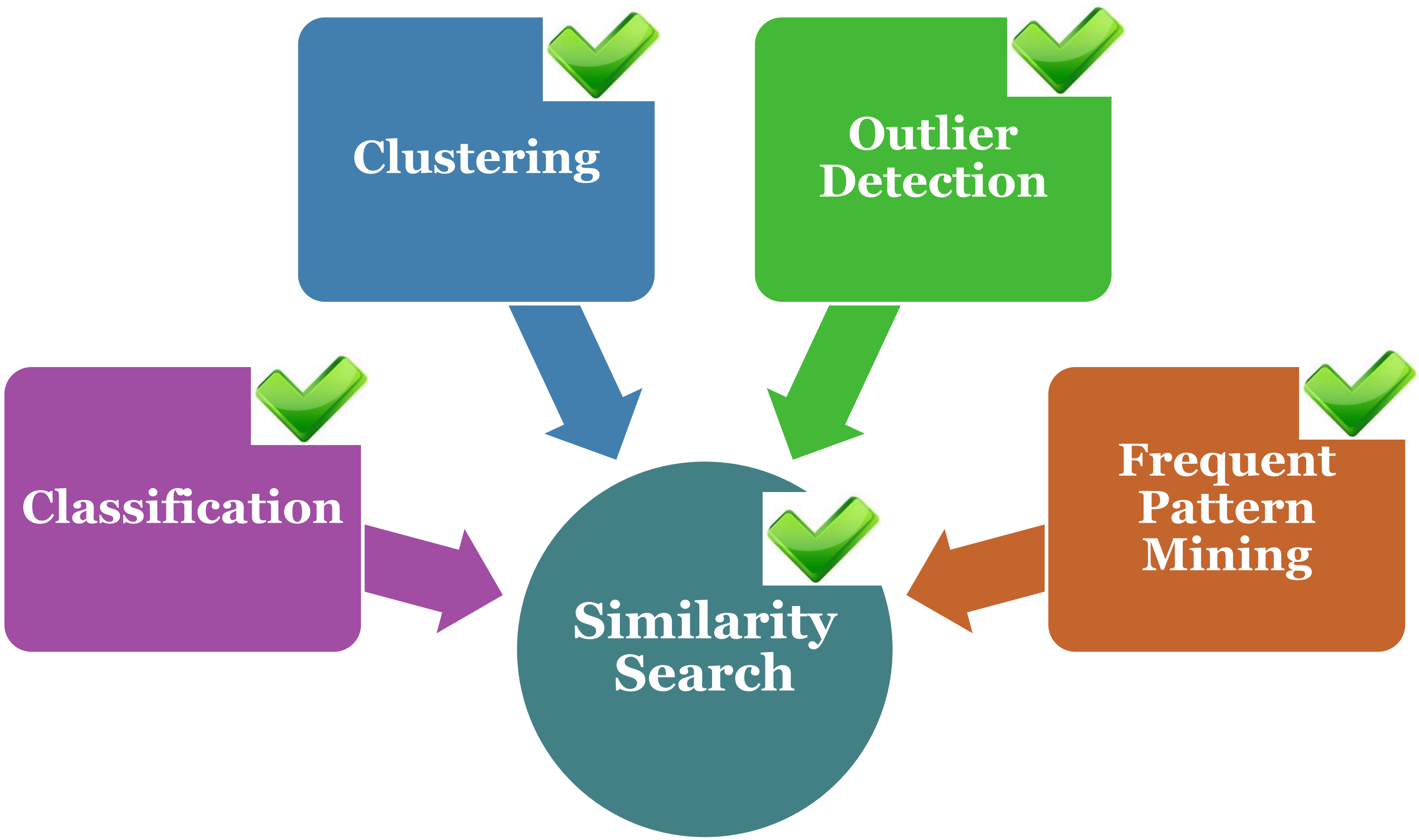
What do we want to do with them? - complex analytics



What do we want to do with them? - complex analytics



What do we want to do with them? - complex analytics



What do we want to do with them? - complex analytics

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

What do we want to do with them? - complex analytics

Clustering

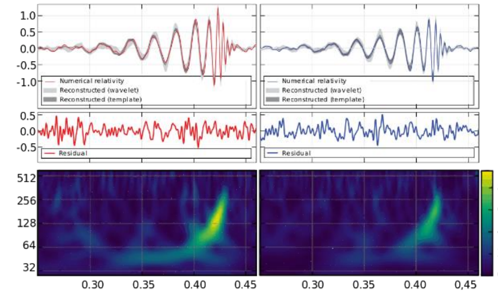
Outlier Detection

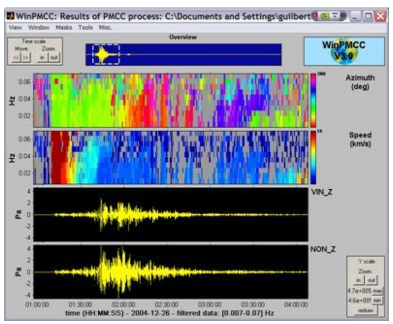
HARD, because of very high dimensionality: each data series has 100s-1000s of points!

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

Real Use-Cases

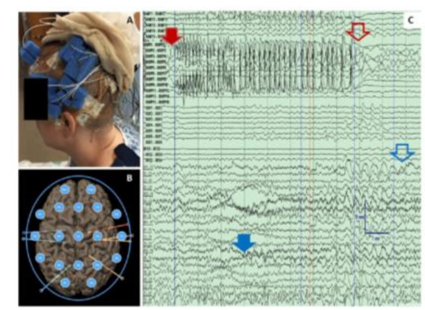
astrophysics: gravitational waves, **TB/hour**
partner: **European Gravitational Observatory (EGO)**
Pisa, Italy

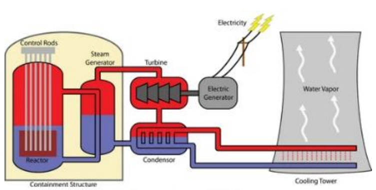


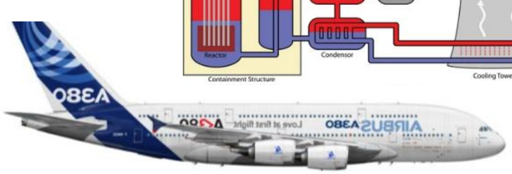


seismology: seismic sequences, **100s of TB**
partner: **Atomic Energy Commission (CEA)**
Paris, France

neuroscience: intracranial EEG sequences, **TB/patient**
partner: **Paris Brain Institute (ICM)**
Paris, France





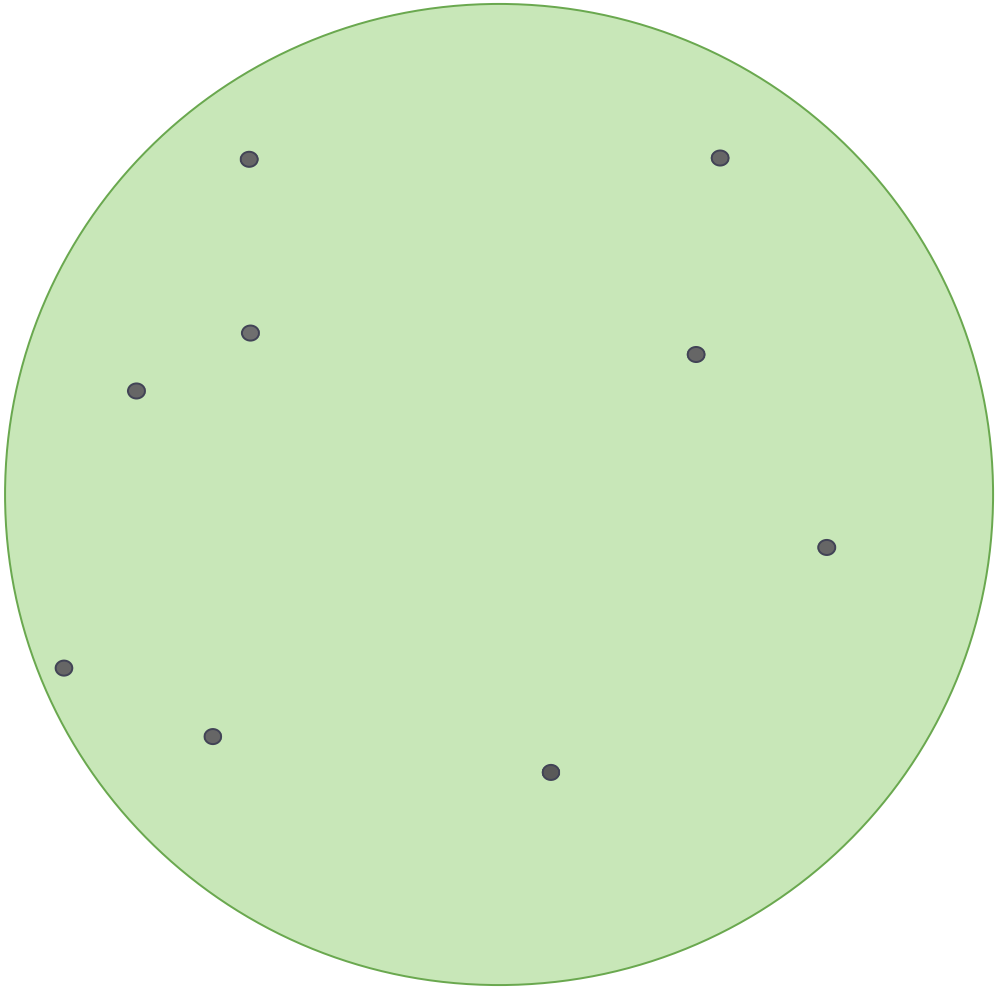


engineering: operation monitoring, **TB-PB**
partners: **Airbus / Électricité de France (EDF)**
Toulouse / Paris, France

Nearest Neighbor (NN) Queries…

Publications

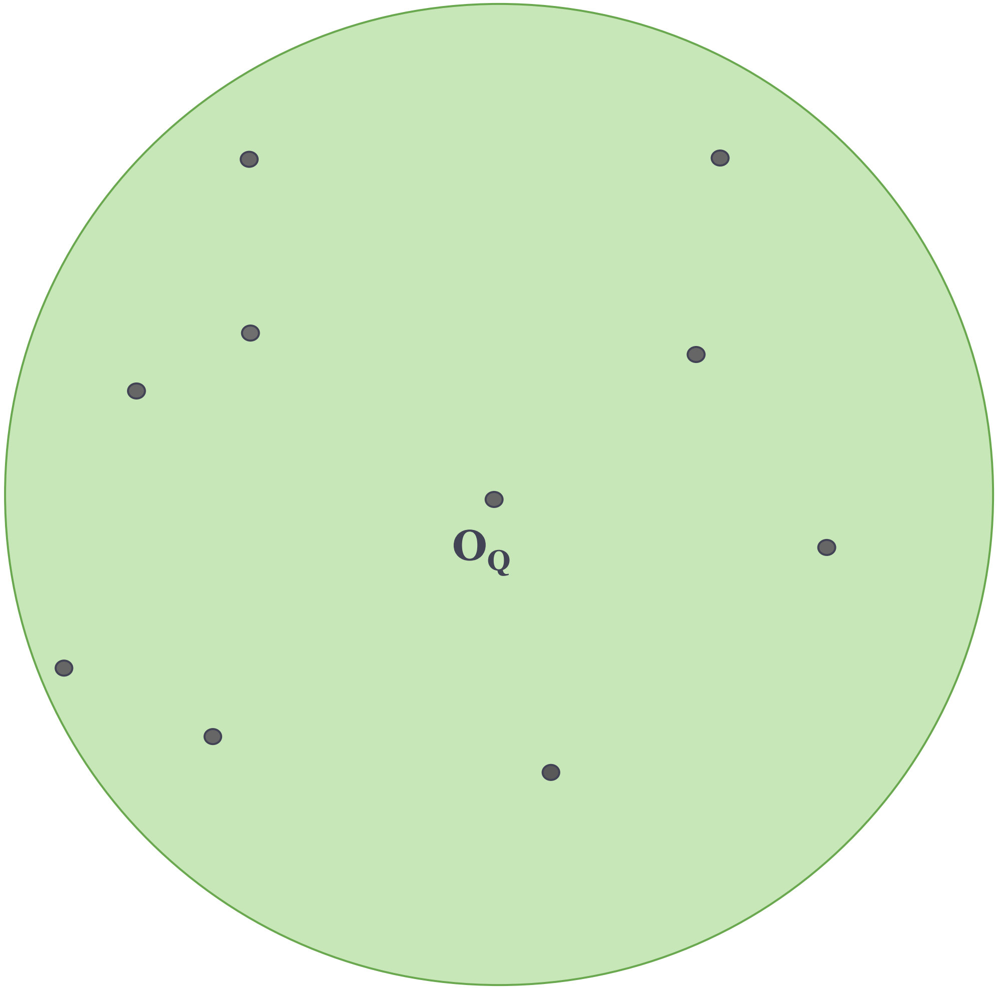
PVLDB'19



Nearest Neighbor (NN) Queries…

Publications

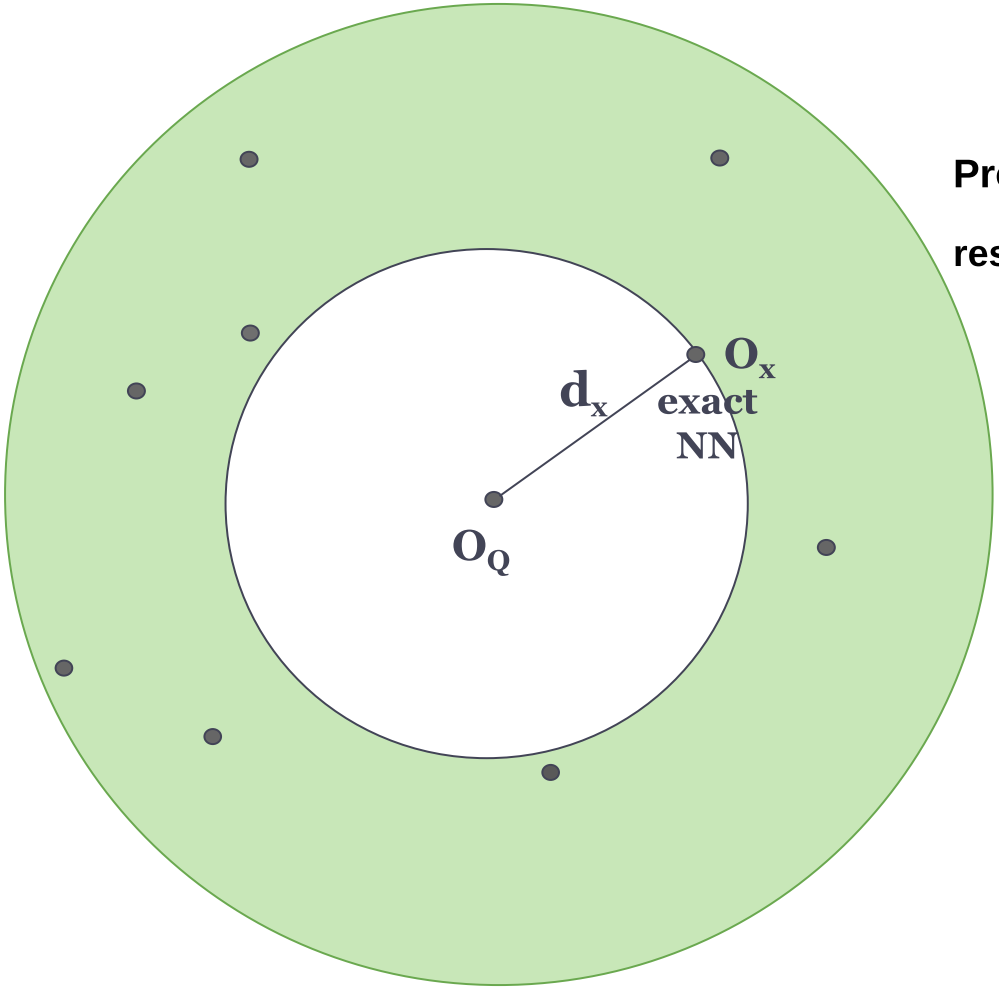
PVLDB'19



Nearest Neighbor (NN) Queries…

Publications

PVLDB'19



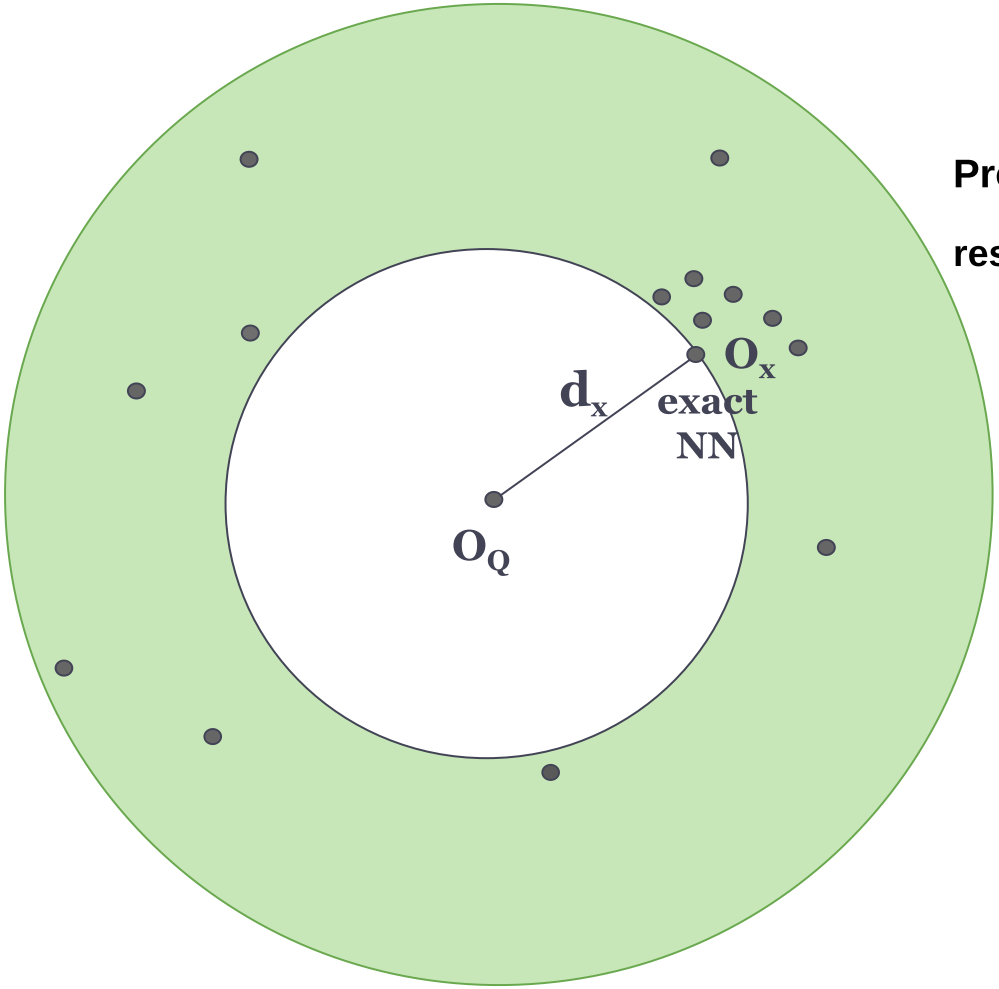
Prob(d_x = min{d_i}) = 1

result is exact NN

Nearest Neighbor (NN) Queries…

Publications

PVLDB'19



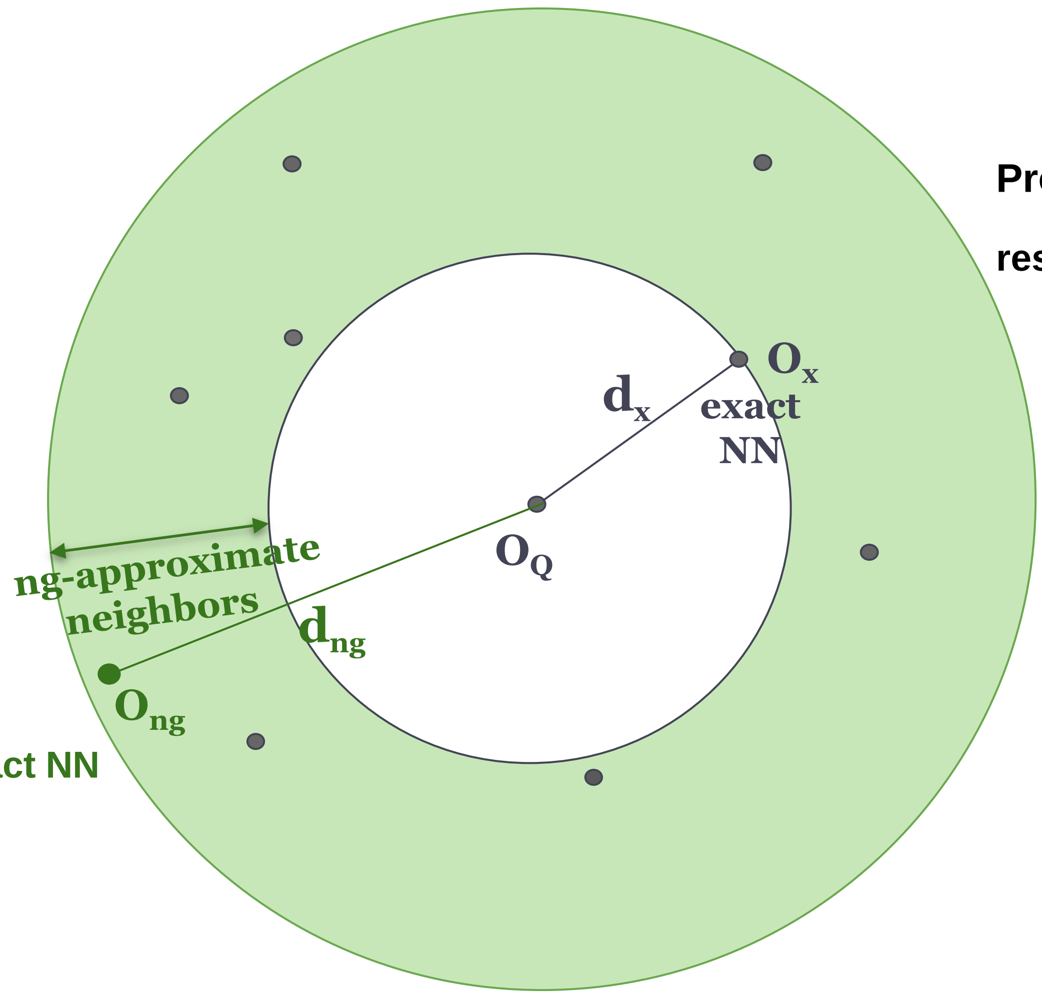
Prob(d_x = min{d_i}) = 1

result is exact NN

Nearest Neighbor (NN) Queries…

Publications

PVLDB‘19



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

result is exact NN

$\mathrm{Prob}(d_{ng} <>= ?) = ?$

result within ? of exact NN

Nearest Neighbor (NN) Queries…

Publications

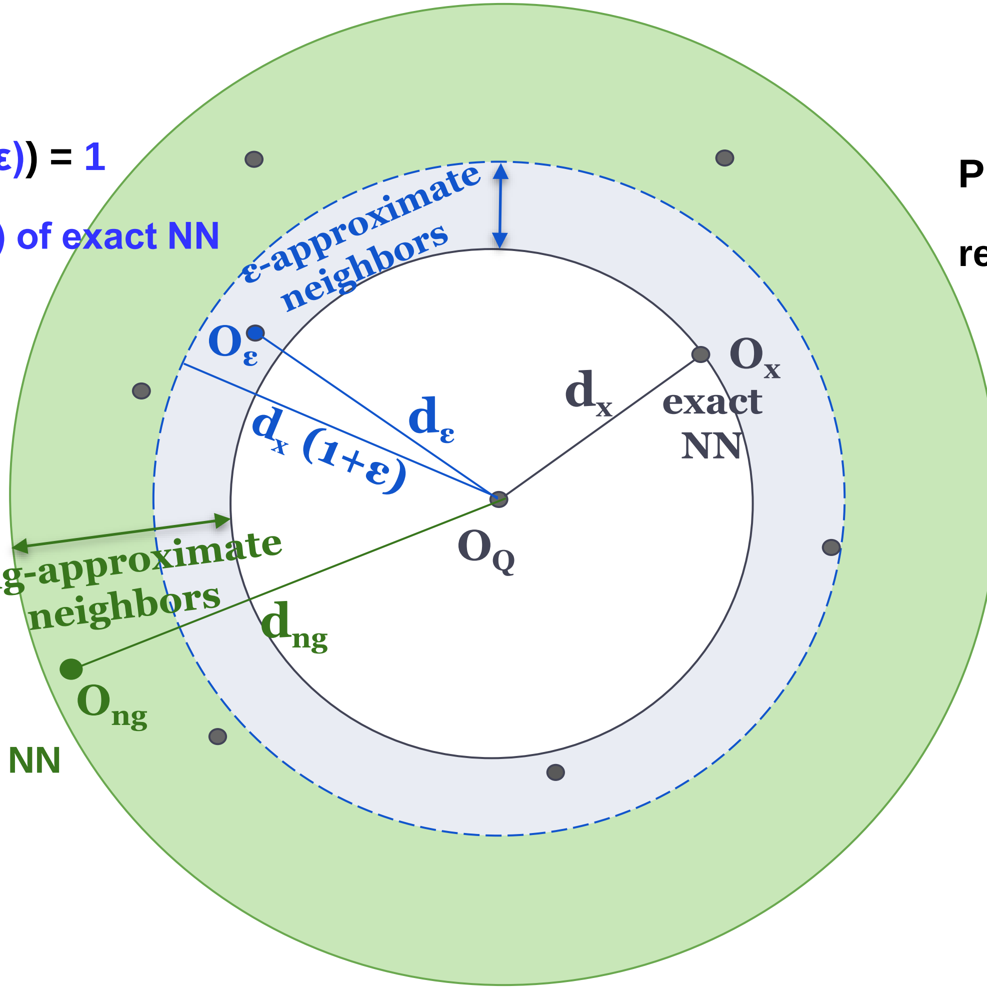
PVLDB‘19

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

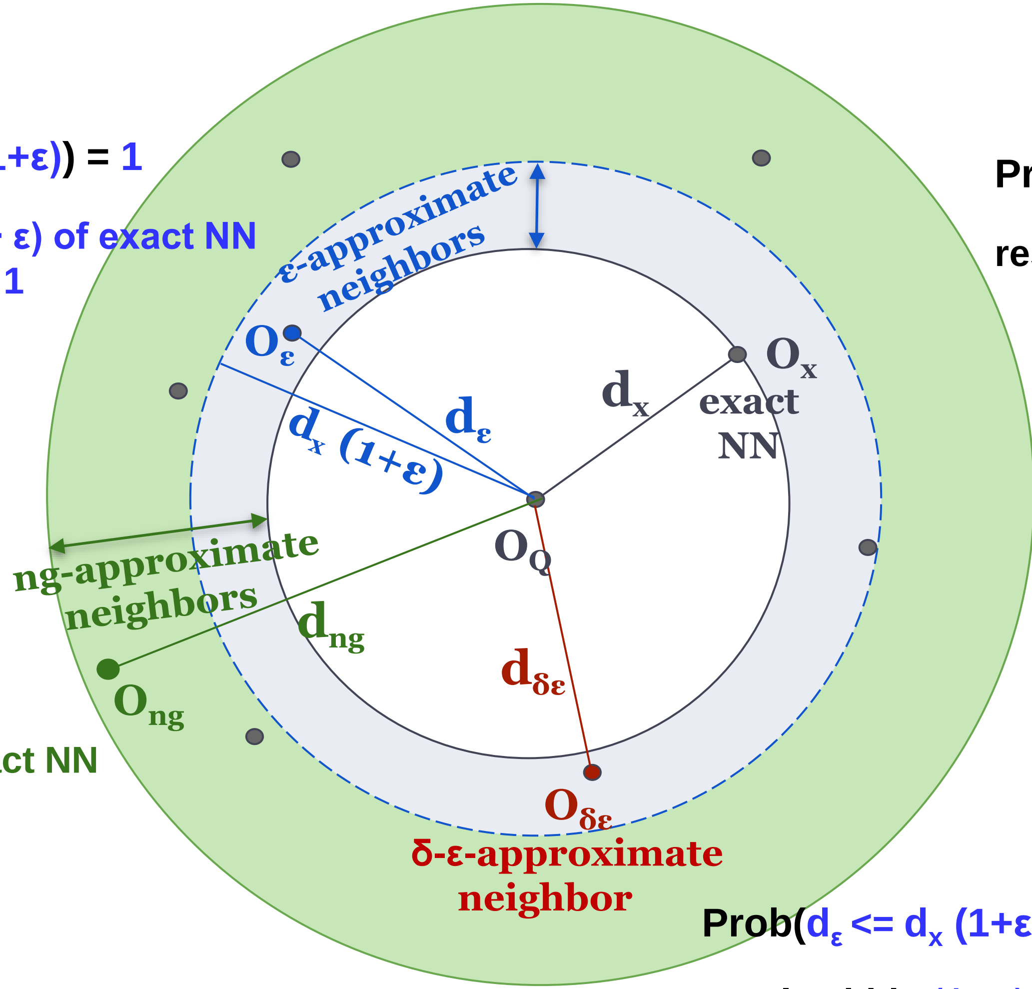
PVLDB'19

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

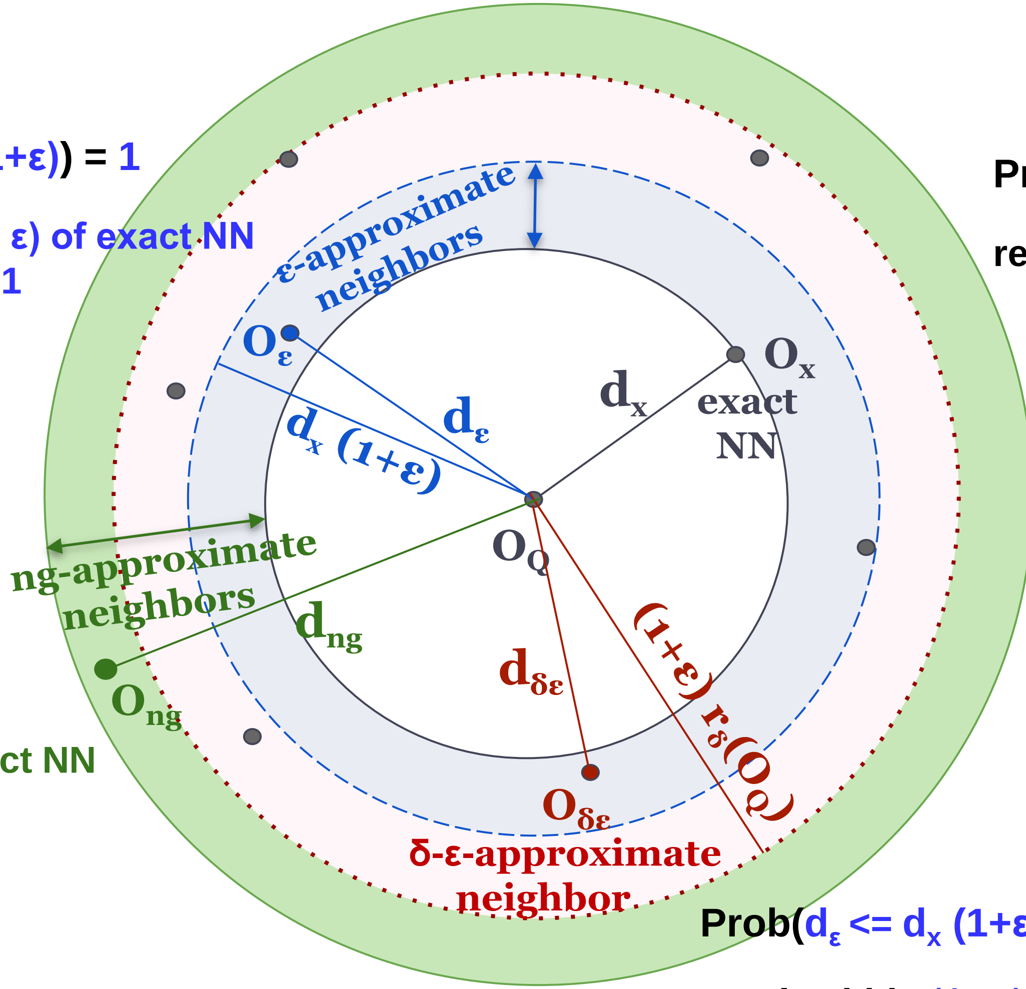
PVLDB‘19

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

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

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

result is exact NN



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

result within ? of exact NN

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

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

Problem Variations

High-d Vectors Distance Measures

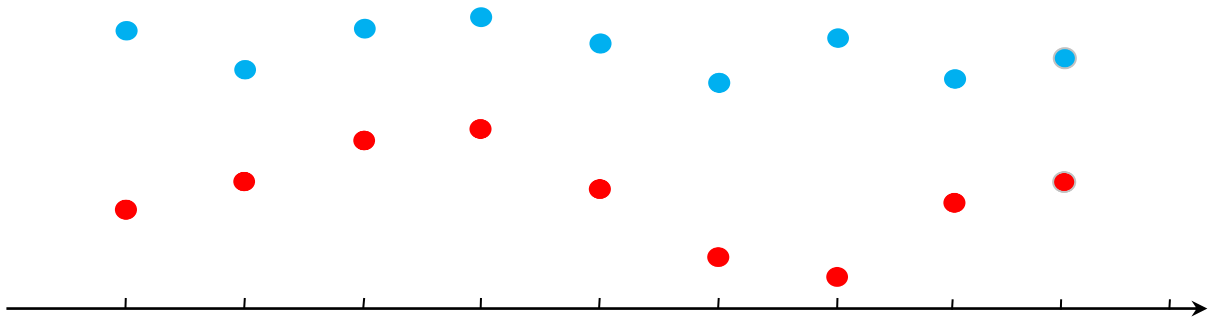
- similarity search is based on measuring distance between vectors
- A variety of distance measures have been proposed
  - L_p distances ($0<p\leq2, \infty$), (Euclidean for p = 2)
  - Cosine distance
  - Correlation
  - Hamming distance
  - …

Problem Variations

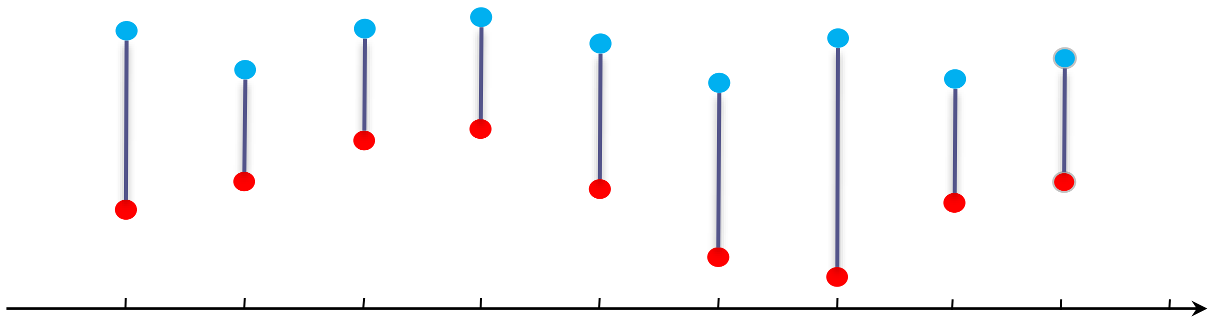
High-d Vectors Distance Measures

- similarity search is based on measuring distance between vectors
- A variety of distance measures have been proposed
  - L_p distances ($0<p\leq 2, \infty$), (Euclidean for p = 2)
  - Cosine distance
  - Correlation
  - Hamming distance
  - …

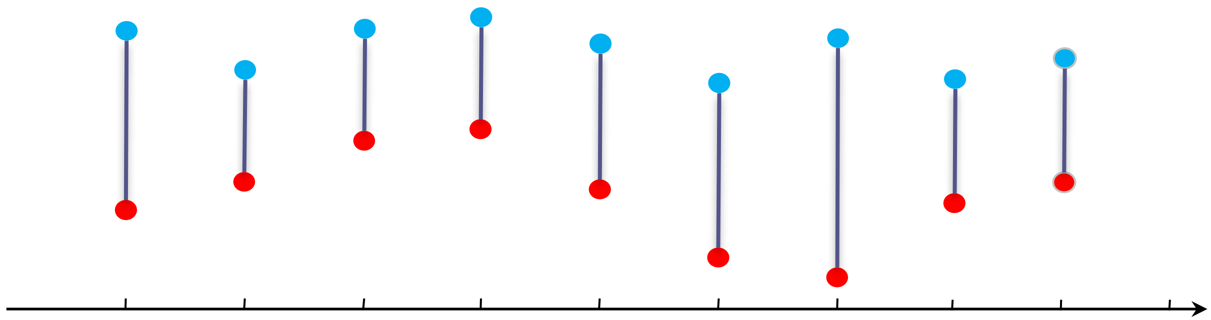
Euclidean Distance



Euclidean Distance



Euclidean Distance



- Euclidean distance
  - pair-wise point distance

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

Similarity Matching
Fast Euclidean Distance

- similarity matching requires many distance computations
  - can significantly slow down processing
    - because of large number of data series in the collection
    - because of high dimensionality of each data series

Similarity Matching
Fast Euclidean Distance

- similarity matching requires many distance computations
  - can significantly slow down processing
    - because of large number of data series in the collection
    - because of high dimensionality of each data series

- in case of Euclidean Distance, we can speedup processing by
  - smart implementation of distance function
  - early abandoning

Similarity Matching
Fast Euclidean Distance

- similarity matching requires many distance computations
  - can significantly slow down processing
    - because of large number of data series in the collection
    - because of high dimensionality of each data series

- in case of Euclidean Distance, we can speedup processing by
  - smart implementation of distance function
  - early abandoning

- result in considerable performance improvement

Publications

Keogh-DMKD'03

Similarity Matching
Fast Euclidean Distance

- smart implementation of distance function

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

Publications

Keogh-DMKD'03

Similarity Matching
Fast Euclidean Distance

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

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

Publications

Keogh-DMKD'03

Similarity Matching
Fast Euclidean Distance

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

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

- does not alter the results
- saves precious CPU cycles

Publications

Keogh-DMKD'03

Similarity Matching
Fast Euclidean Distance

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

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

Publications

Keogh-DMKD'03

Similarity Matching
Fast Euclidean Distance

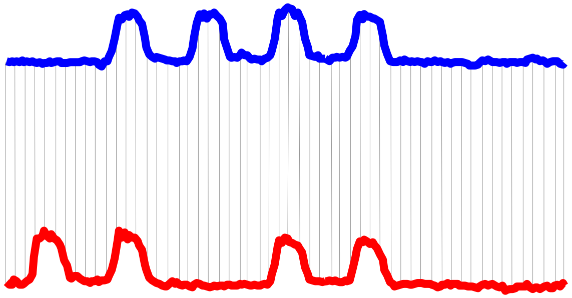
- early abandoning
  - stop the distance computation as soon as it exceeds the value of bsf

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

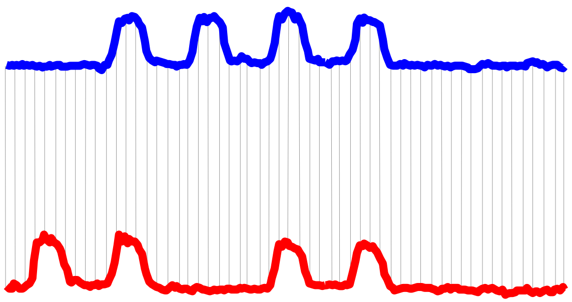
- does not alter the results
- avoids useless computations

Distance Measures: Euclidean, DTW, LCSS

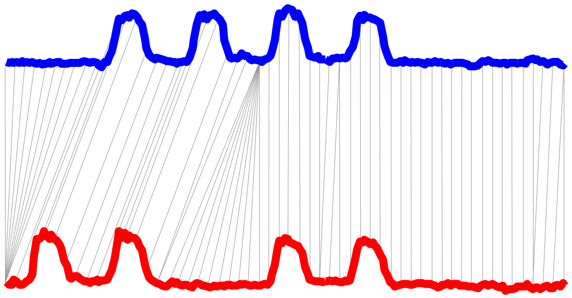
- Euclidean
  - rigid



Distance Measures: Euclidean, DTW, LCSS



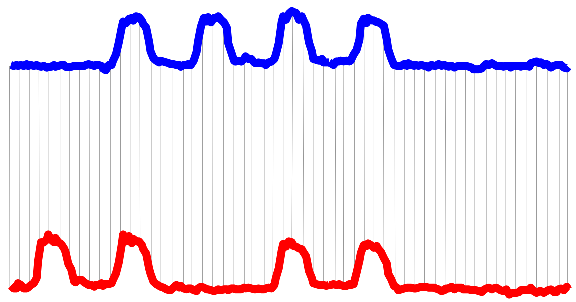
- Euclidean
  - rigid



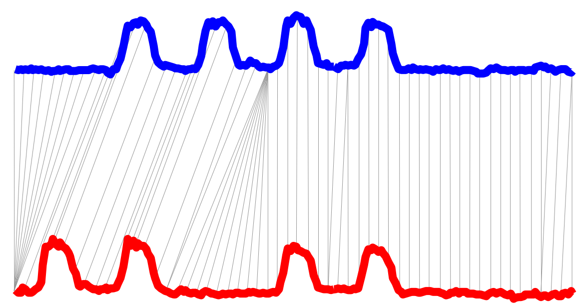
- Dynamic Time Warping (DTW)
  - allows local scaling

Distance Measures: Euclidean, DTW, LCSS

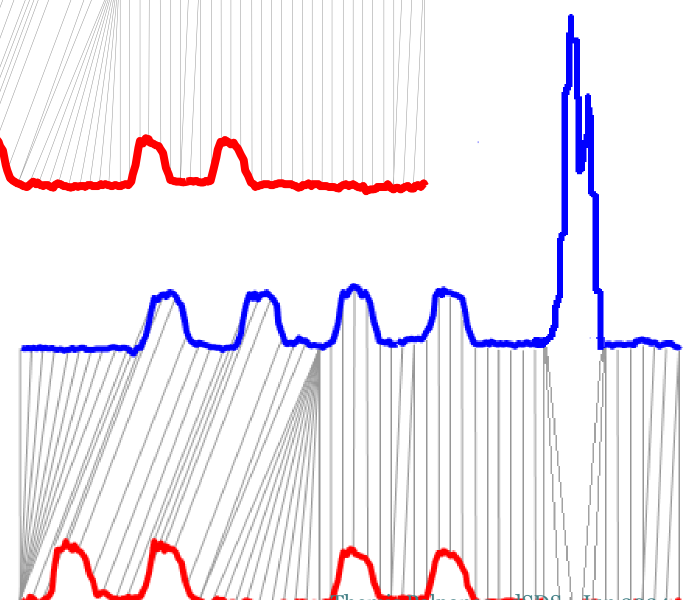
- Euclidean
  - rigid



- Dynamic Time Warping (DTW)
  - allows local scaling



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



Pearson's Correlation Coefficient

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

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

Pearson's Correlation Coefficient

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

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

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

Pearson's Correlation Coefficient

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

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

- takes values in [-1,1]
  - 0 – no correlation
  - -1, 1 – inverse/direct correlation
- there is a statistical test connected to PC, where null hypothesis is the no correlation case (correlation coefficient = 0)
  - test is used to ensure that the correlation similarity is not caused by a random process

PC and ED

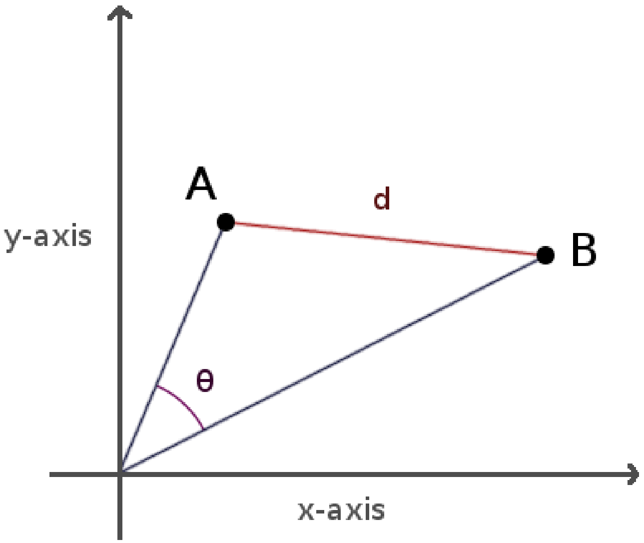
- Euclidean distance: $ED = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2},$
- In case of Z-normalized data series (mean = 0, stddev = 1):

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

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

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

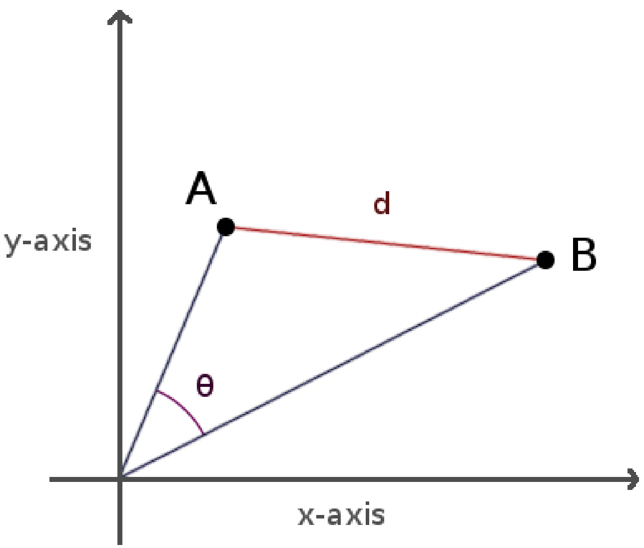
Distance Measures: Cosine Distance



$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}},$$

- Cosine distance = 1 - cosine similarity

Distance Measures: Cosine Distance



$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

- Cosine distance = 1 - cosine similarity

- ED vs. Cosine similarity
  - If A and B are normalized to unit length in L_2, the square of ED is proportional to the cosine distance:
  - $||A||_2=||B||_2=1 \rightarrow ||A-B||_2=2-2\cos(A,B)$

Maximum Inner Product Search (MIPS)

- Problem Definition:
  - Given a collection of candidate vectors S and a query Q , find a candidate vector C maximizing the inner product with the query: :
    - Given $S \subset R^d$ and $Q \in R^d$, $C = \text{argmax}_{X \in S} Q^T X$

Maximum Inner Product Search (MIPS)

- Problem Definition:
  - Given a collection of candidate vectors S and a query Q , find a candidate vector C maximizing the inner product with the query: :
    - Given $S \subset R^d$ and $Q \in R^d$, $C = \text{argmax}_{X \in S}\, Q^T X$
  - MIPS is closely related to NN search:
    - If $\|Q\|_2 = 1$, $\|Q - X\|_2 = 1 + \|X\|_2 - 2Q^T X$
  - MIPS and NN search are equivalent when all vectors X in S have constant length c
  - Otherwise, MIPS can be converted to NN search with ED or Cosine similarity [1][2][3]

[1] Anshumali Shrivastava and Ping Li. 2014a. Asymmetric LSH (ALSH) for Sublinear Time Maximum Inner Product Search (MIPS). In NIPS. 2321–2329.
[2] Yoram Bachrach, Yehuda Finkelstein, Ran Gilad-Bachrach, Liran Katzir, Noam Koenigstein, Nir Nice, and Ulrich Paquet. 2014. Speeding Up the Xbox Recommender System Using a Euclidean Transformation for Inner-product Spaces. In RecSys. 257–264.
[3] B. Neyshabur and N. Srebro. 2014. On Symmetric and Asymmetric LSHs for Inner Product Search. ArXiv e-prints (Oct. 2014).

Data Series Similarity Search Pre-processing Tasks

Pre-Processing
z-Normalization

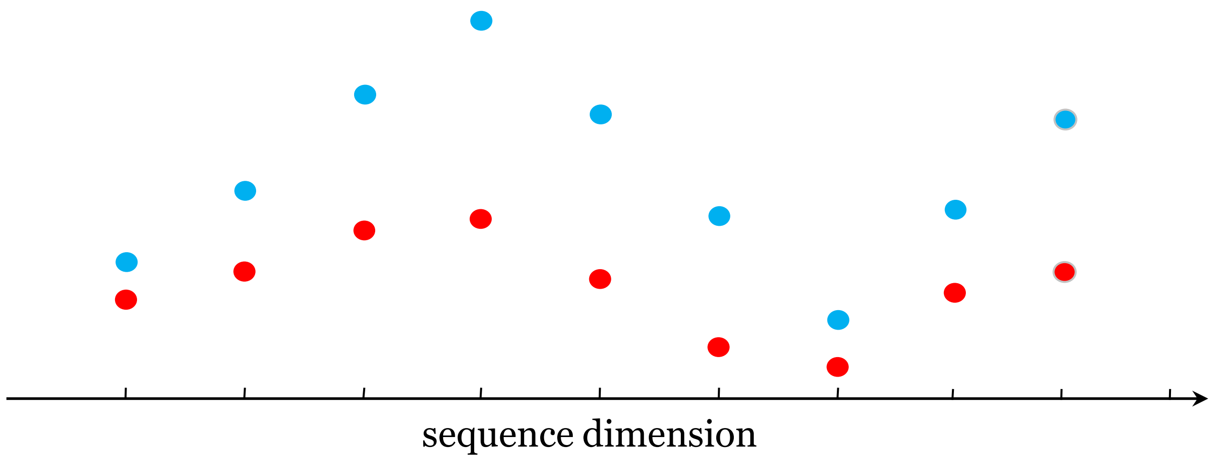
- data series encode trends
- usually interested in identifying similar trends

Pre-Processing
z-Normalization

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

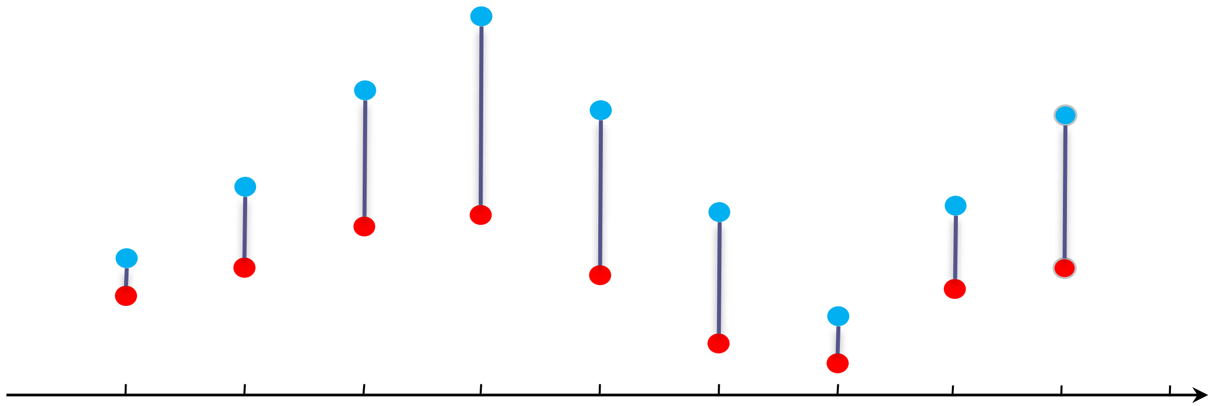
- but absolute values may mask this similarity

Pre-Processing
z-Normalization



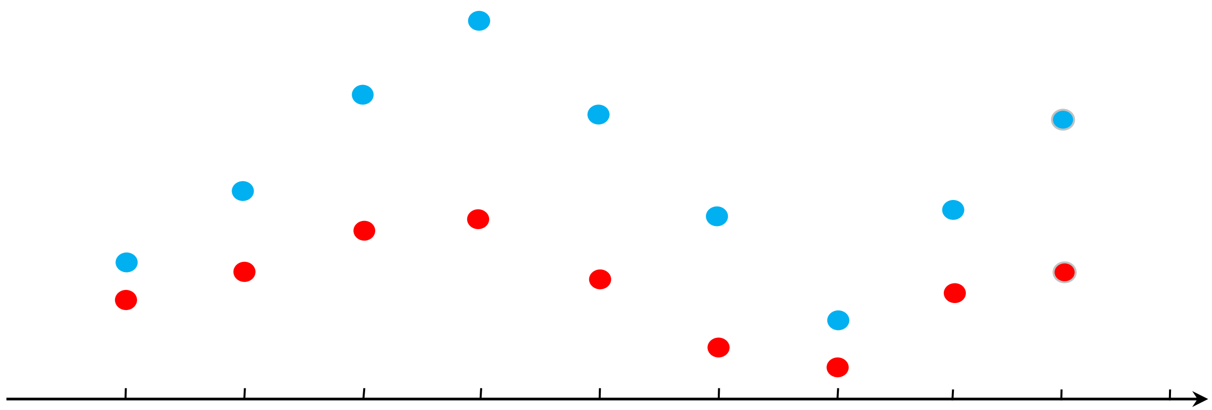
- two data series with similar trends

Pre-Processing
z-Normalization



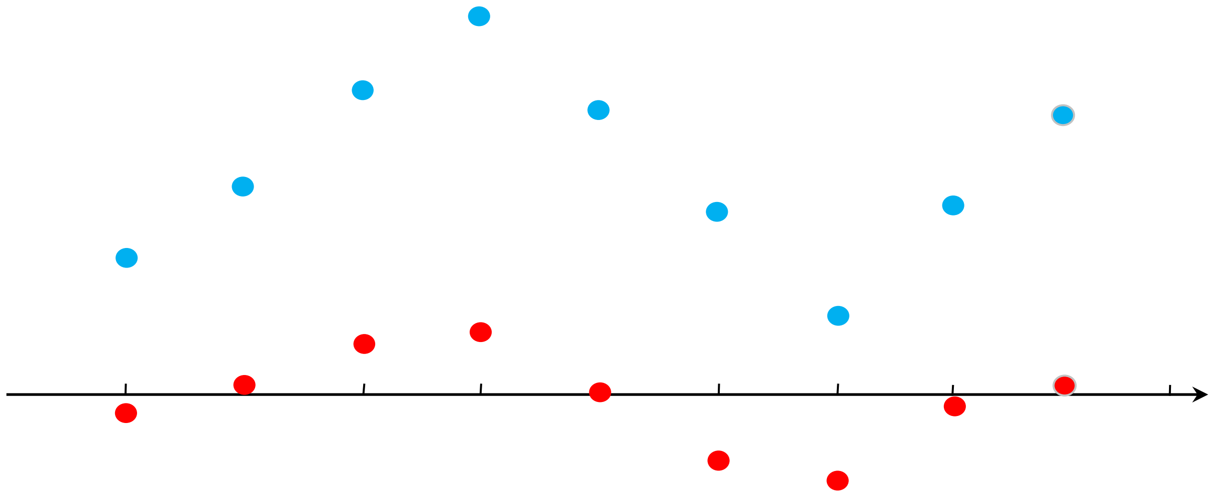
- two data series with similar trends
- but large distance…

Pre-Processing
z-Normalization



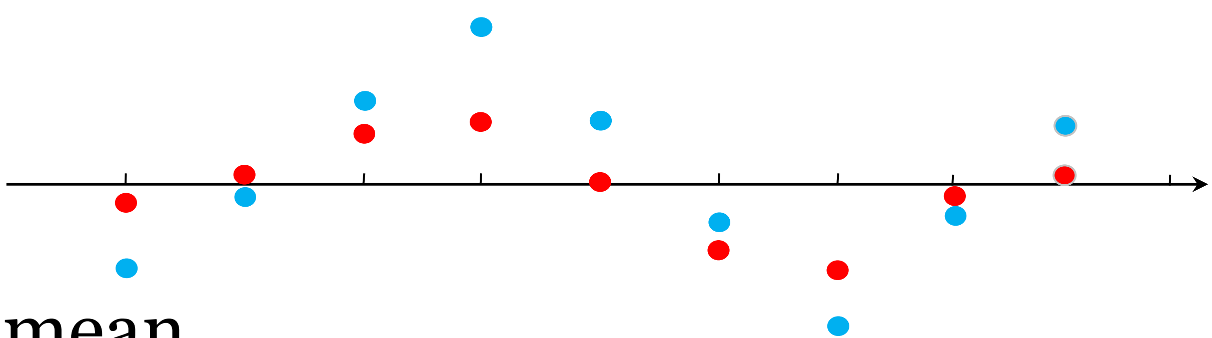
- zero mean
  - compute the mean of the sequence
  - subtract the mean from every value of the sequence

Pre-Processing z-Normalization



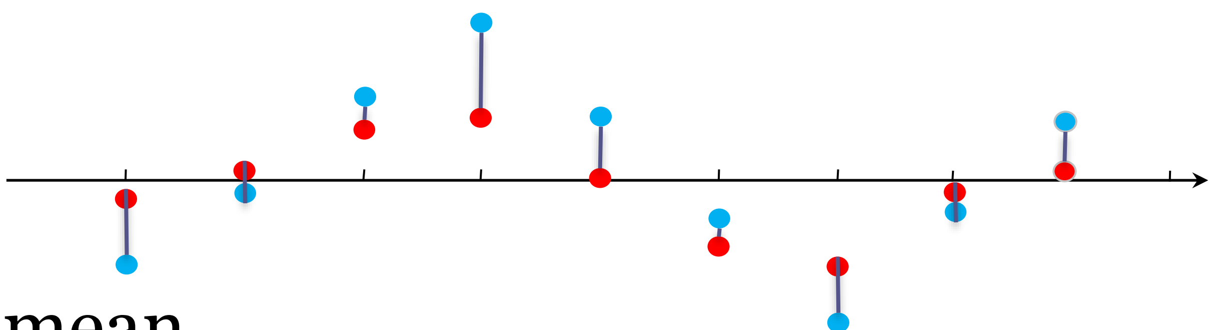
- zero mean
  - compute the mean of the sequence
  - subtract the mean from every value of the sequence

Pre-Processing
z-Normalization



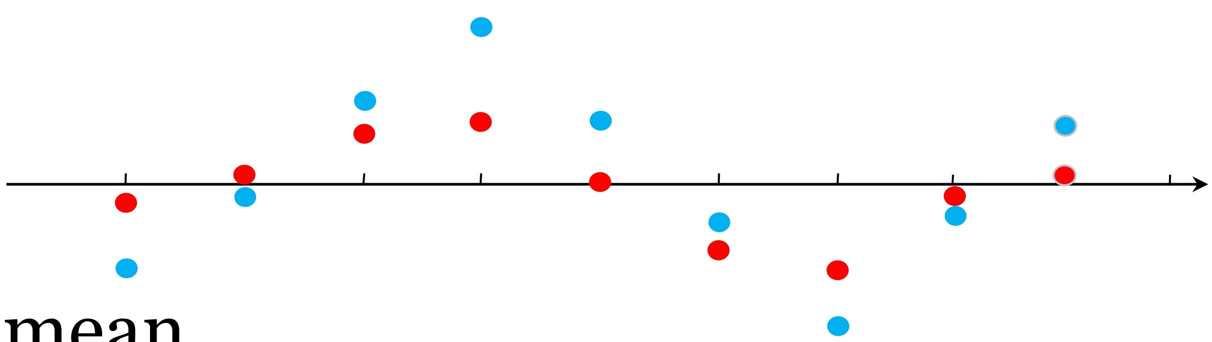
- zero mean
  - compute the mean of the sequence
  - subtract the mean from every value of the sequence

Pre-Processing z-Normalization



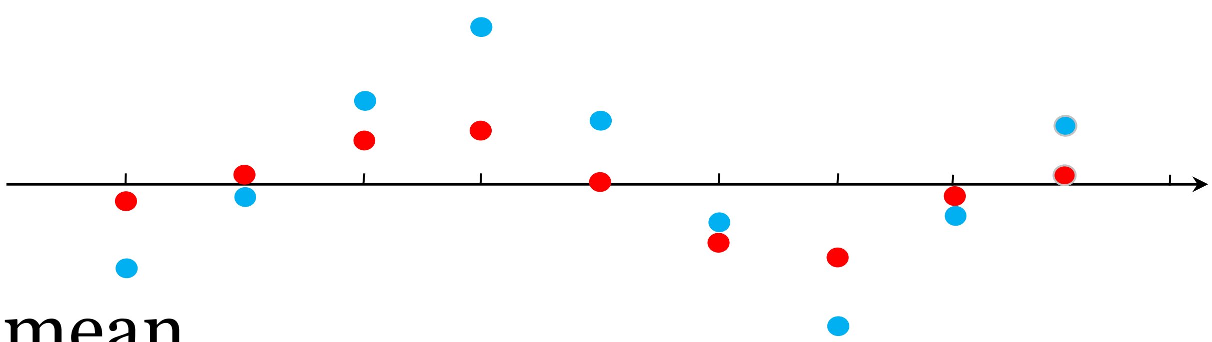
- zero mean
  - compute the mean of the sequence
  - subtract the mean from every value of the sequence

Pre-Processing
z-Normalization



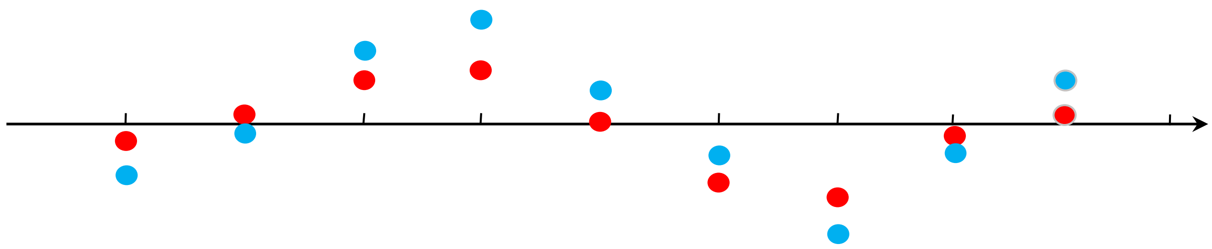
- zero mean
- standard deviation one
  - compute the standard deviation of the sequence
  - divide every value of the sequence by the stddev

Pre-Processing
z-Normalization



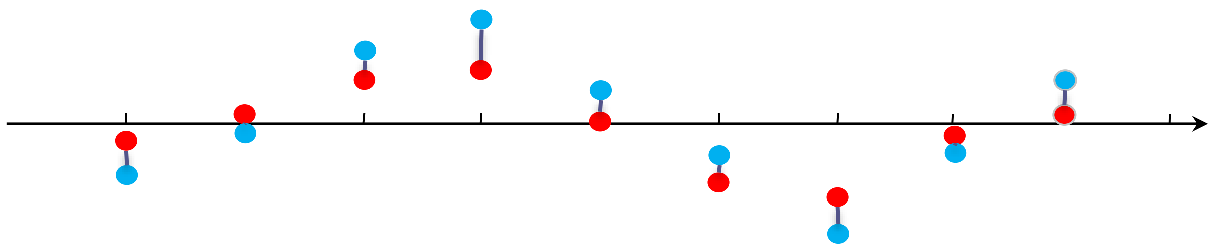
- zero mean
- standard deviation one
  - compute the standard deviation of the sequence
  - divide every value of the sequence by the stddev

Pre-Processing
z-Normalization



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

Pre-Processing
z-Normalization



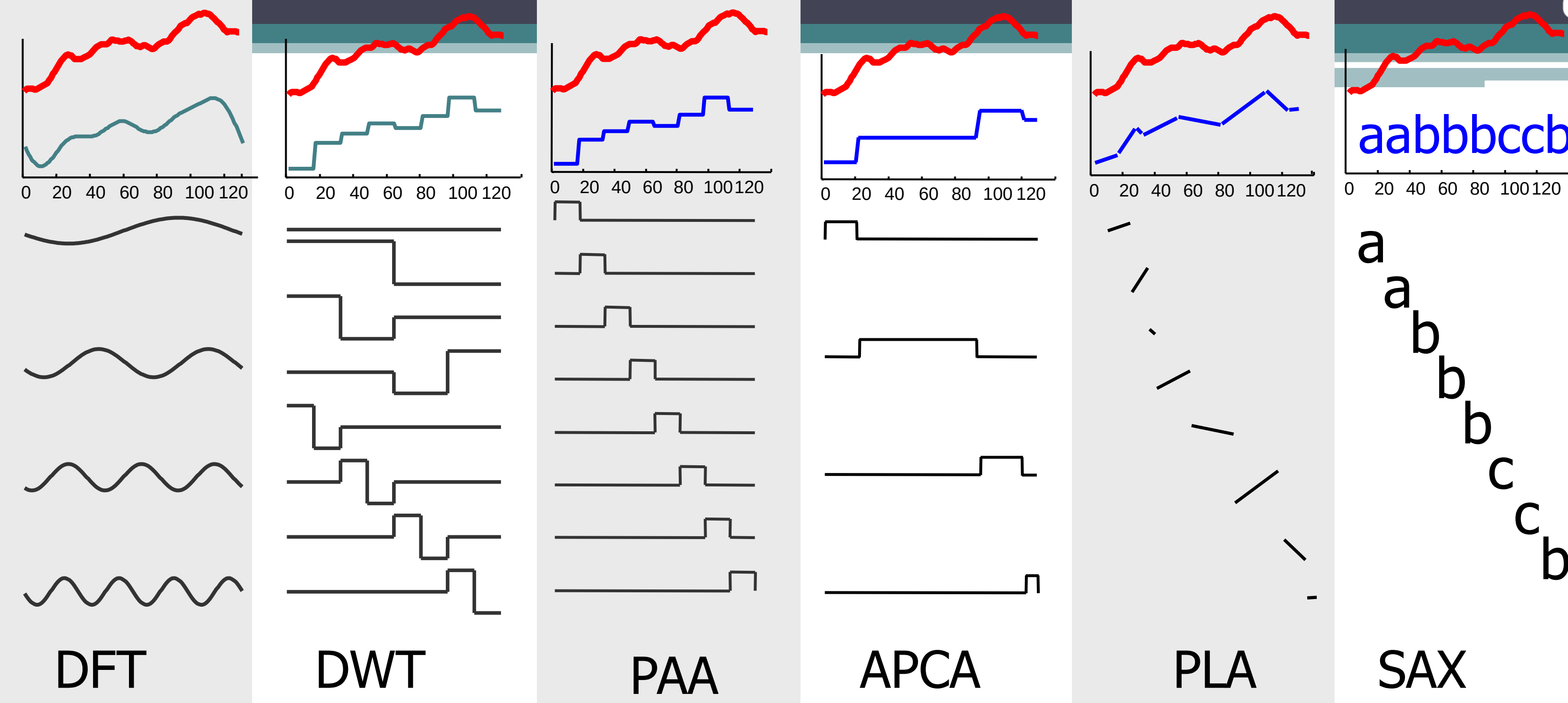
- zero mean
- standard deviation one

Pre-Processing
z-Normalization

- when to z-normalize
  - interested in trends

Pre-Processing
z-Normalization

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



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

Comparison of Representations

- which representation is the best?

Comparison of Representations

- which representation is the best?

- depends on data characteristics
  - periodic, smooth, spiky, …

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 Search Common Framework

GEMINI Framework

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

GEMINI Framework

Publications

Faloutsos-SIGMOD'94

GEMINI Solution: Quick filter-and-refine:

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

Publications

Faloutsos-SIGMOD'94

GEMINI: contractiveness

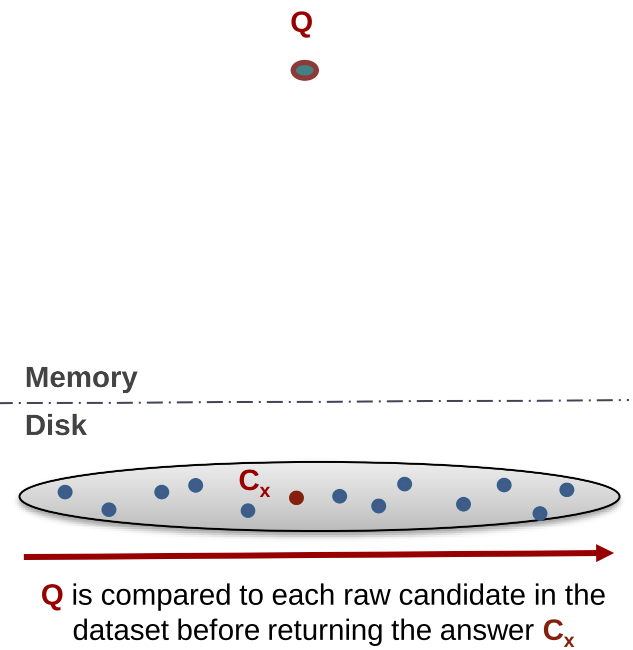
- GEMINI works when:

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

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

Data Series Similarity Search Classes of Methods

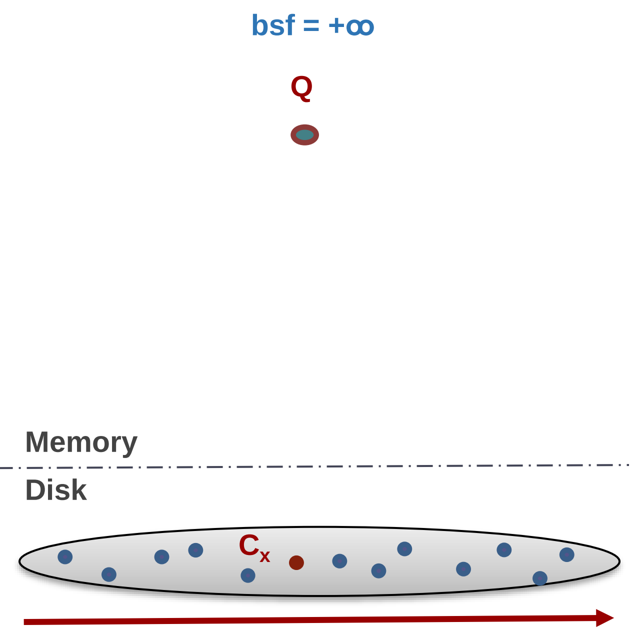
Similarity Matching
Serial Scan



(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching
Serial Scan

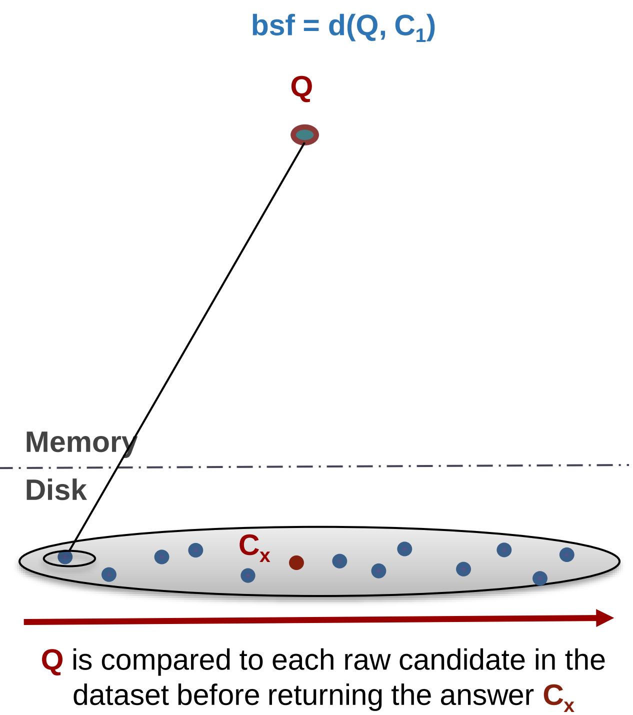


Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

Answering a similarity search query using different access paths

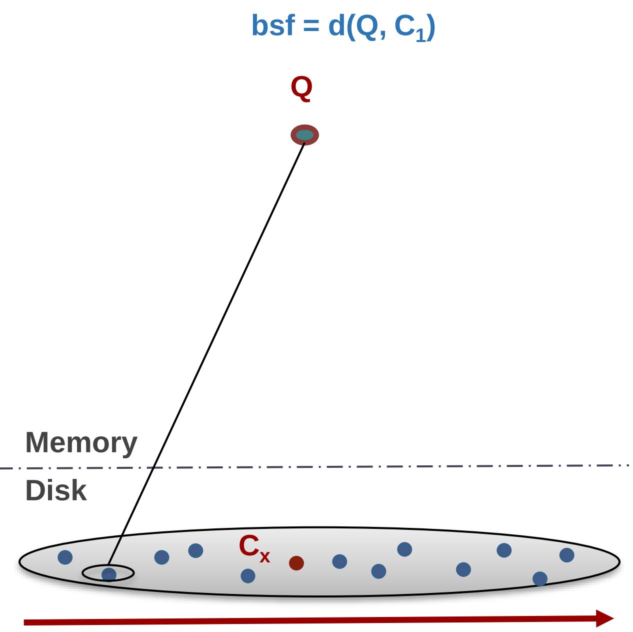
Similarity Matching
Serial Scan



(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching
Serial Scan

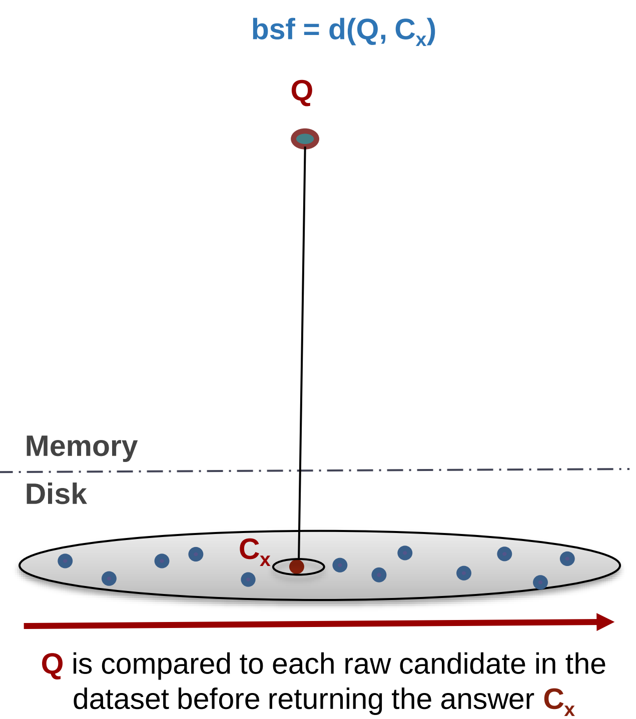


Q is compared to each raw candidate in the dataset before returning the answer **C_x**

(a) Serial scan

Answering a similarity search query using different access paths

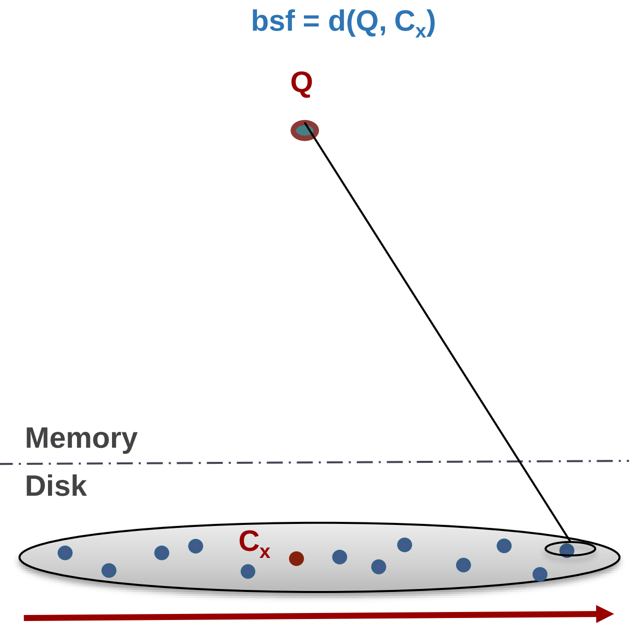
Similarity Matching
Serial Scan



(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching
Serial Scan

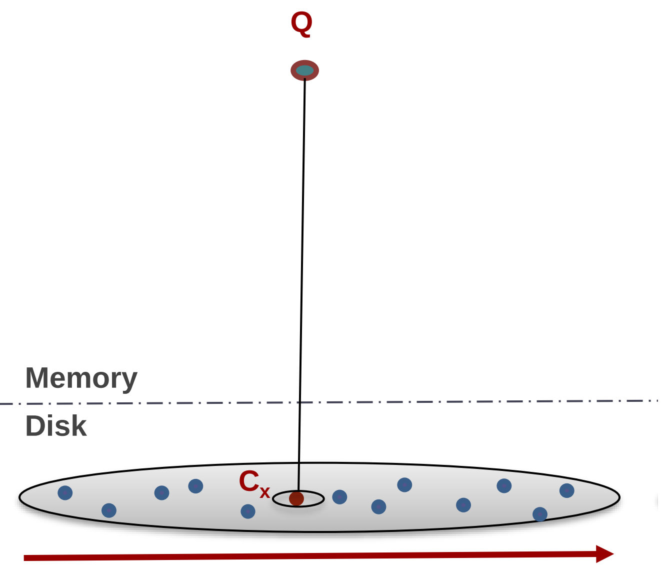


Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching
Serial Scan

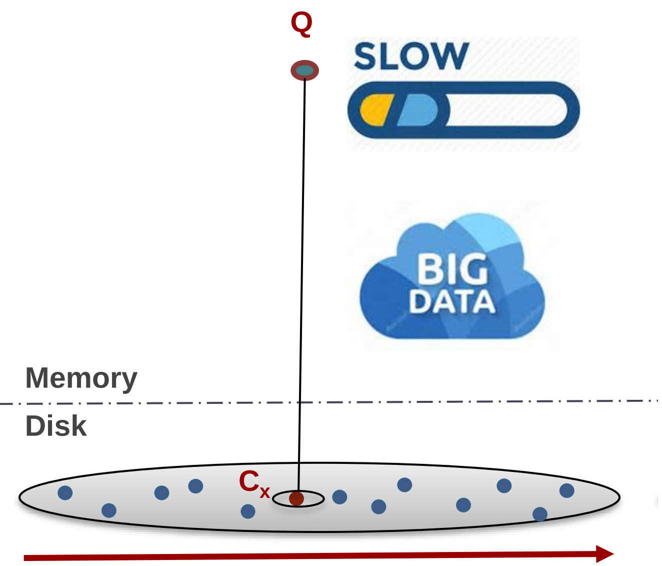


Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching
Serial Scan

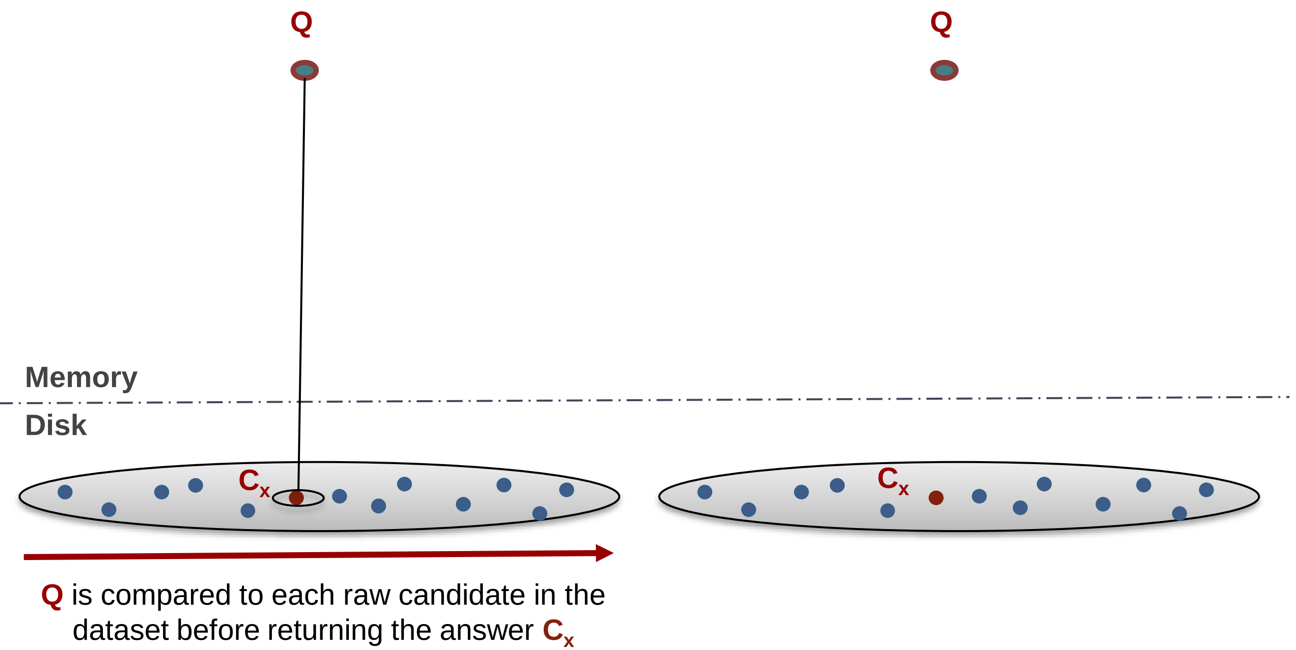


Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

Answering a similarity search query using different access paths

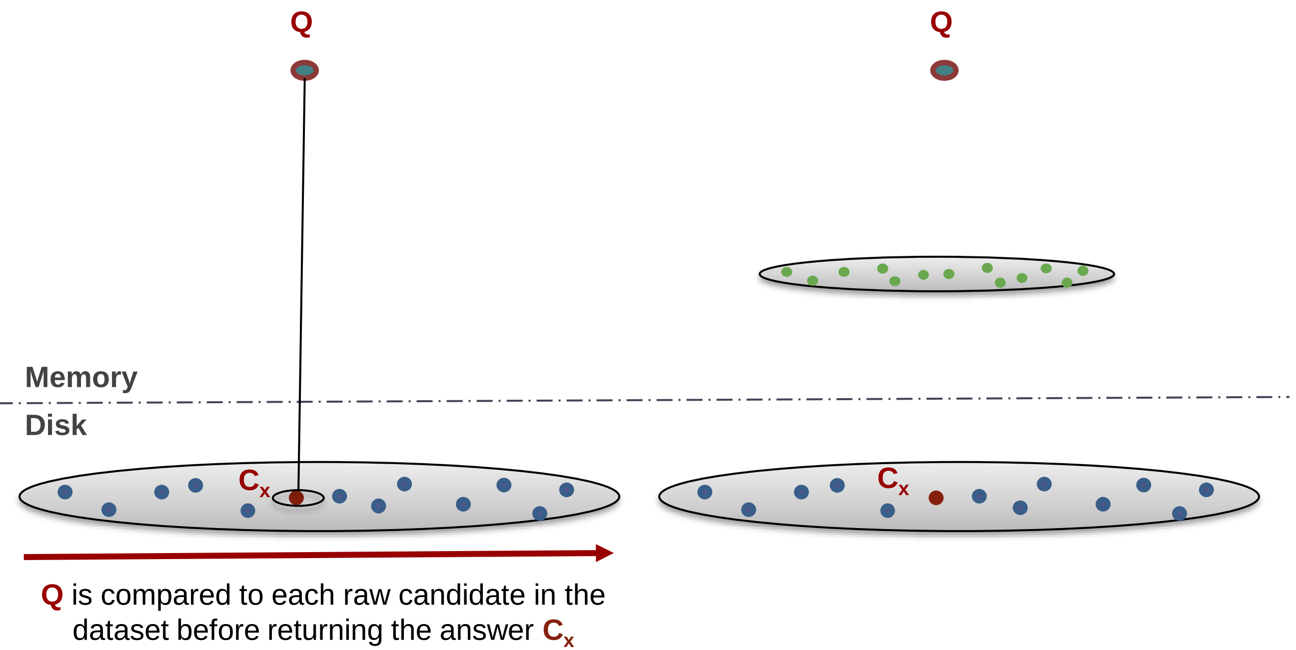
Indexes vs. Scans



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

Answering a similarity search query using different access paths

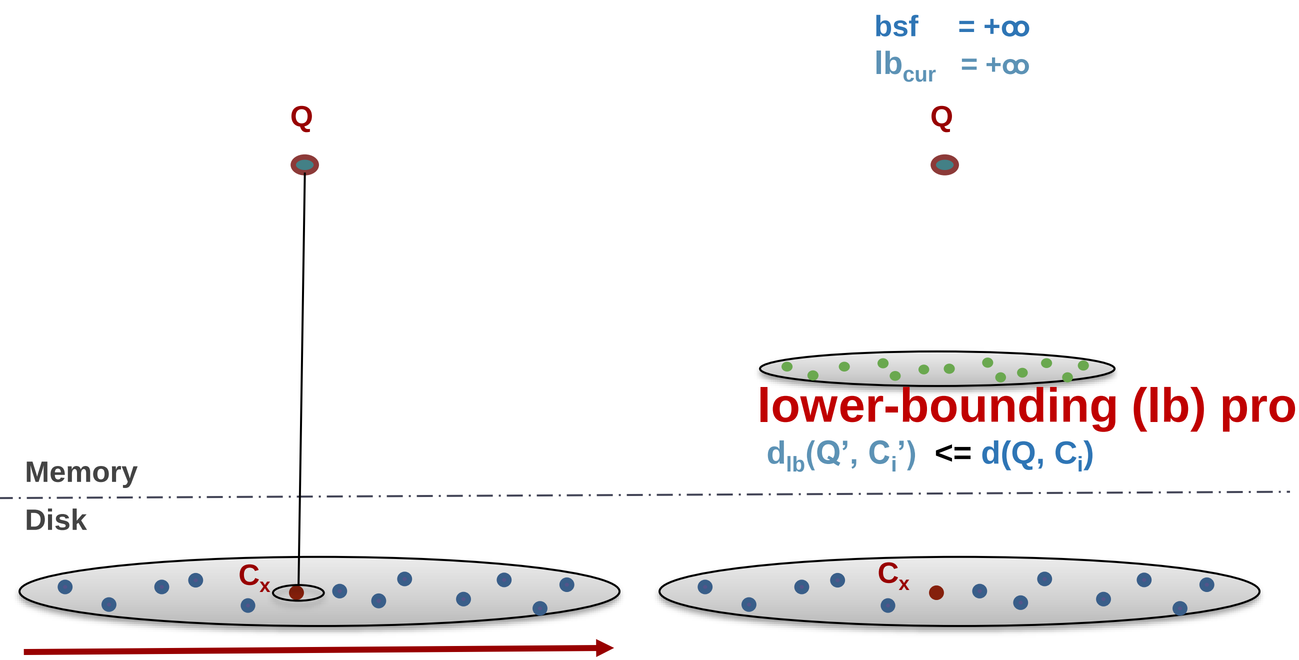
Indexes vs. Scans



(a) Serial scan

(b) Skip-sequential scan

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



lower-bounding (lb) property:

$d_{lb}(Q', C_i') <= d(Q, C_i)$

Publications

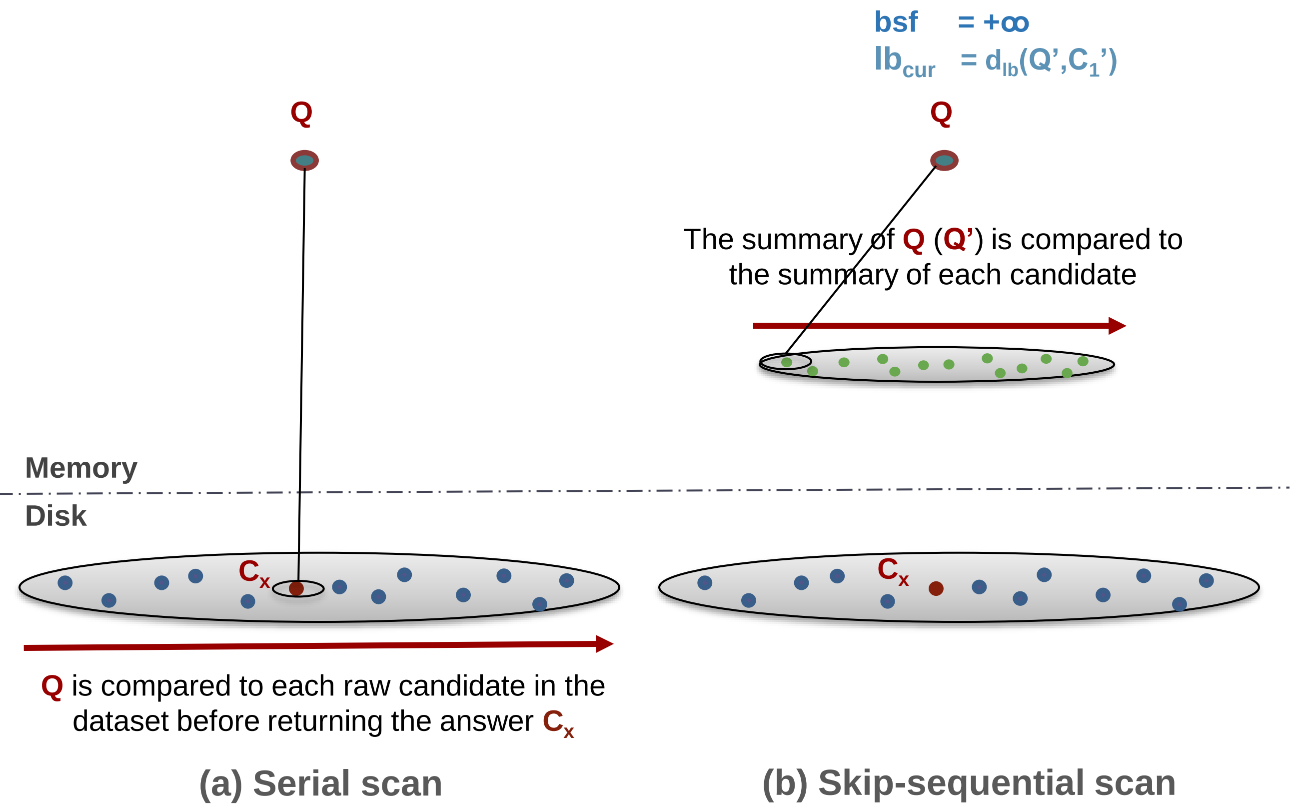
Faloutsos-SIGMOD'94

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

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

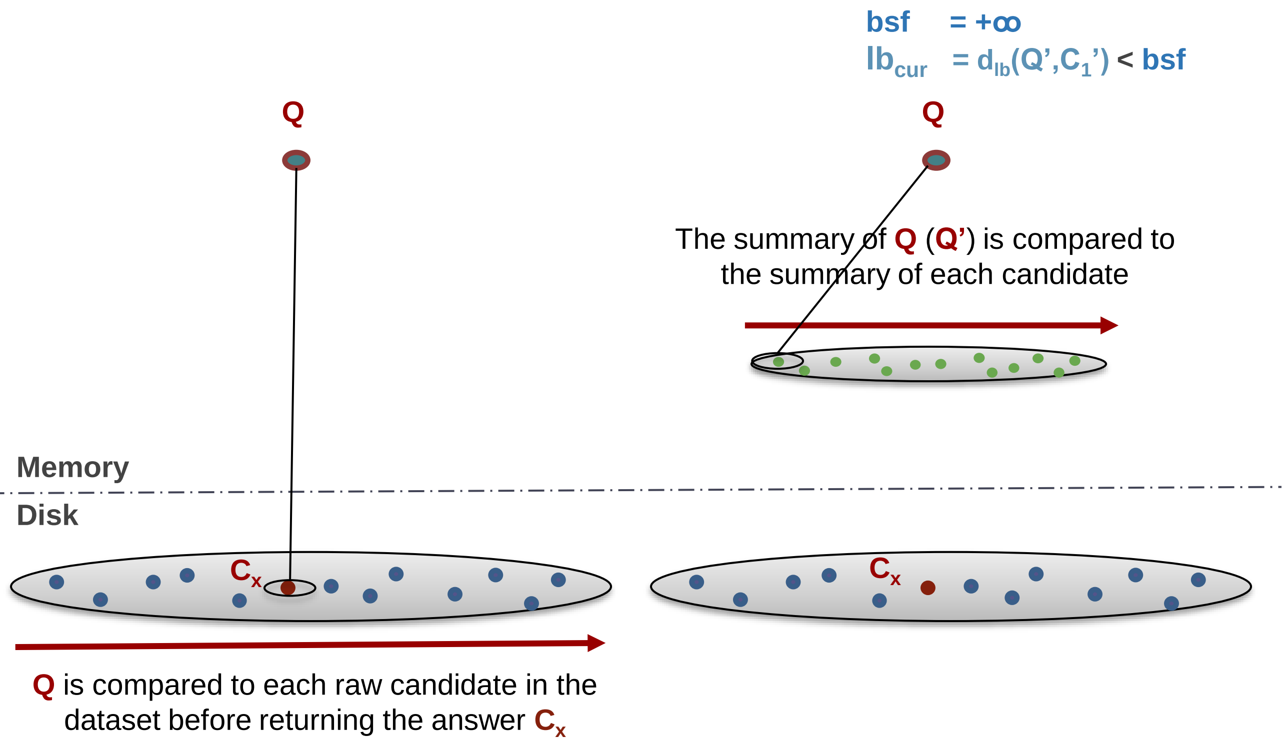
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

Indexes vs. Scans

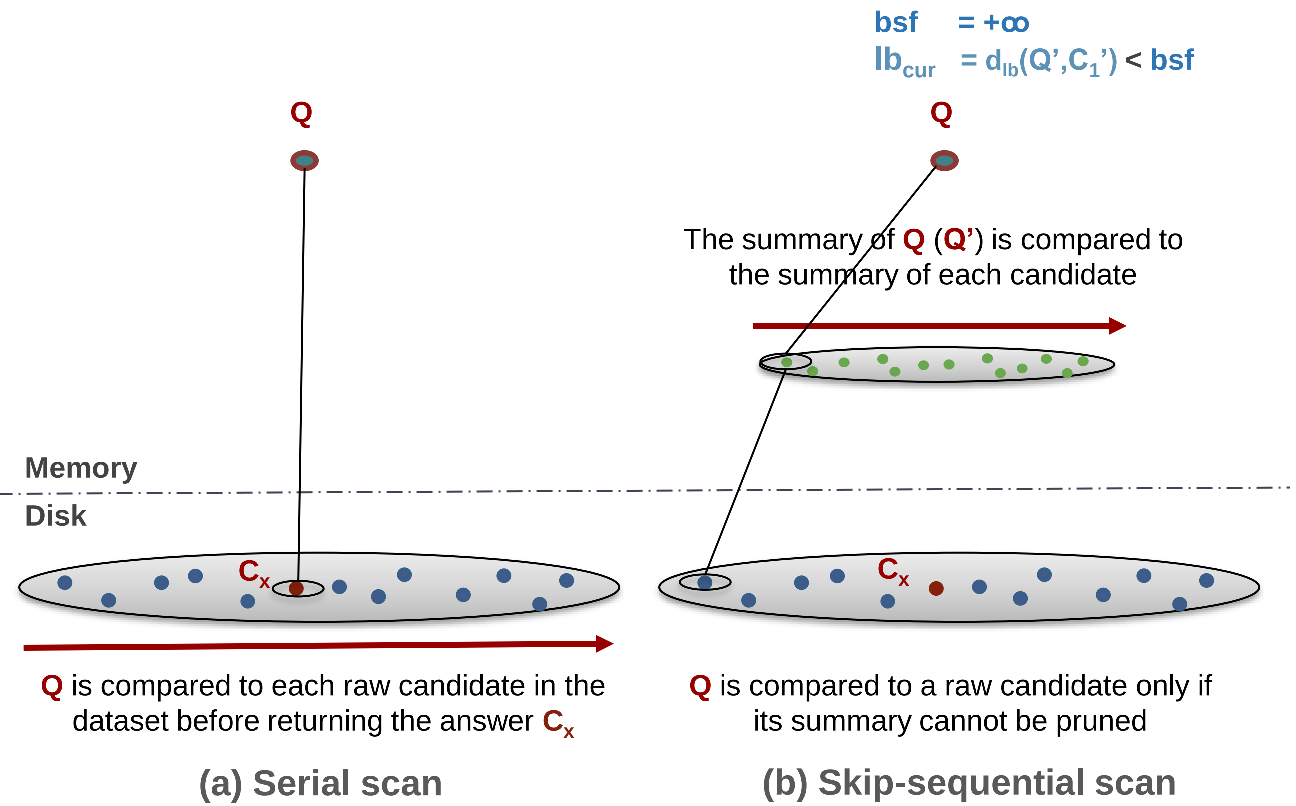


(a) Serial scan

(b) Skip-sequential scan

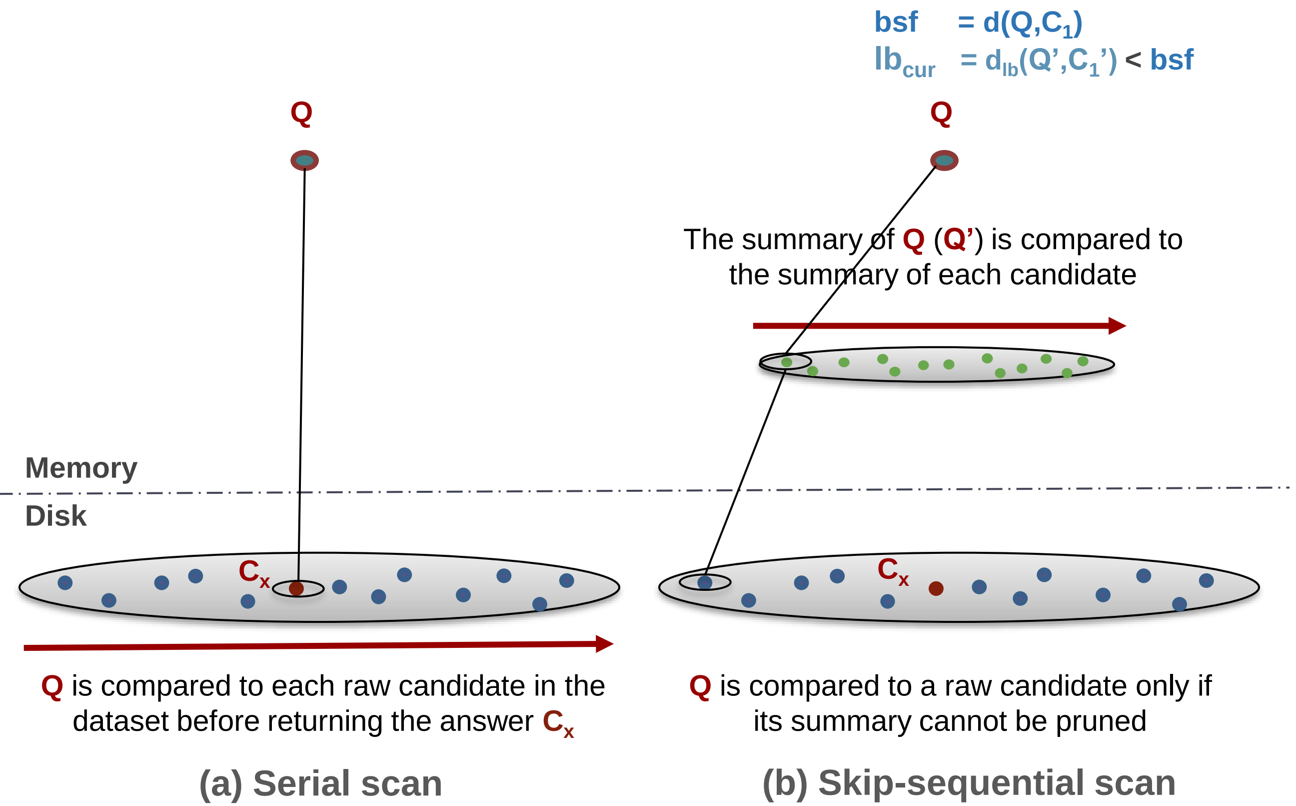
Answering a similarity search query using different access paths

Indexes vs. Scans



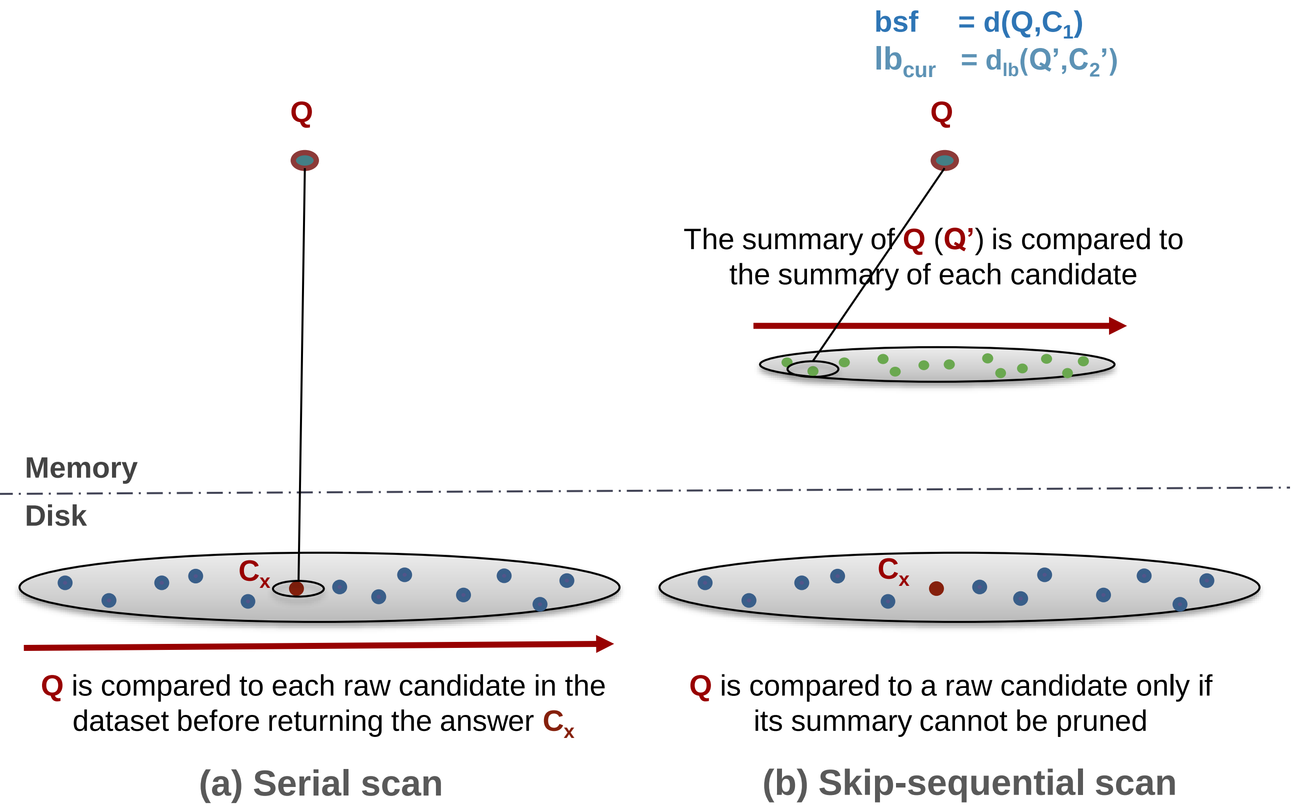
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

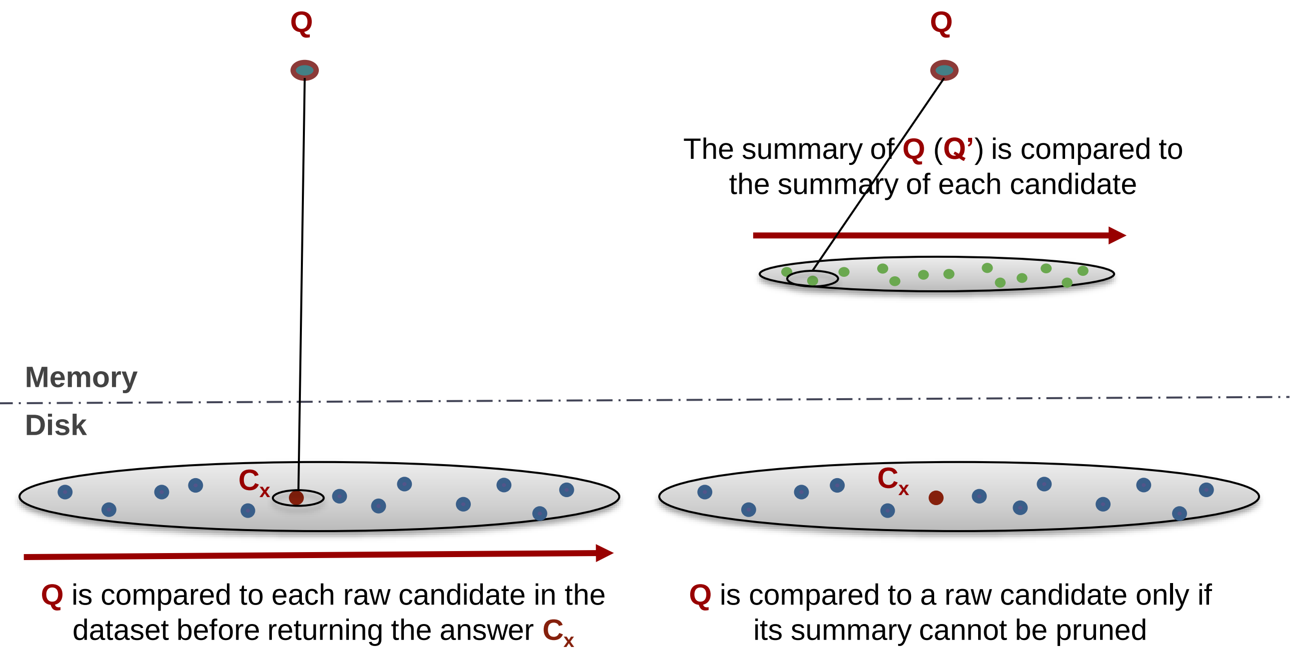
Indexes vs. Scans



Answering a similarity search query using different access paths

Indexes vs. Scans

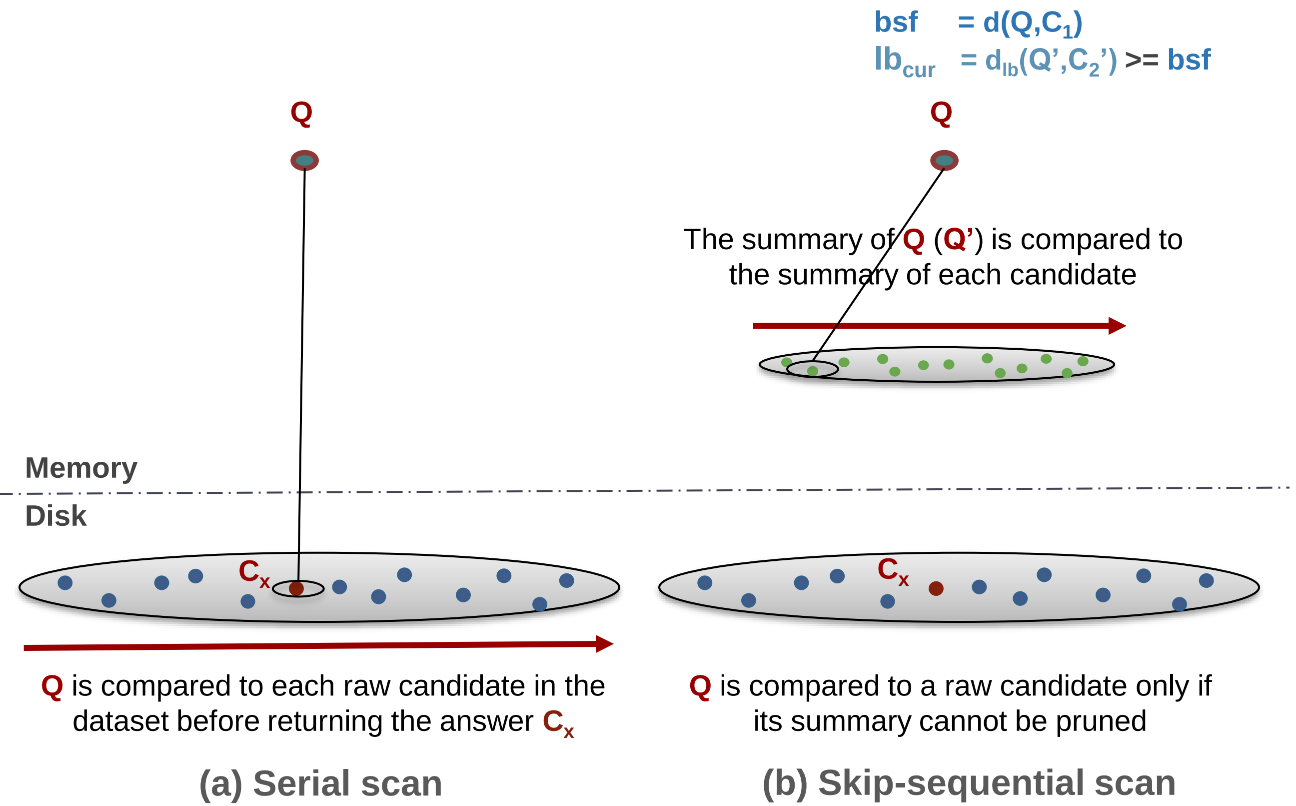
bsf $= d(Q,C_1)$
$lb_{cur} = d_{lb}(Q',C_2') >=$ bsf



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

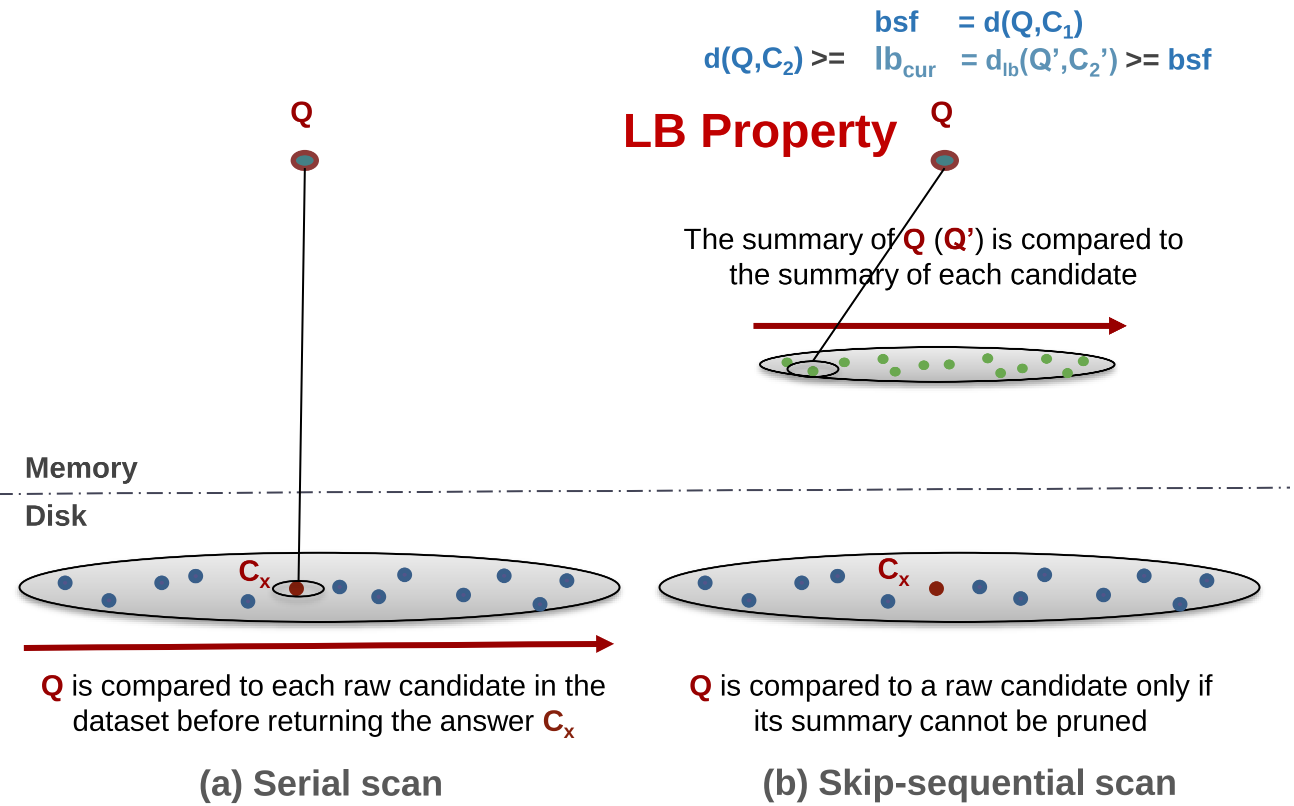
<u>**Answering a similarity search query using different access paths**</u>

Indexes vs. Scans



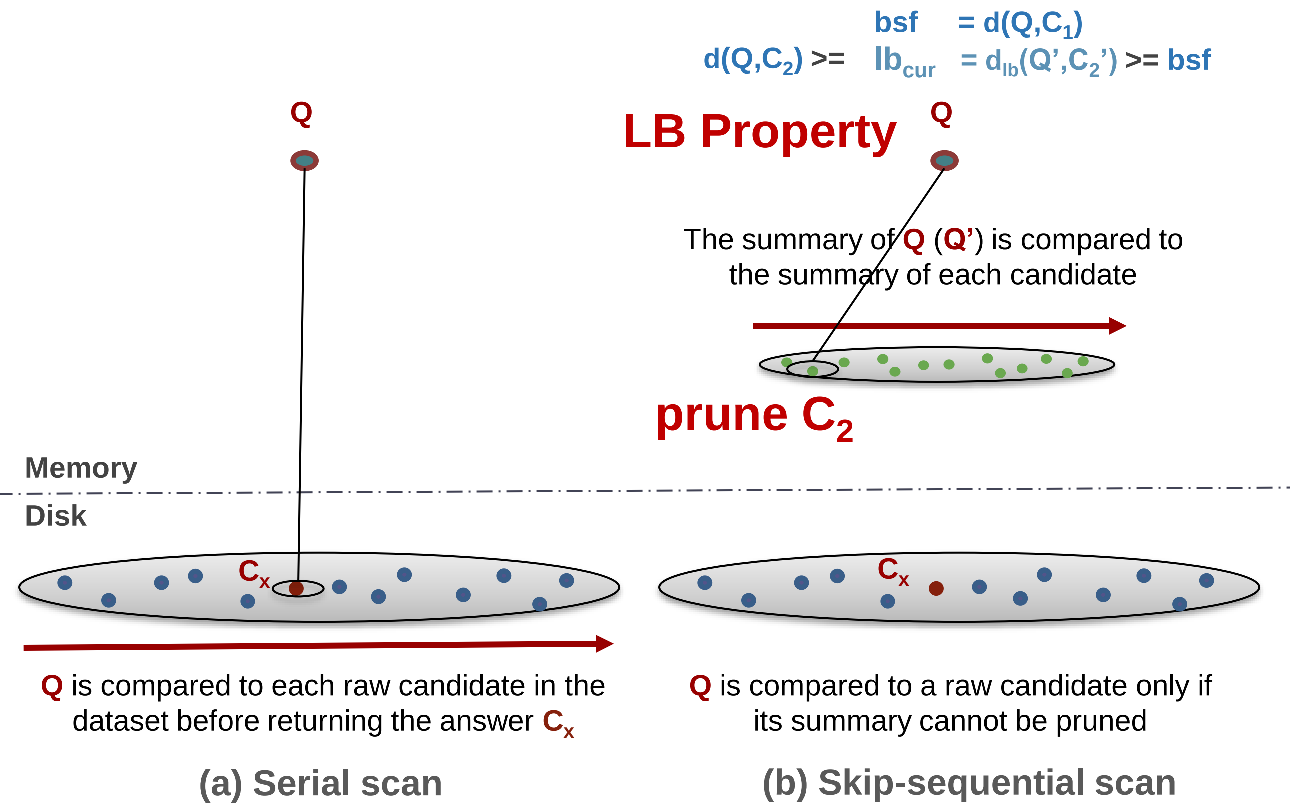
Answering a similarity search query using different access paths

Indexes vs. Scans



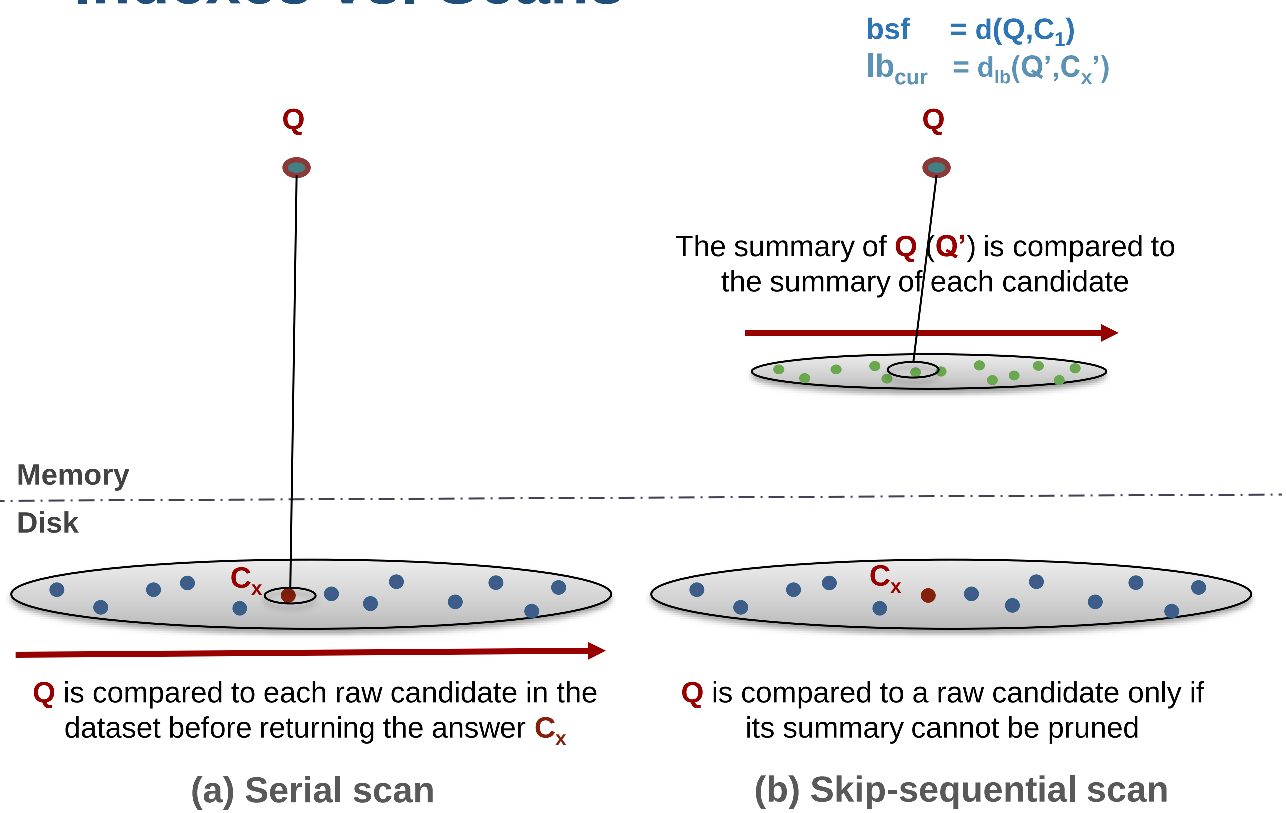
Answering a similarity search query using different access paths

Indexes vs. Scans



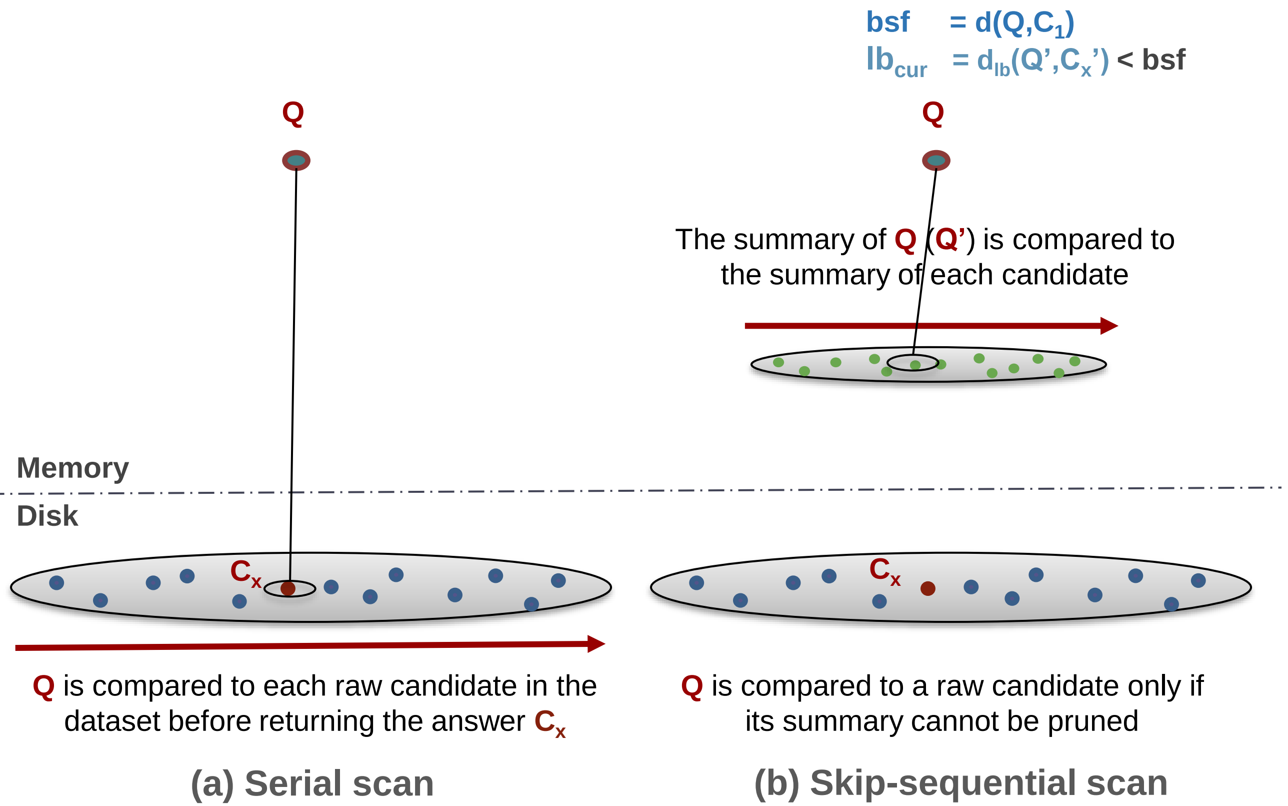
<u>**Answering a similarity search query using different access paths**</u>

Indexes vs. Scans



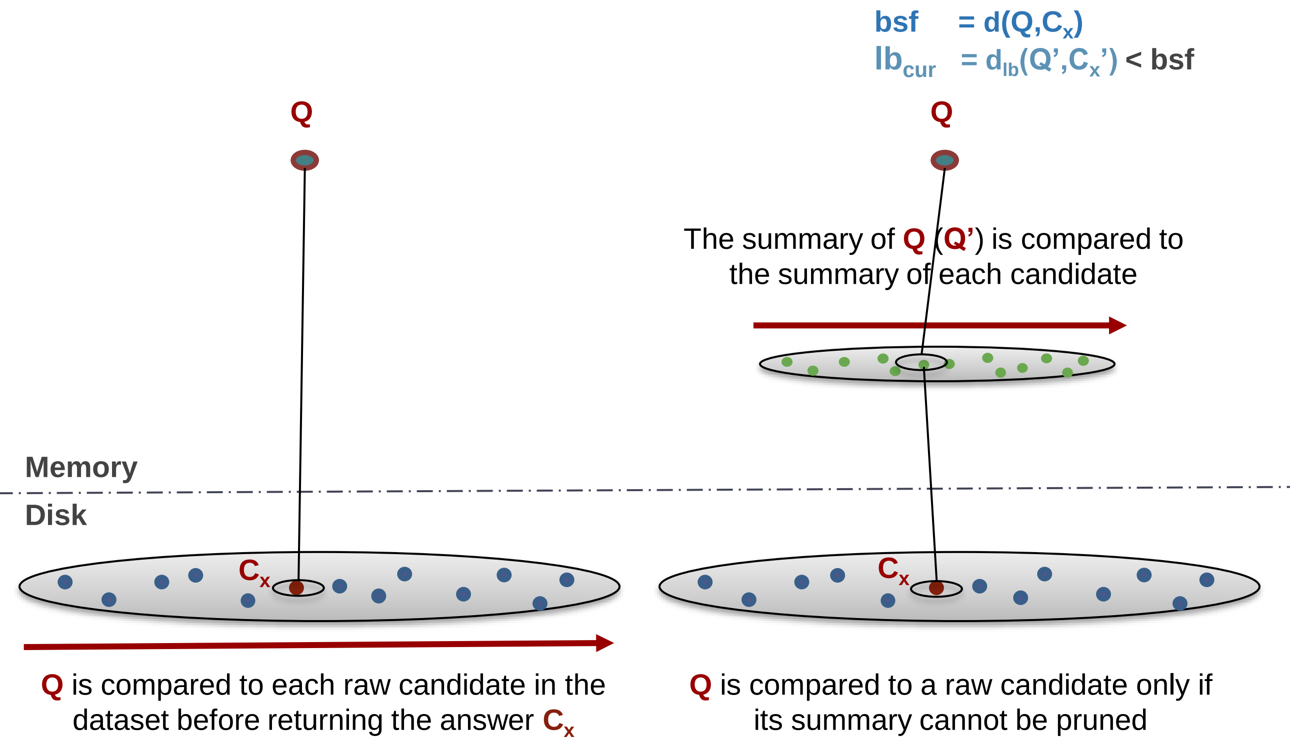
Answering a similarity search query using different access paths

Indexes vs. Scans



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

Indexes vs. Scans



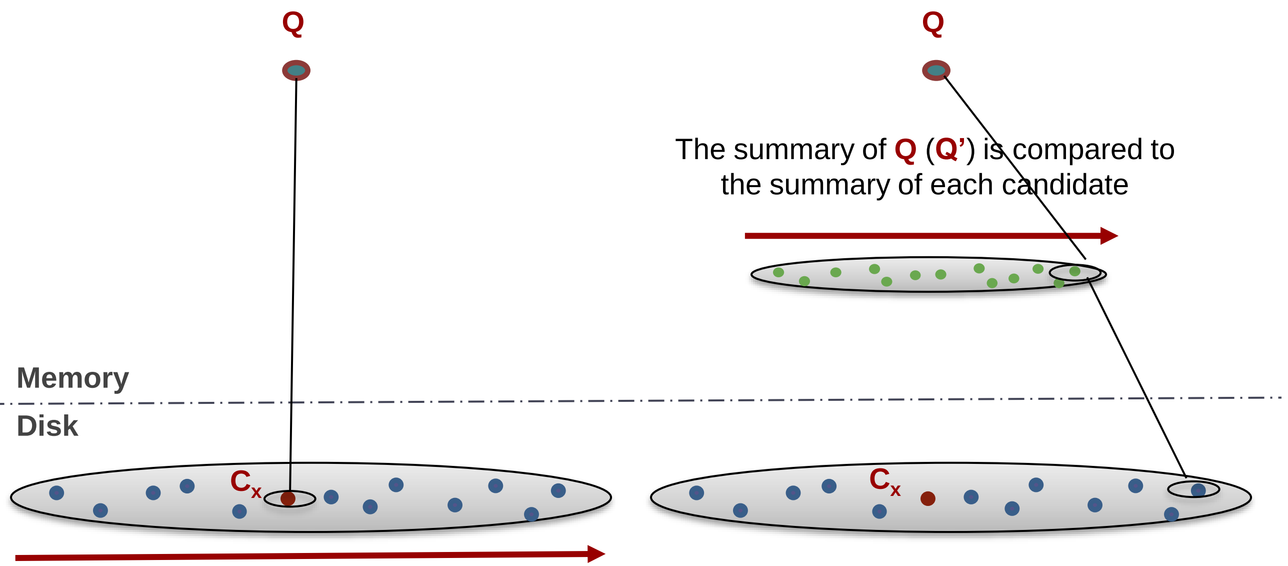
(a) Serial scan

(b) Skip-sequential scan

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

Indexes vs. Scans

bsf = $d(Q,C_x)$
lb_{cur} = $d_{lb}(Q',C_n')$ < bsf



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

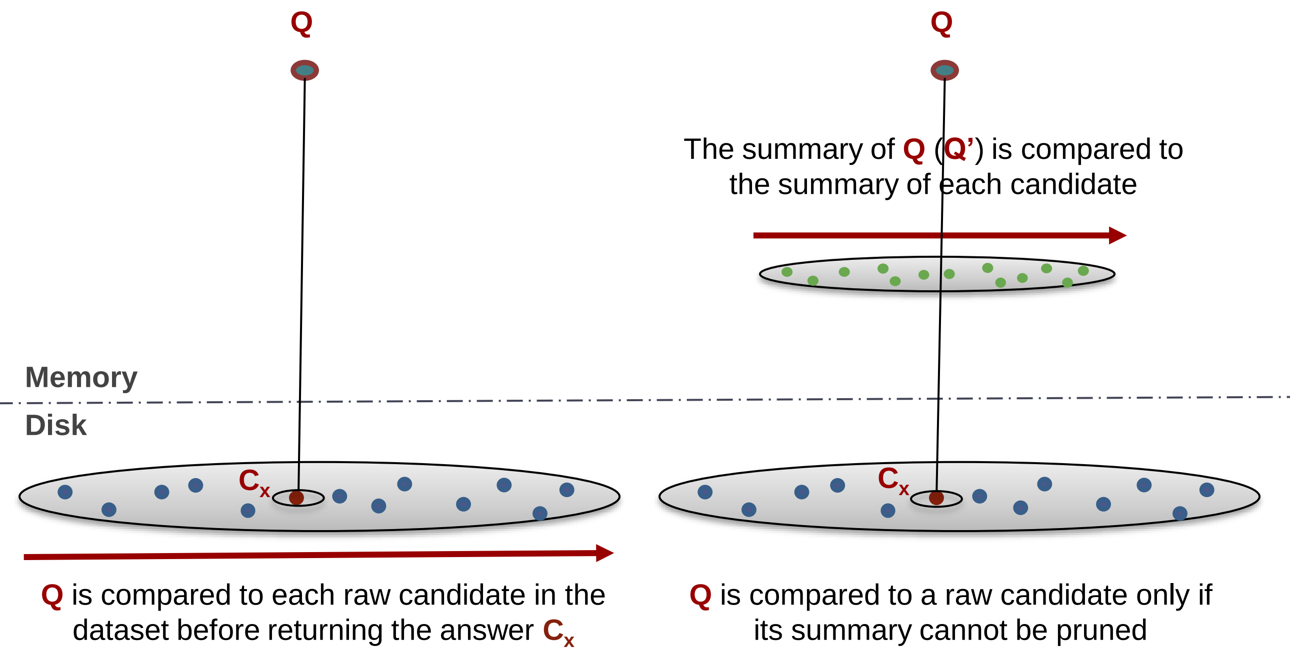
Q is compared to a raw candidate only if its summary cannot be pruned

(a) Serial scan

(b) Skip-sequential scan

Answering a similarity search query using different access paths

Indexes vs. Scans

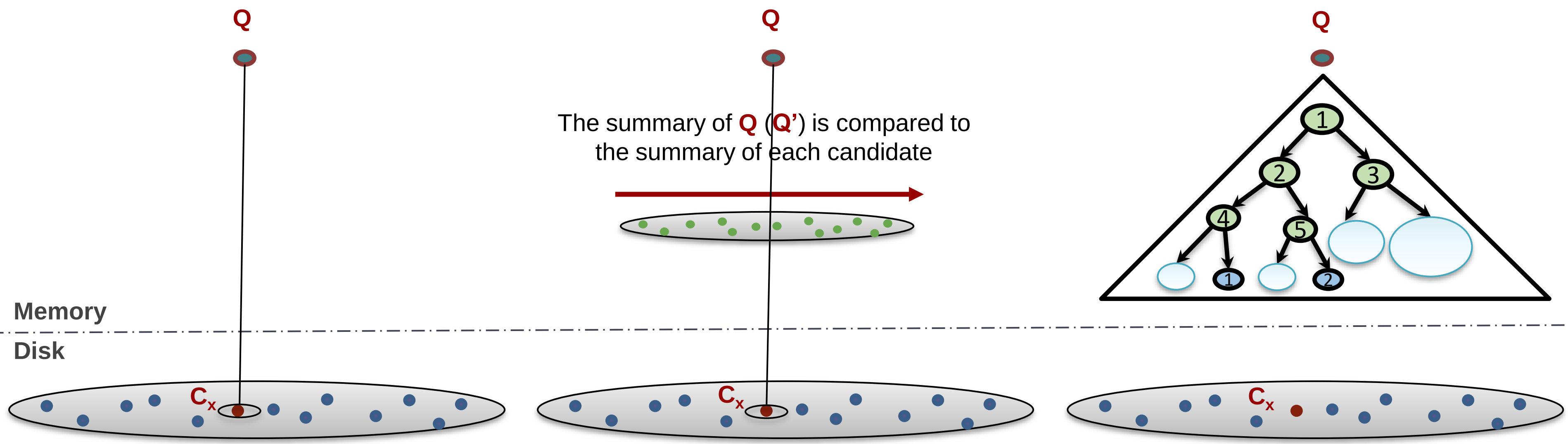


(a) Serial scan

(b) Skip-sequential scan

Answering a similarity search query using different access paths

Indexes vs. Scans



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

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

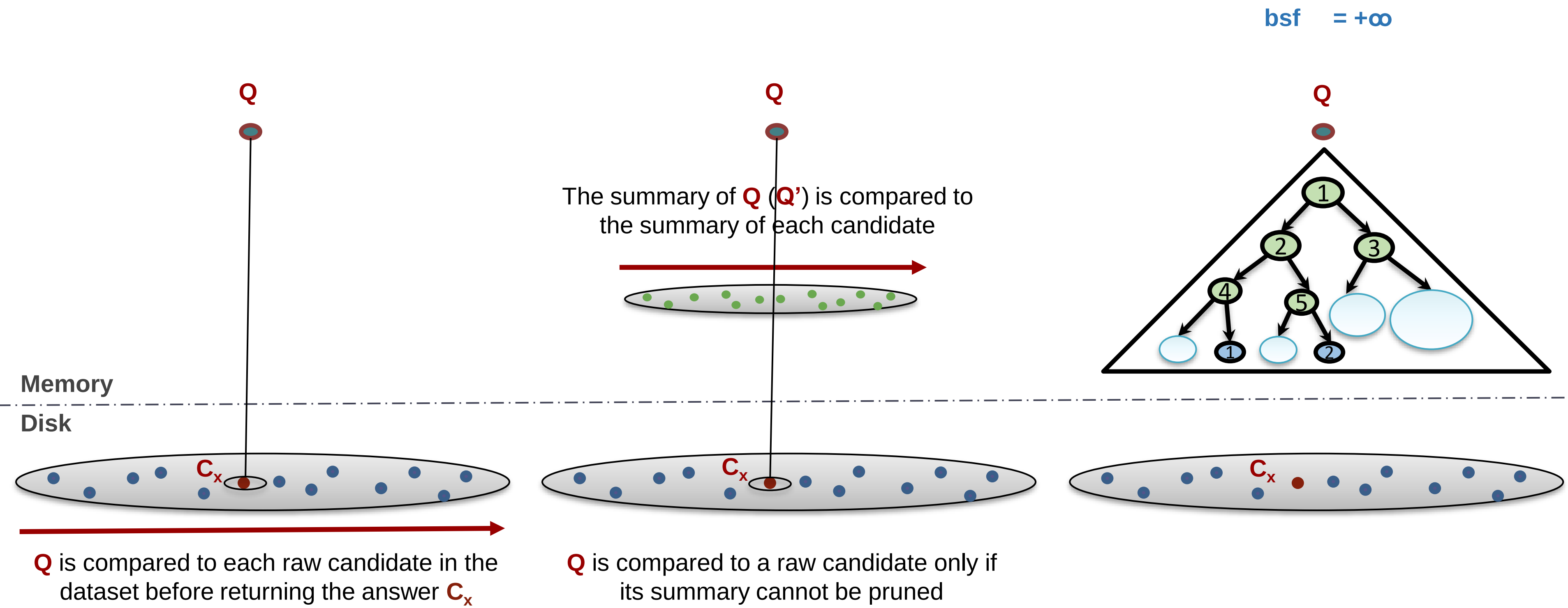
(a) Serial scan

(b) Skip-sequential scan

(c) Tree-based index

Answering a similarity search query using different access paths

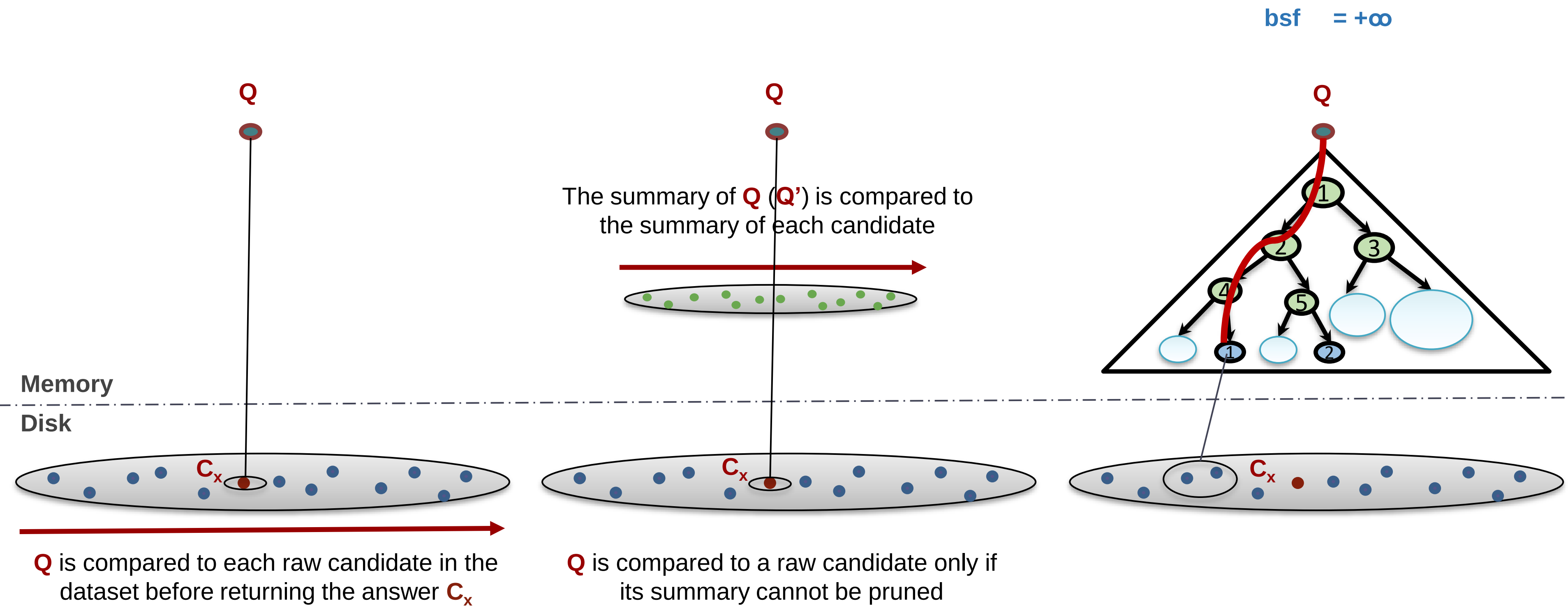
Indexes vs. Scans



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

Answering a similarity search query using different access paths

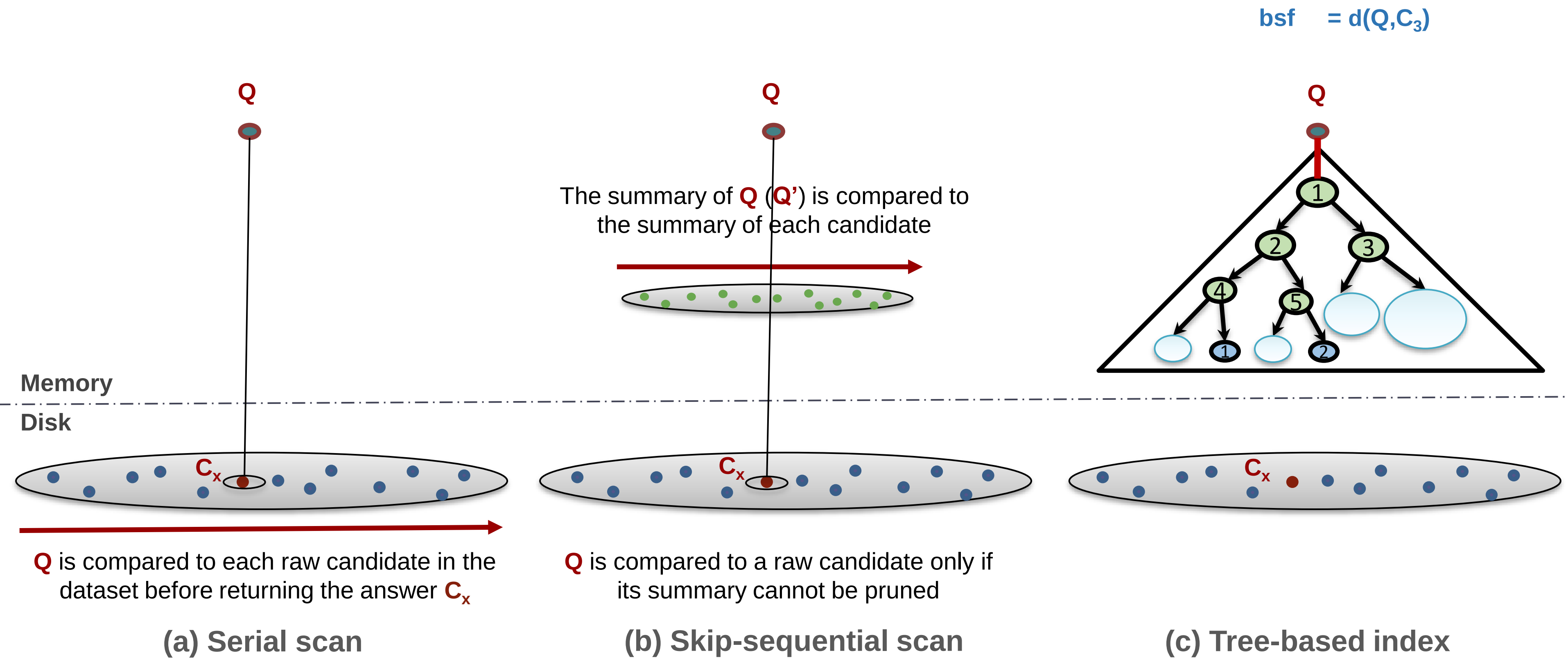
Indexes vs. Scans



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

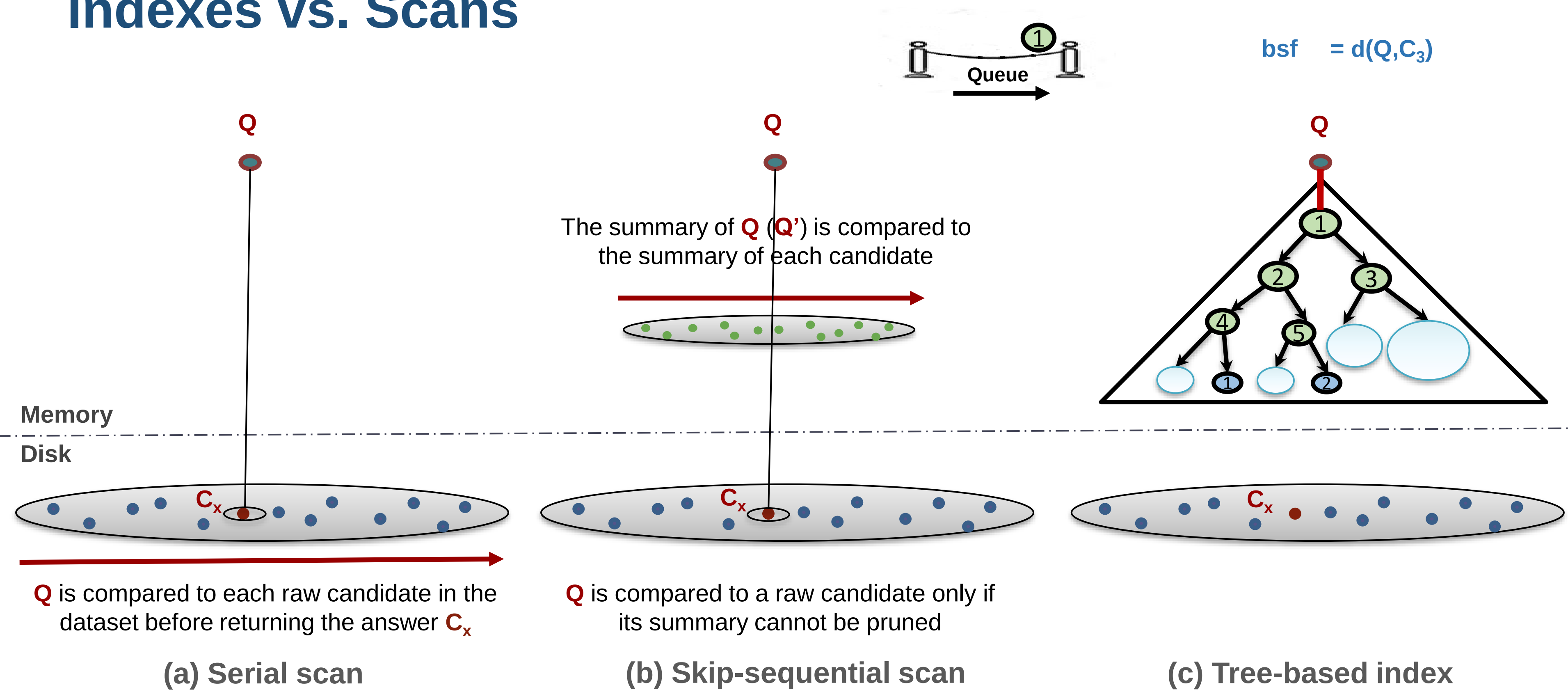
Answering a similarity search query using different access paths

Indexes vs. Scans



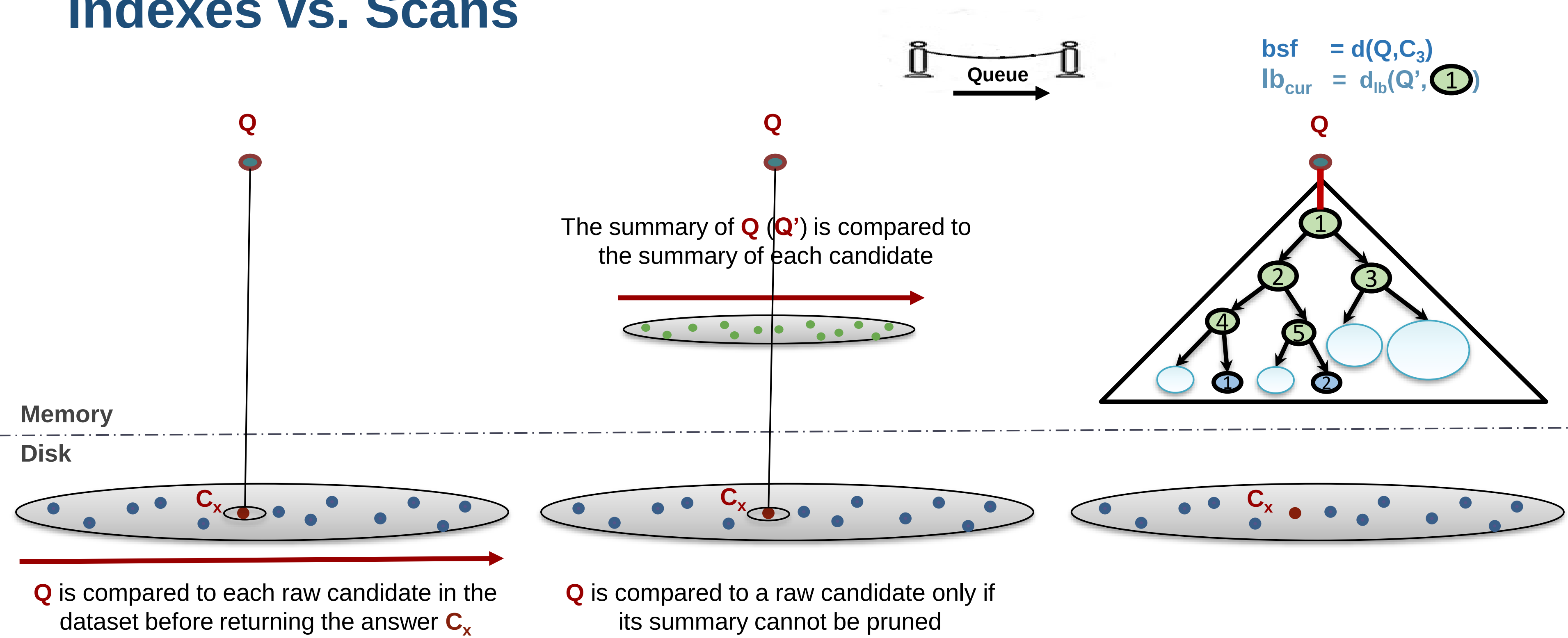
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

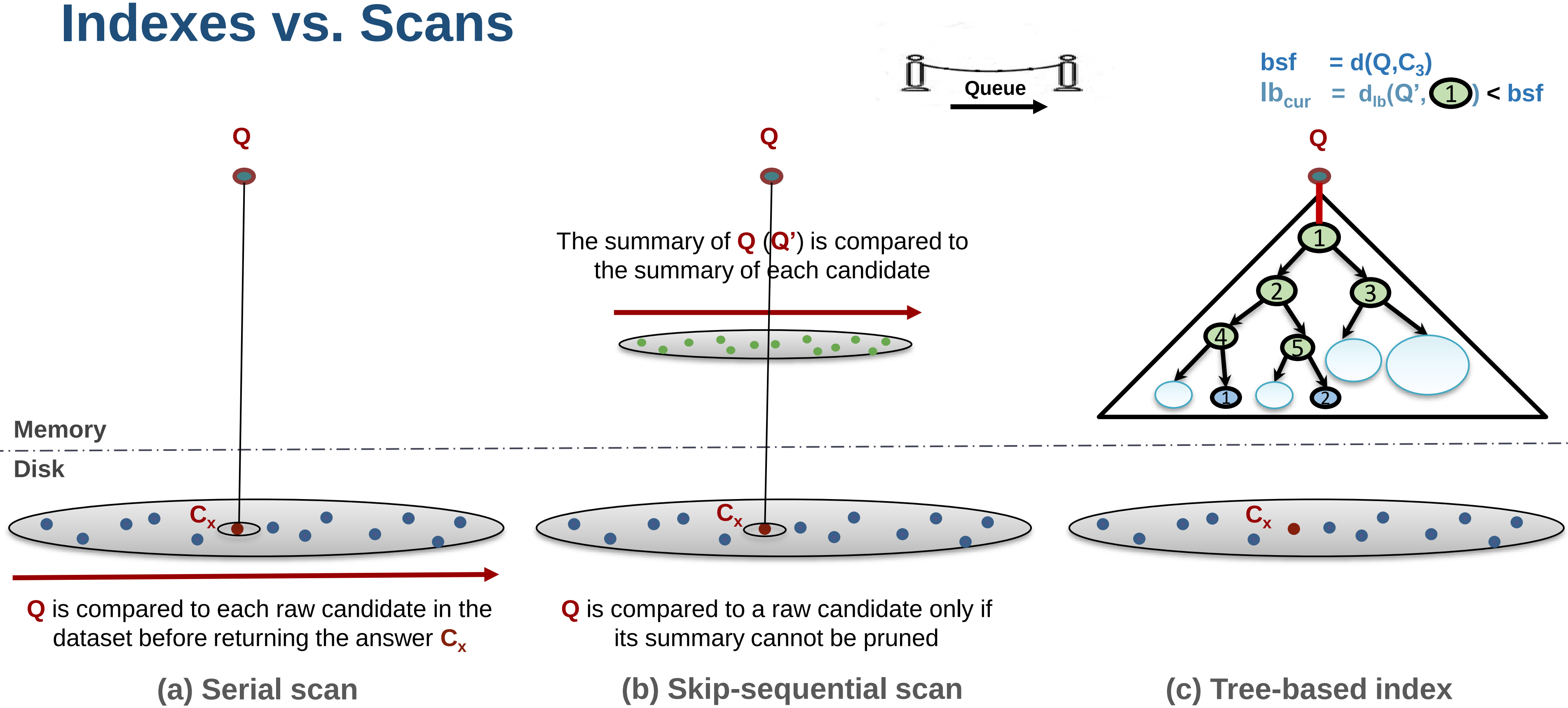
Indexes vs. Scans



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

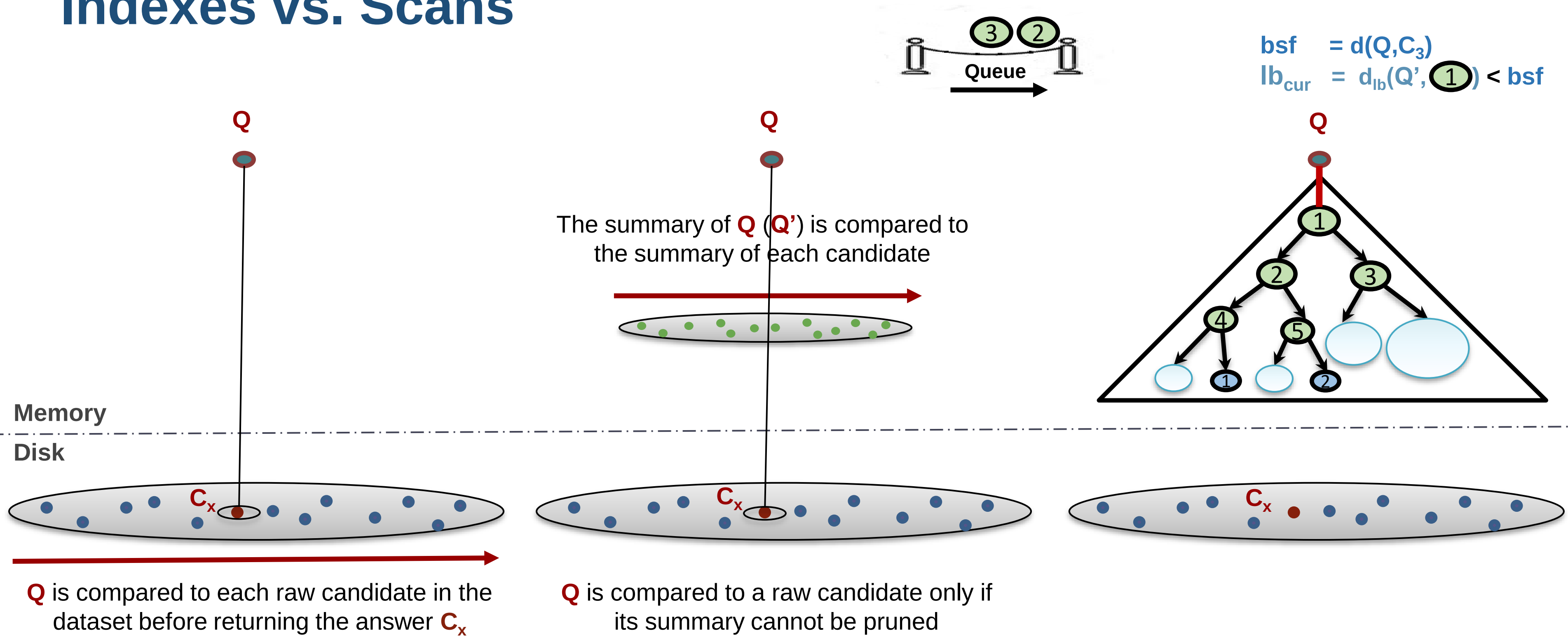
Answering a similarity search query using different access paths

Indexes vs. Scans



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

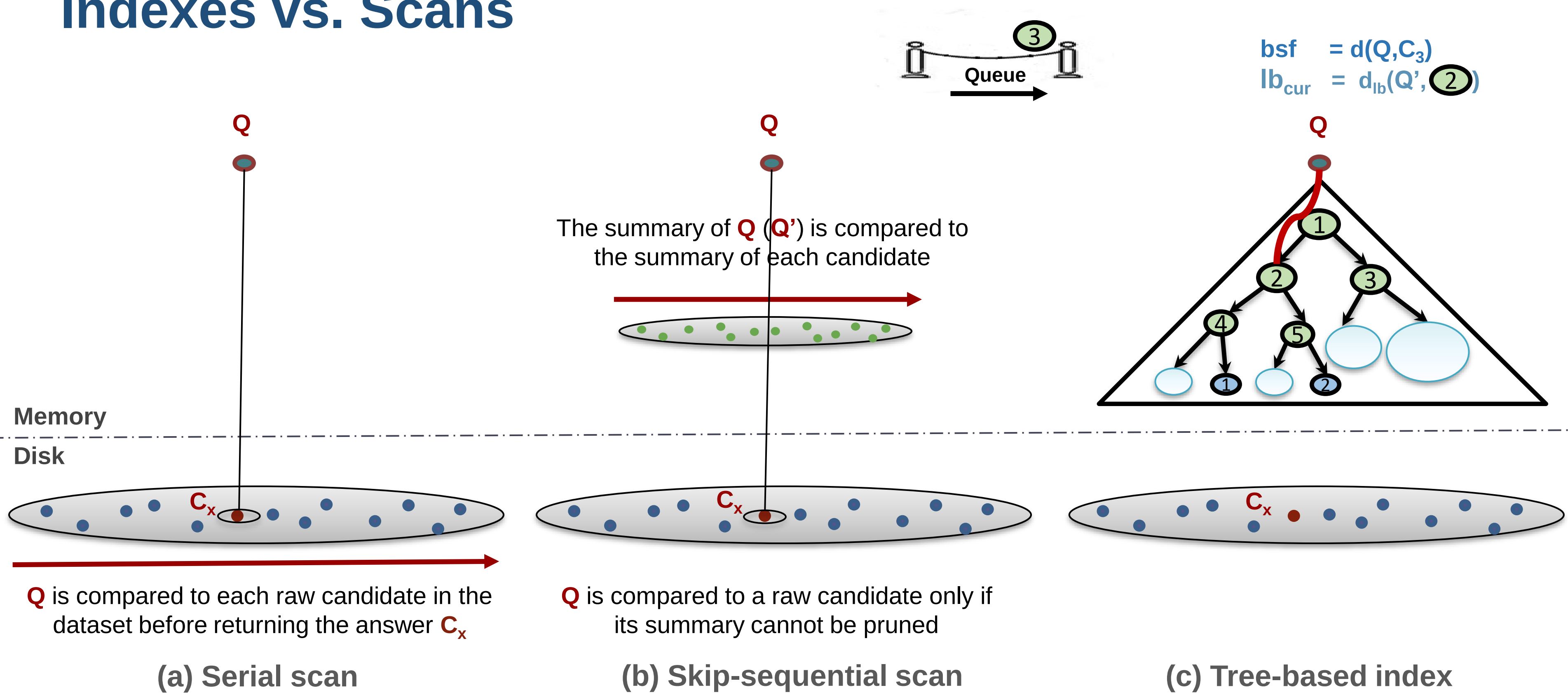
Indexes vs. Scans



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

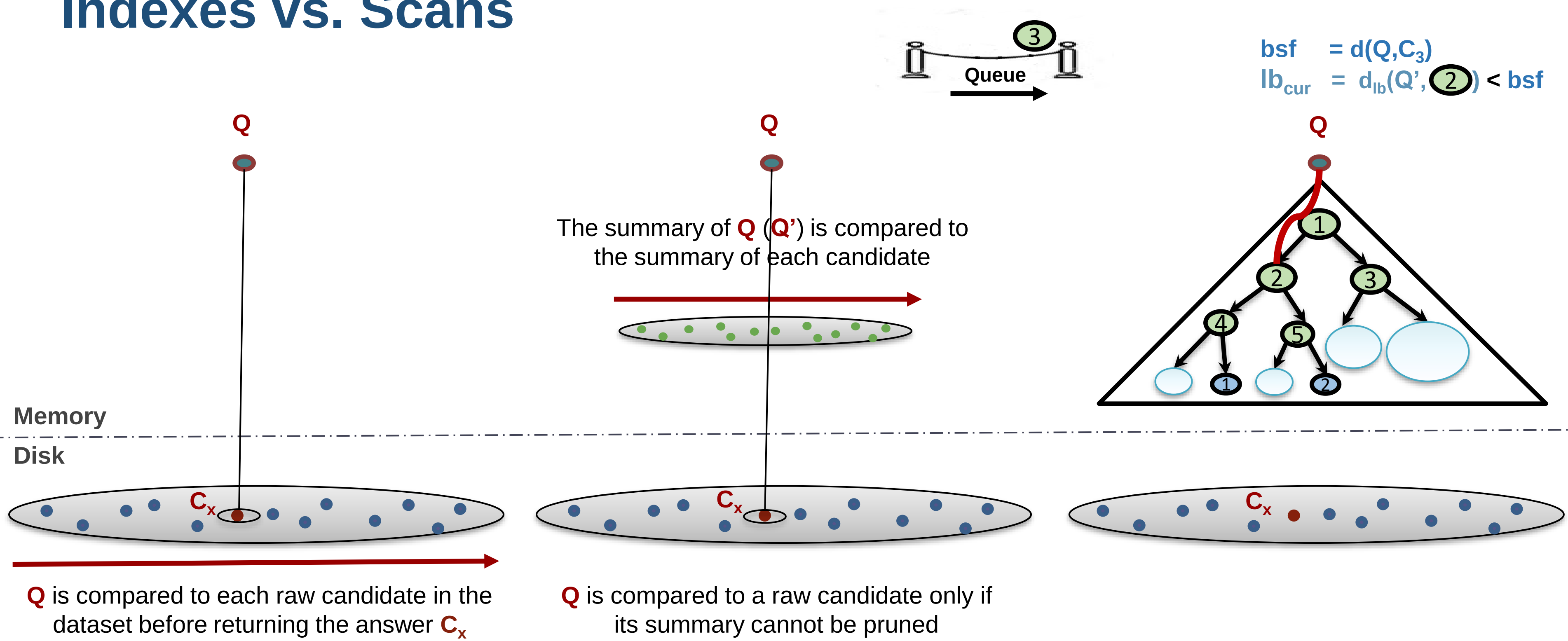
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

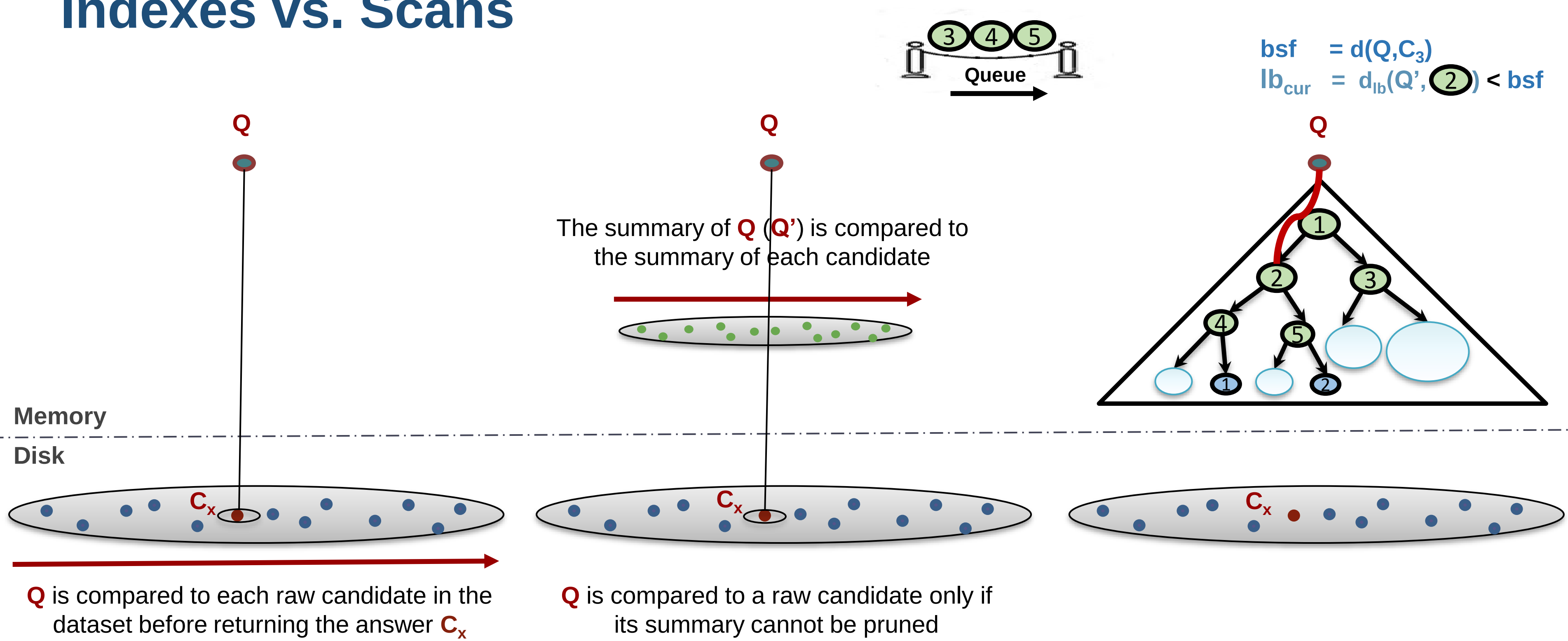
Indexes vs. Scans



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

Answering a similarity search query using different access paths

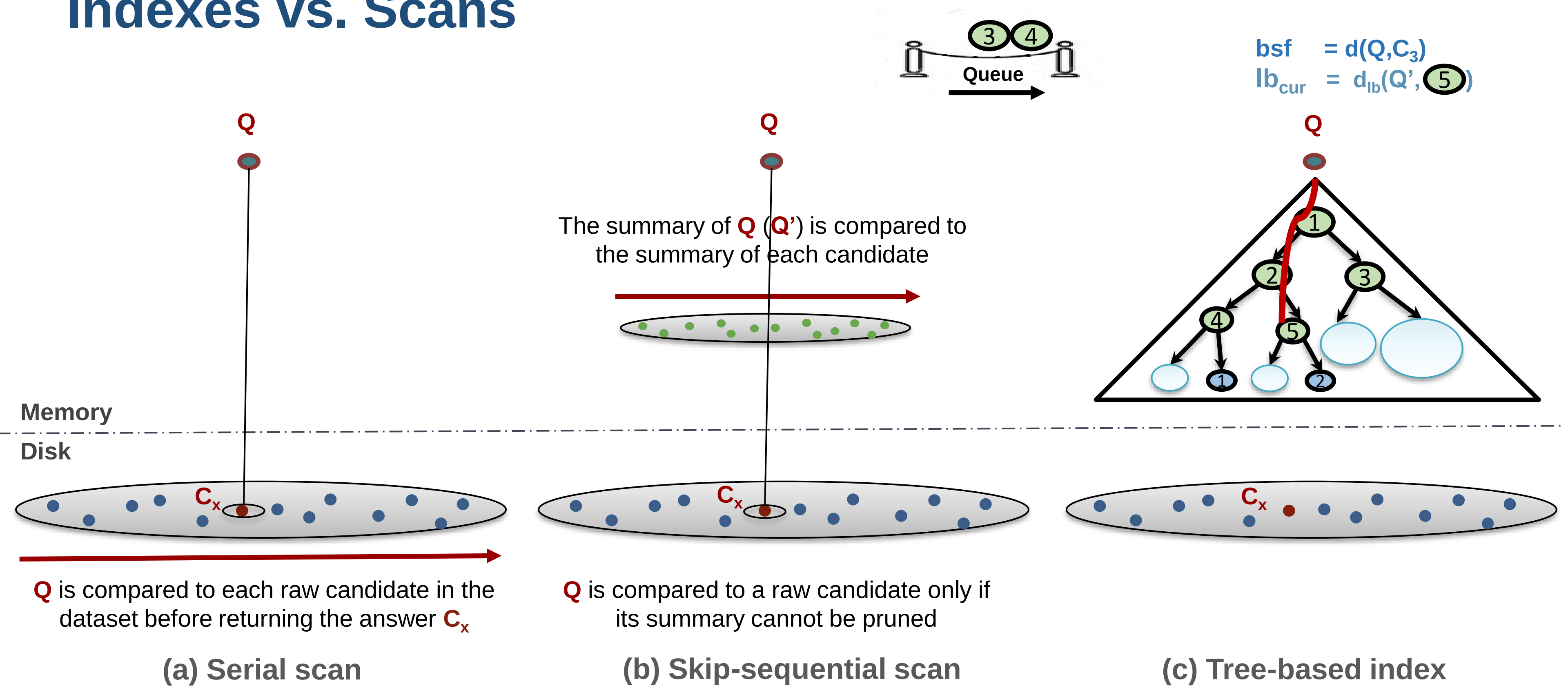
Indexes vs. Scans



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

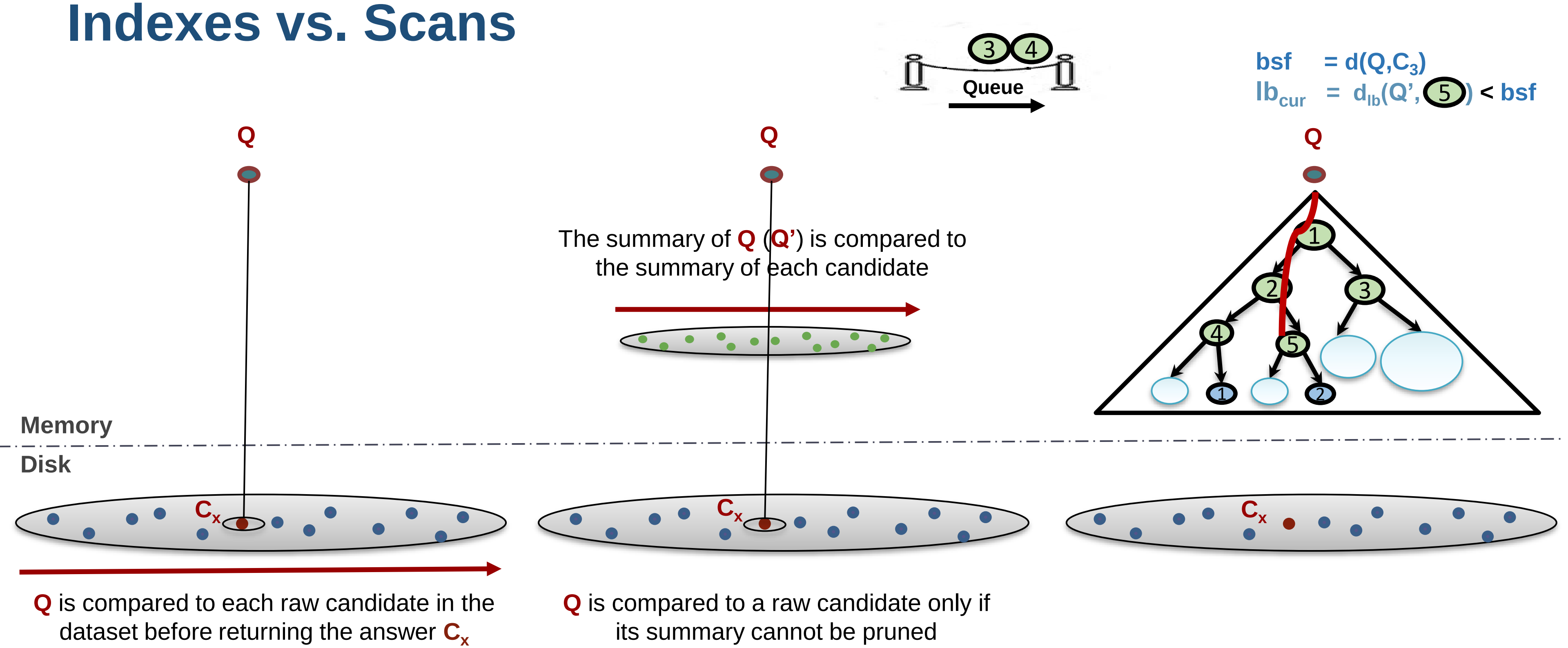
Answering a similarity search query using different access paths

Indexes vs. Scans



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

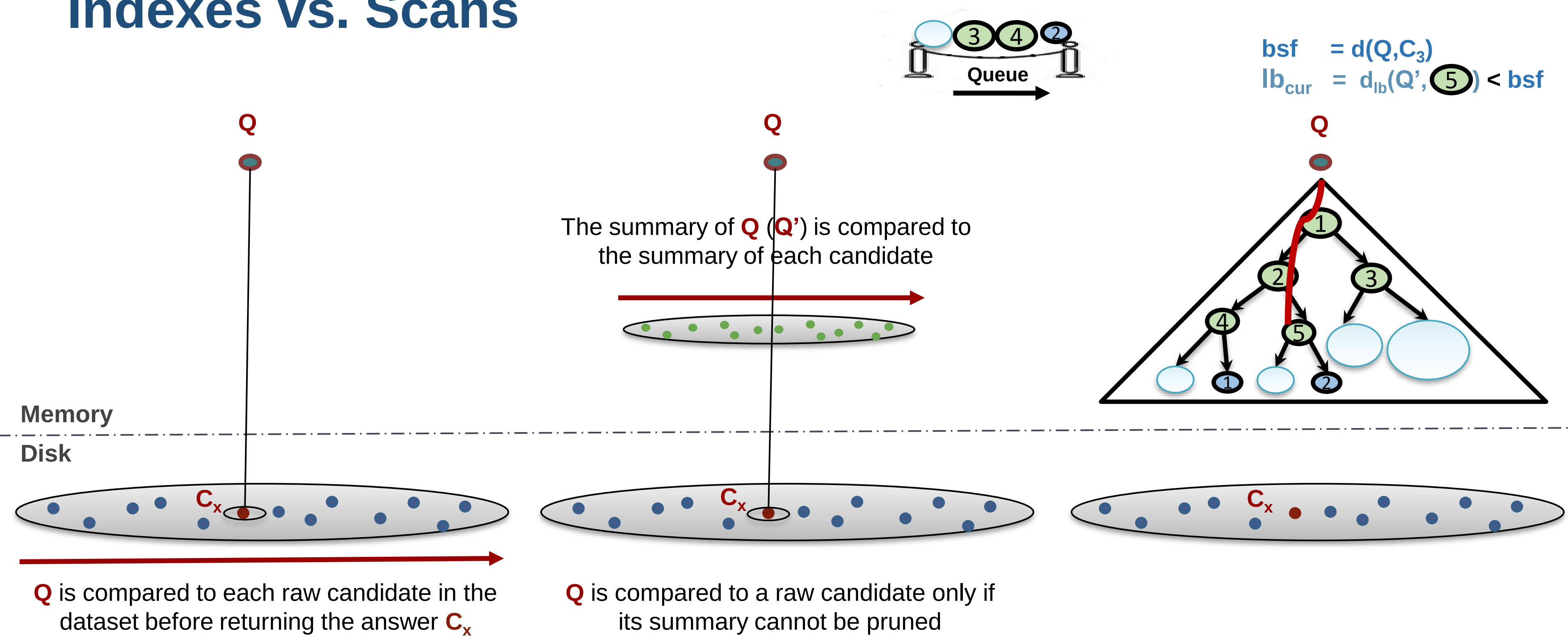
Indexes vs. Scans



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

Answering a similarity search query using different access paths

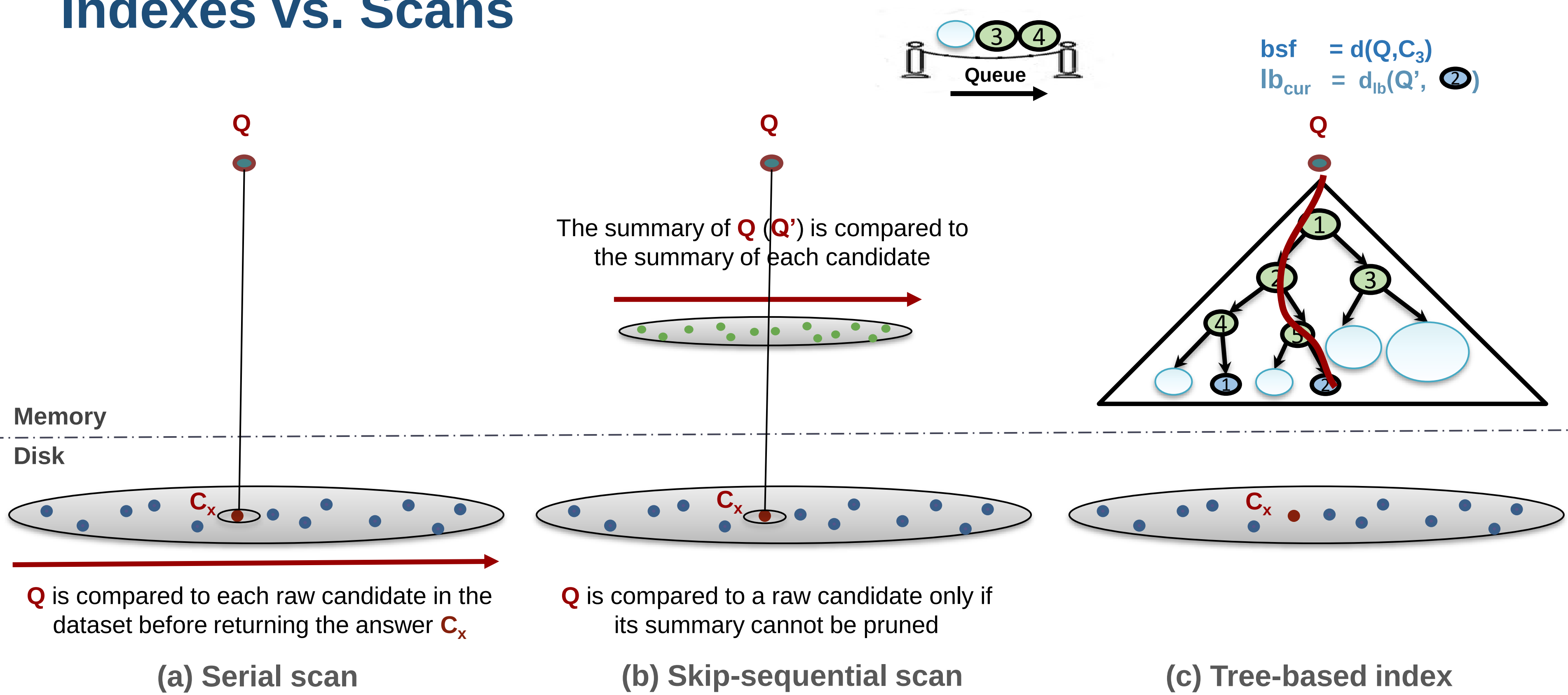
Indexes vs. Scans



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

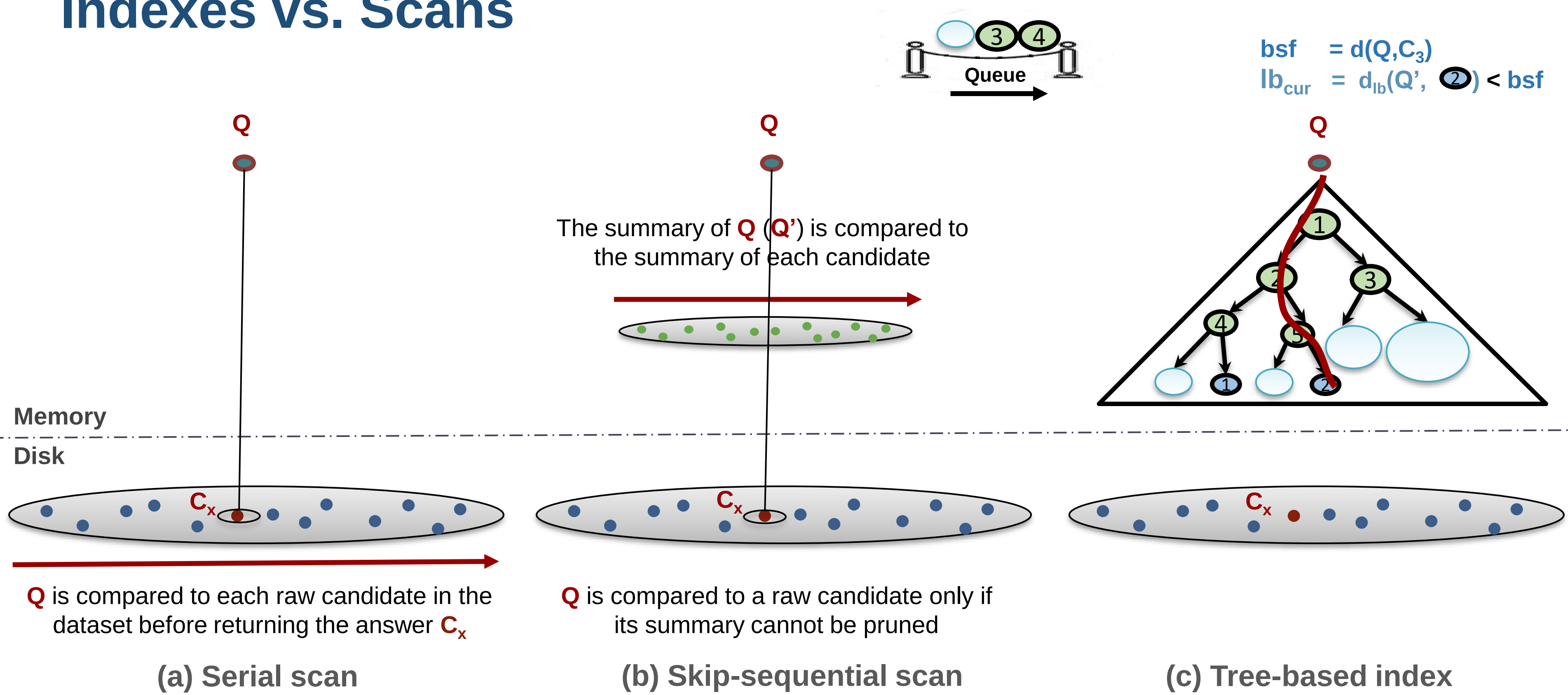
<u>Answering a similarity search query using different access paths</u>

Indexes vs. Scans



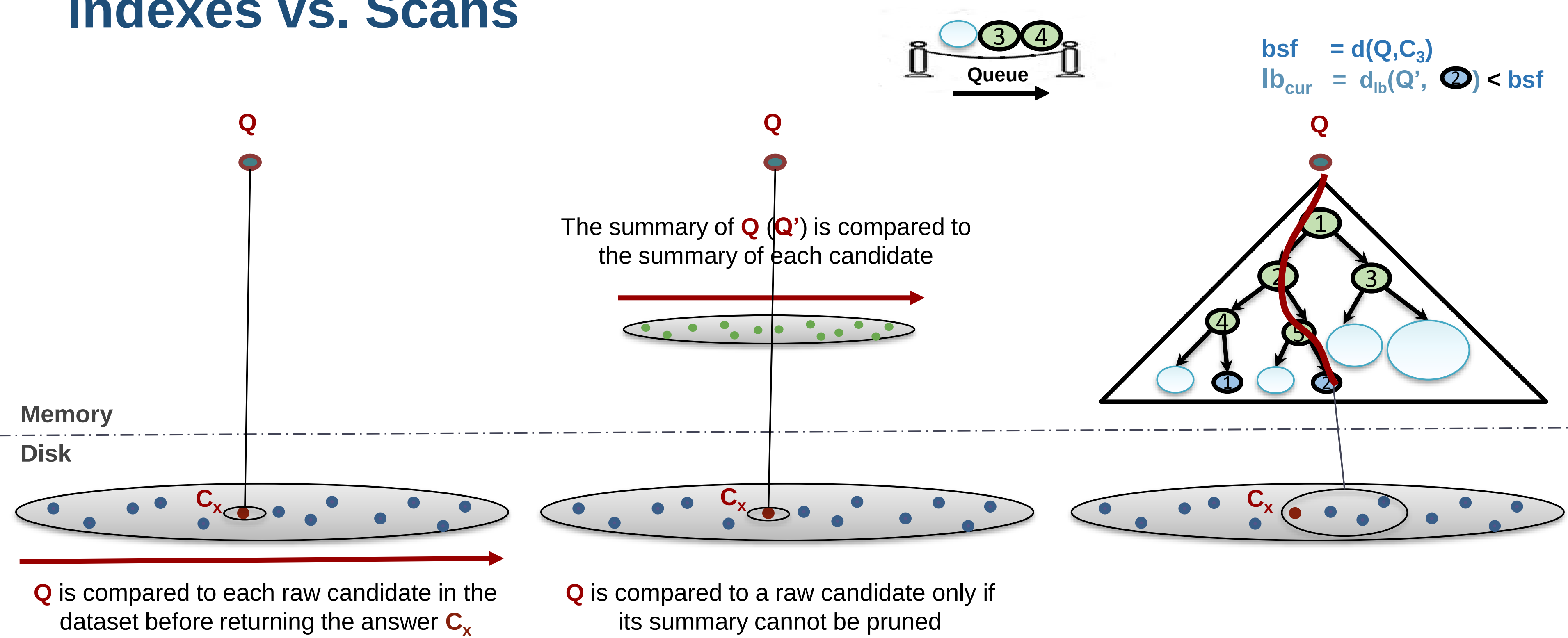
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

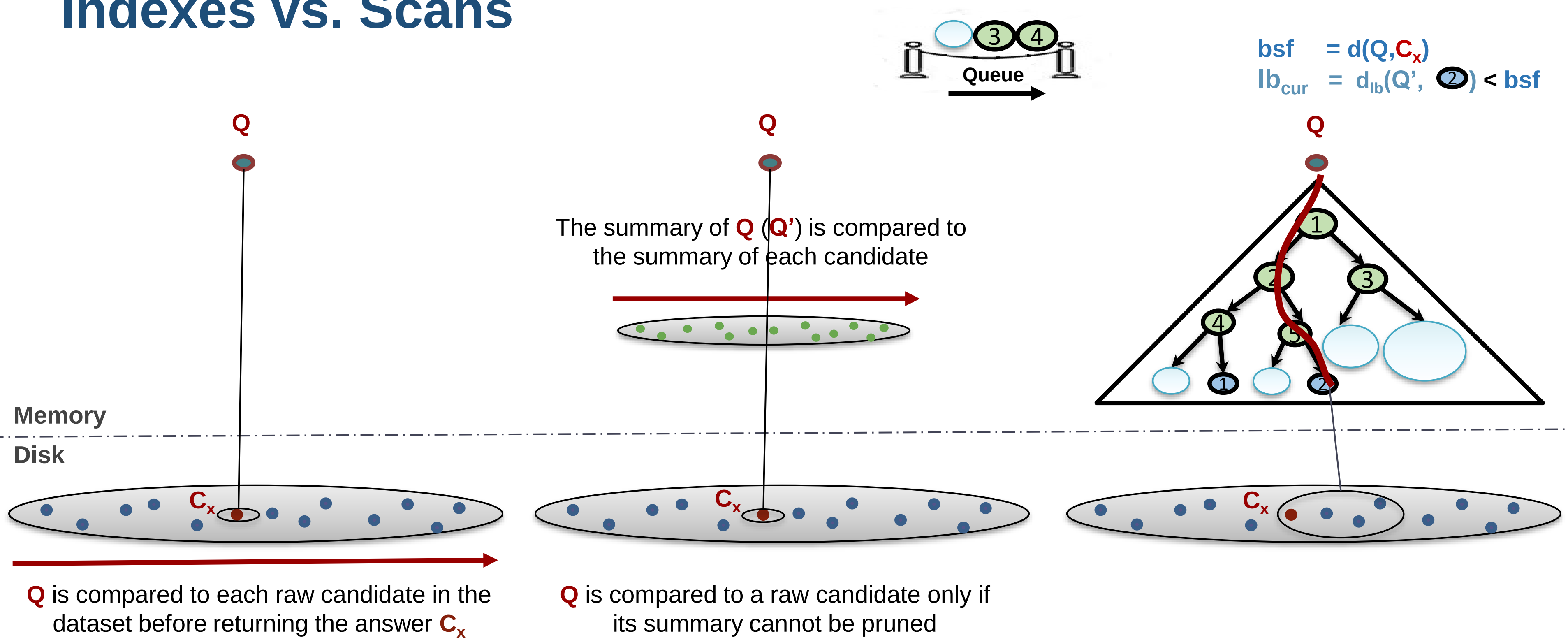
Indexes vs. Scans



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

Answering a similarity search query using different access paths

Indexes vs. Scans



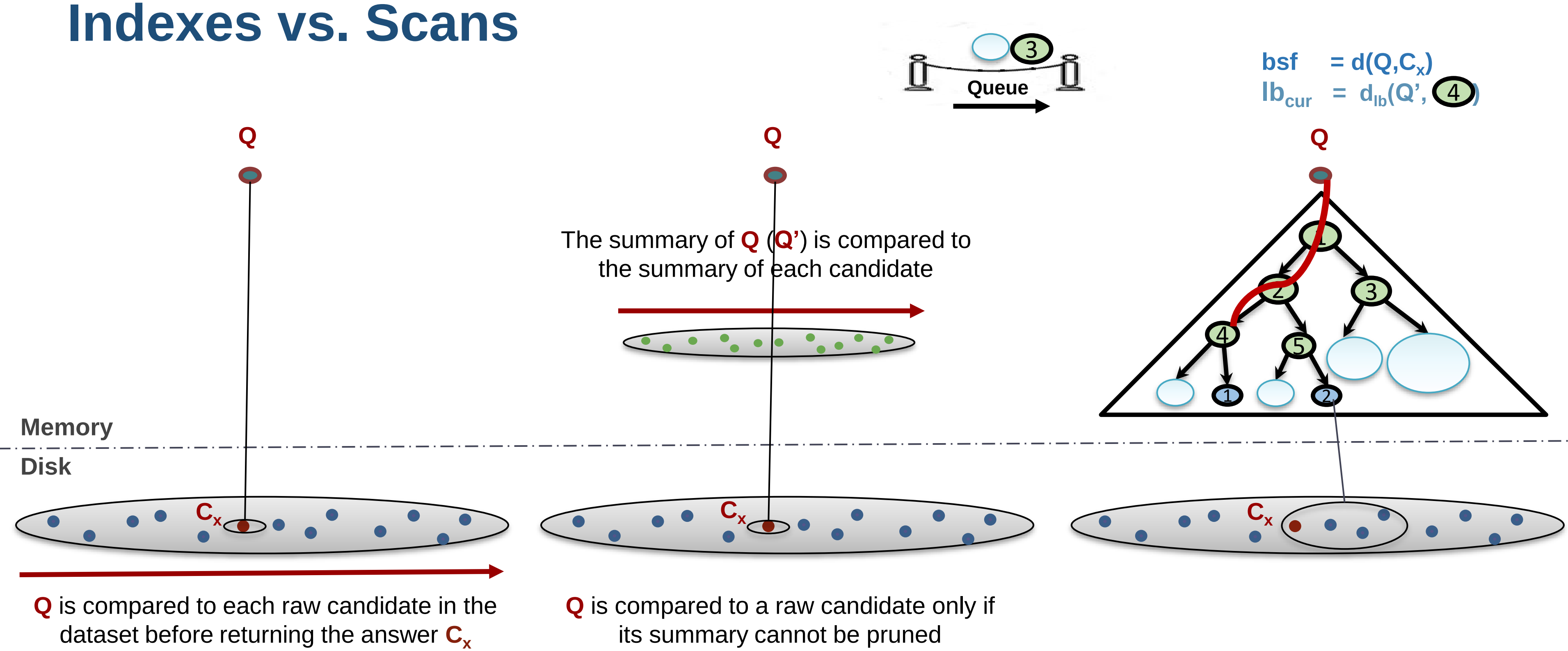
(a) Serial scan

(b) Skip-sequential scan

(c) Tree-based index

Answering a similarity search query using different access paths

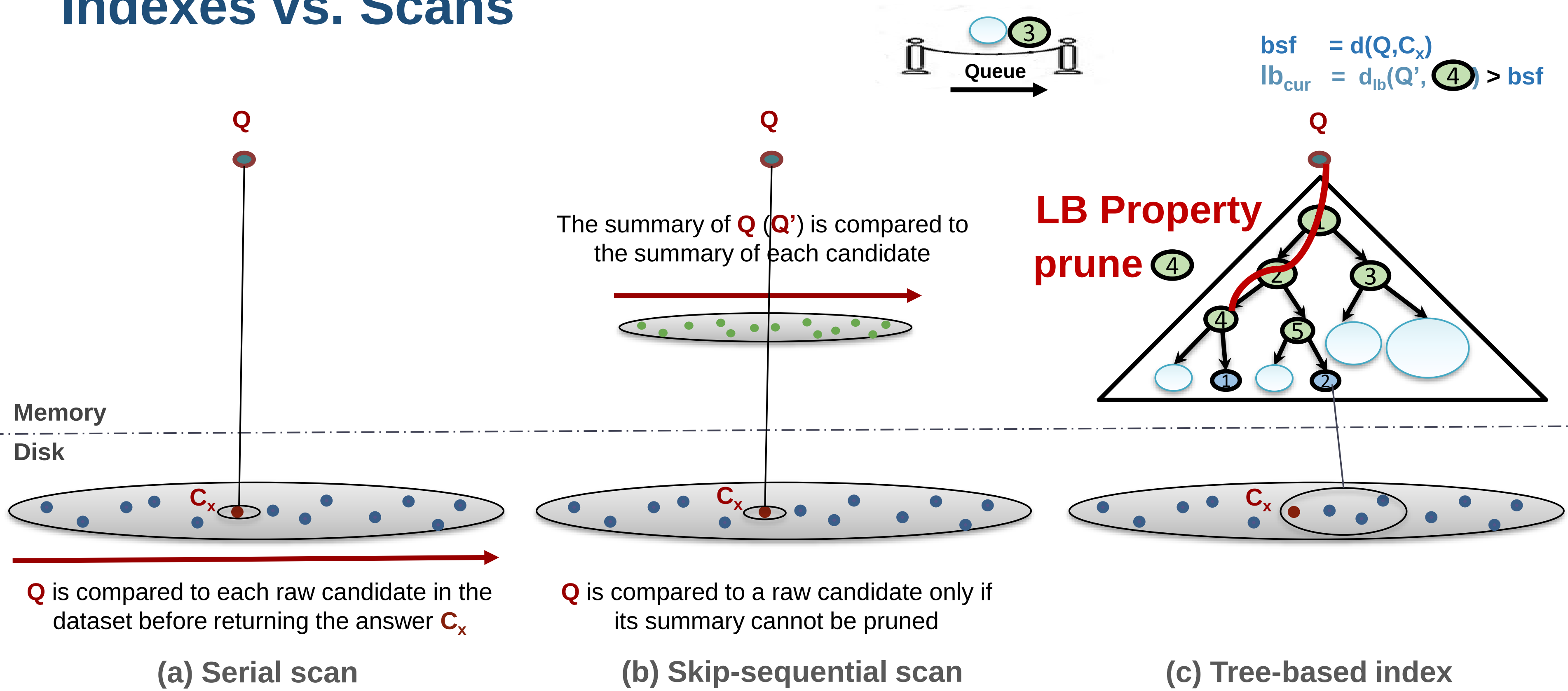
Indexes vs. Scans



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

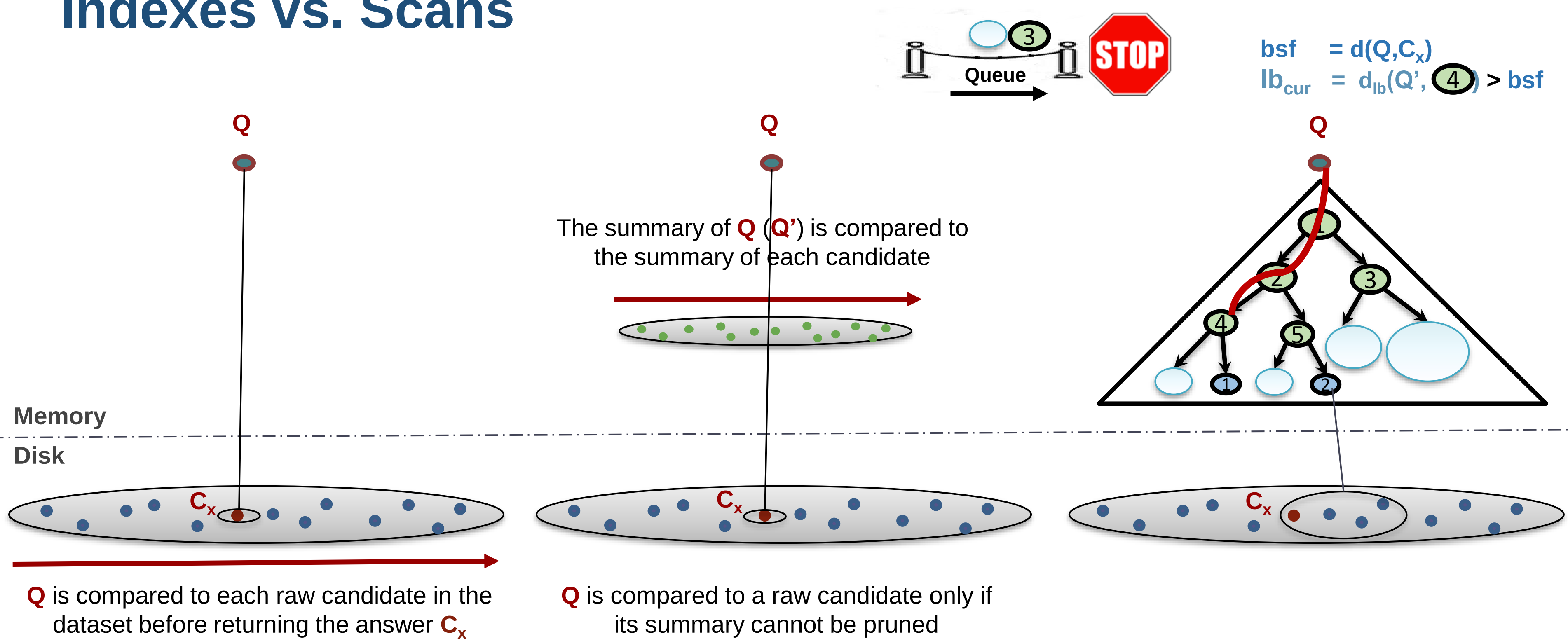
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

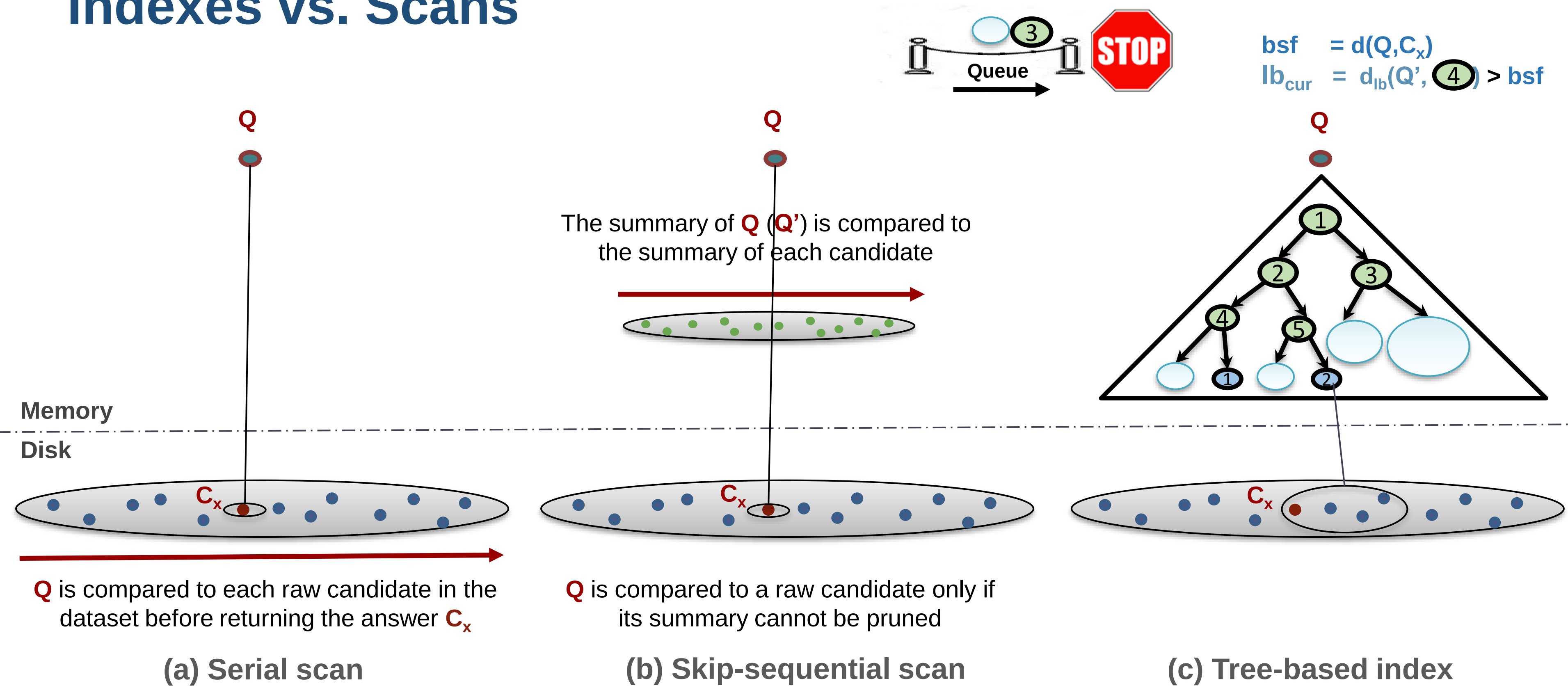
Indexes vs. Scans



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

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

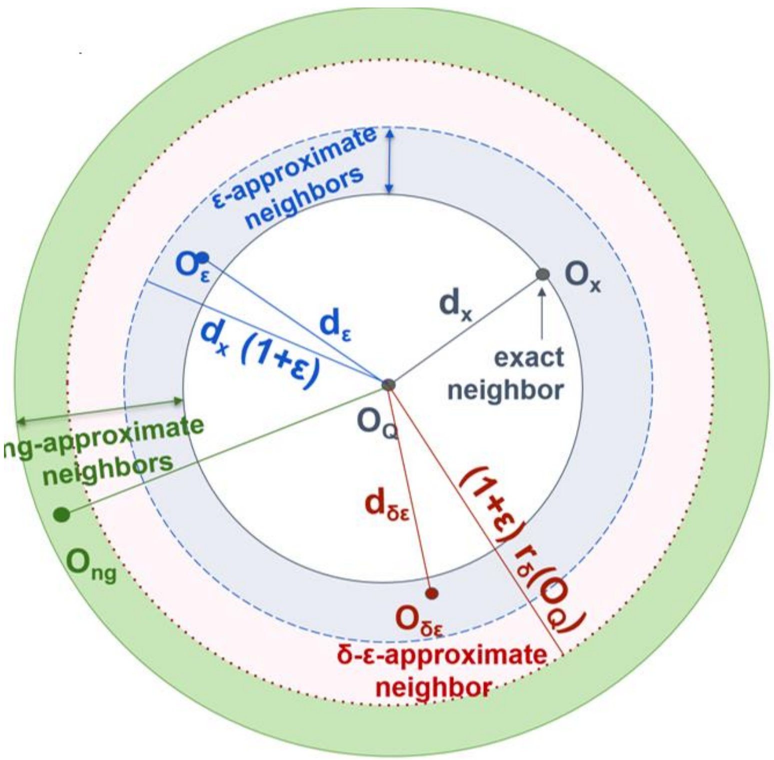
Indexes vs. Scans



Answering a similarity search query using different access paths

Similarity Search
Data Series Extensions

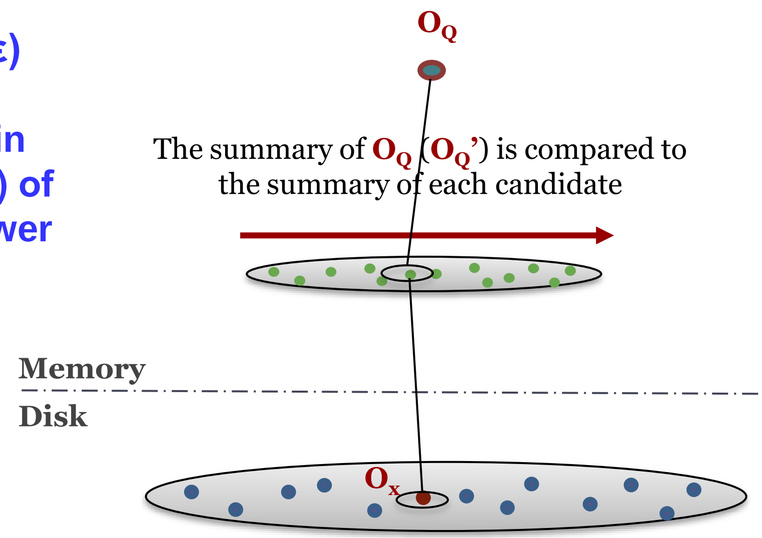
Extensions: Skip-Sequential Scans



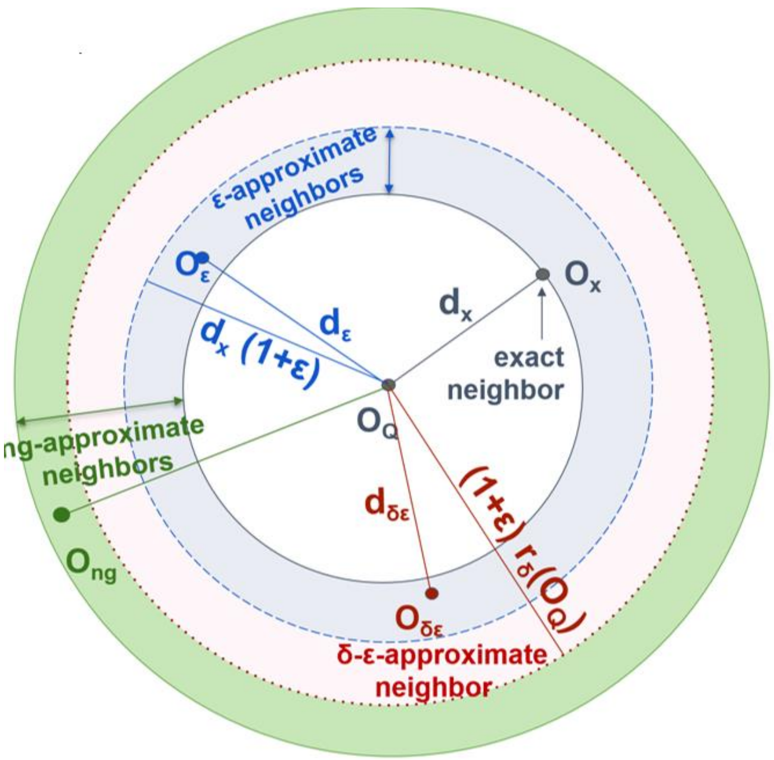
$d_ε$ <= d_x (1+ε)

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



Extensions: Skip-Sequential Scans



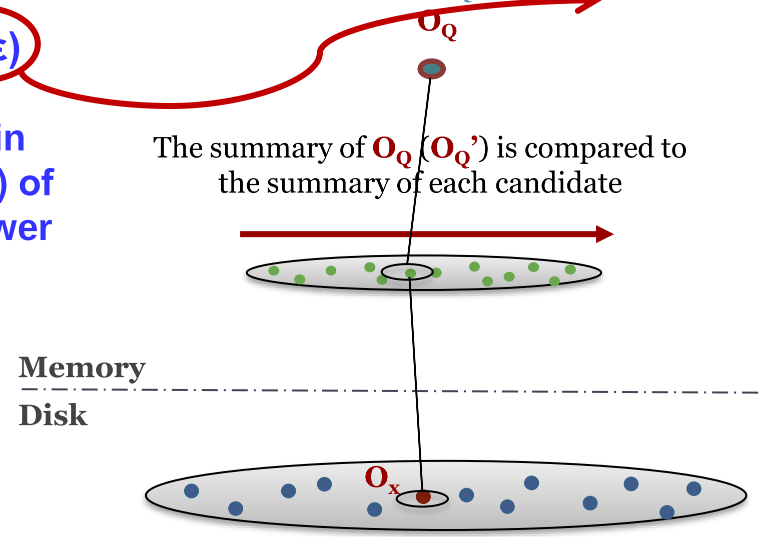
$d_ε$ <= d_x(1+ε)

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

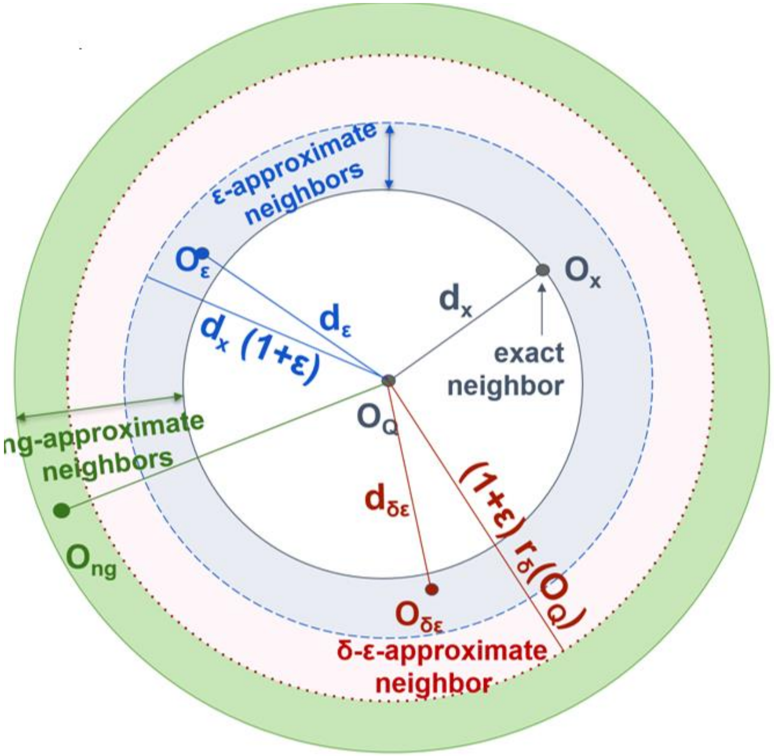
bsf = $d(O_Q,O_1)$

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

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



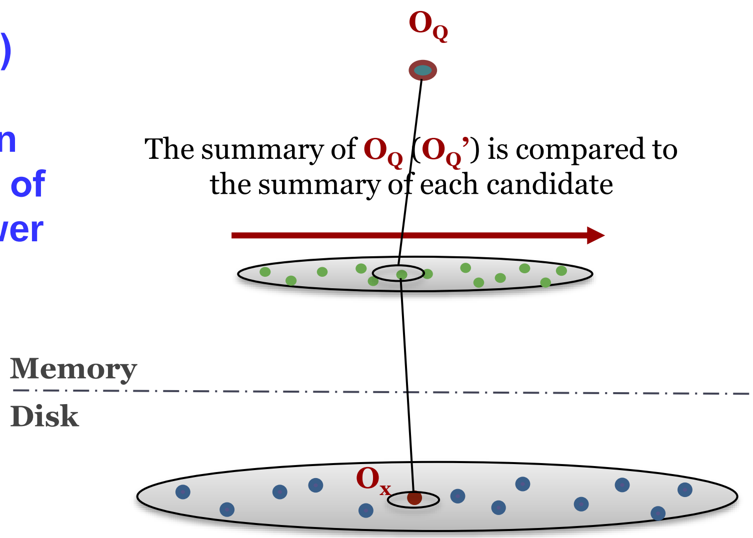
Extensions: Skip-Sequential Scans



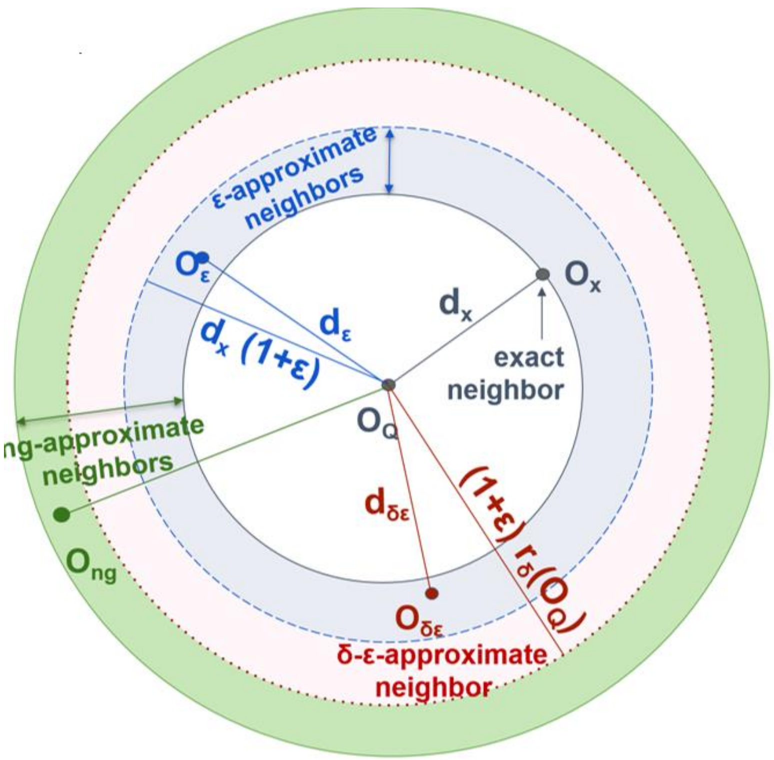
d_ε <= d_x (1+ε)

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

bsf = $d(O_Q, O_1)$

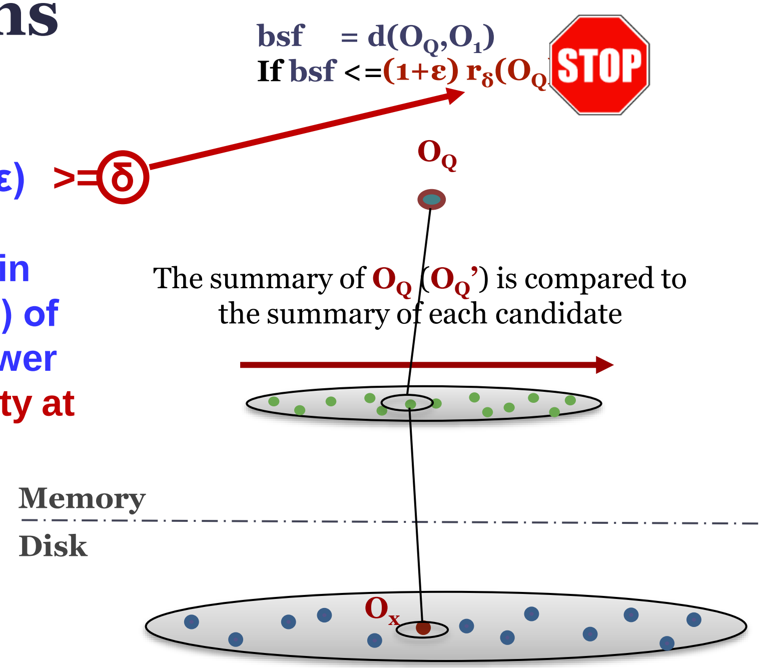


Extensions: Skip-Sequential Scans

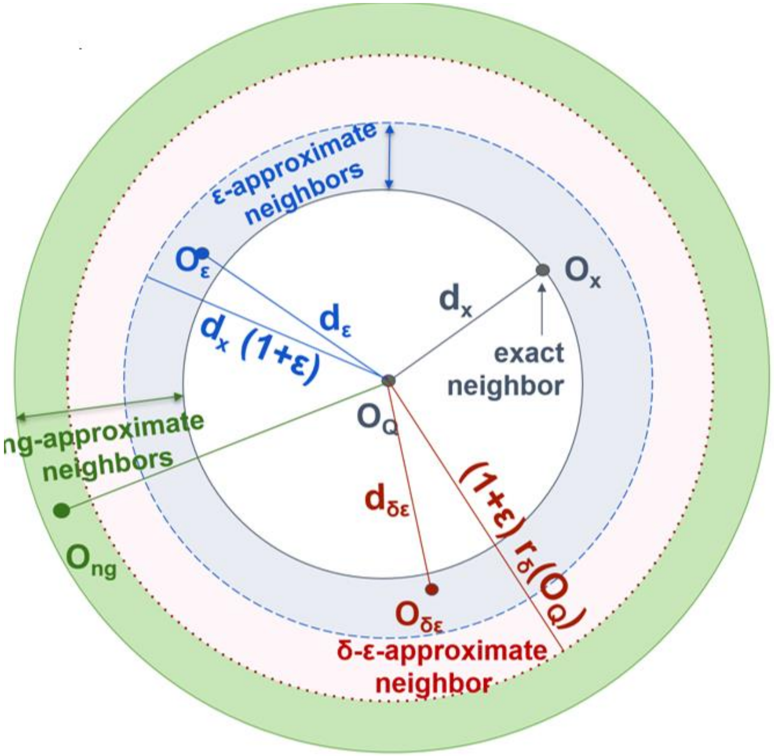


$P\{d_\varepsilon <= d_x\ (1+\varepsilon)\ \ >= \delta$

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



Extensions: Indexes

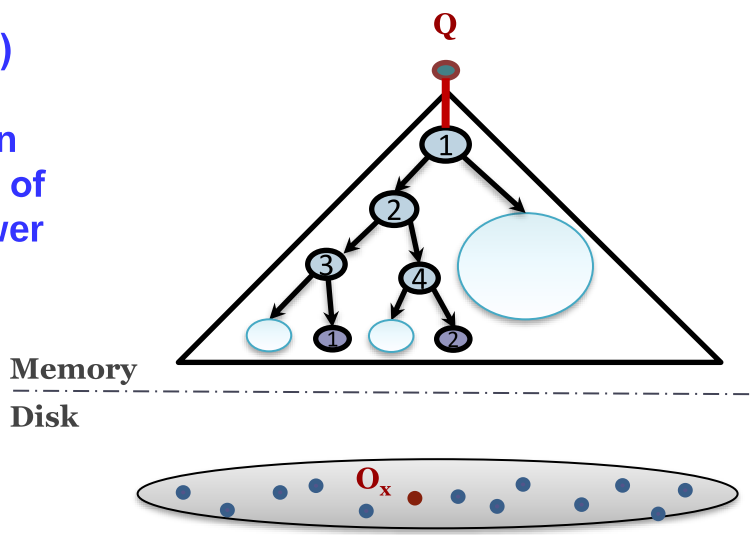


$d_\varepsilon <= d_x\ (1+\varepsilon)$

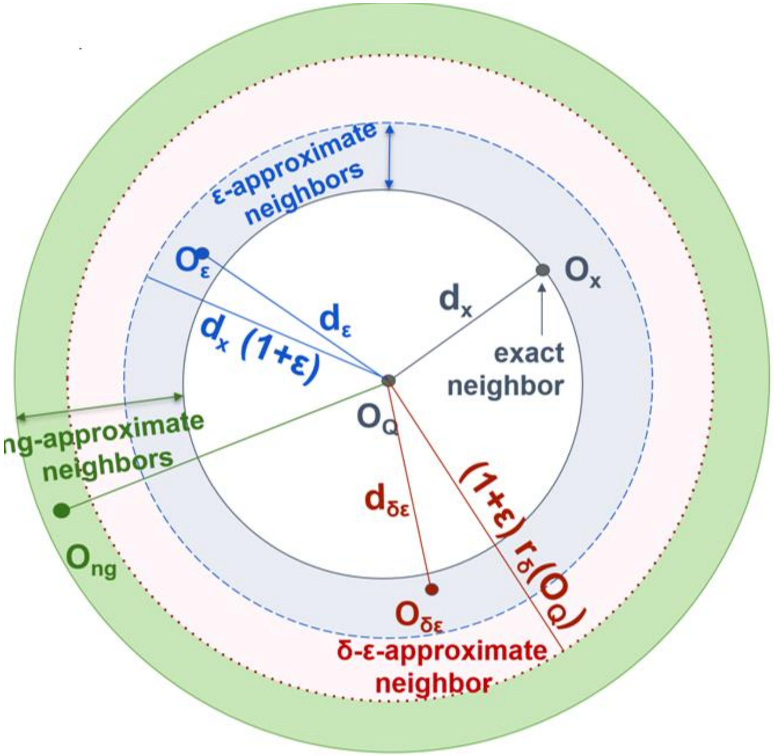
Result is within distance (1+ ε) of the exact answer

bsf $= d(O_Q,O_3)$

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

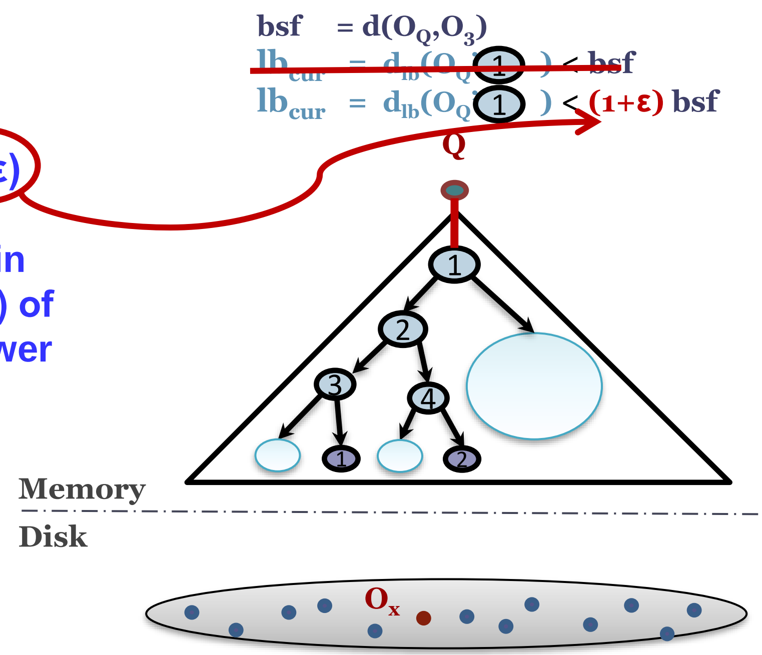


Extensions: Indexes

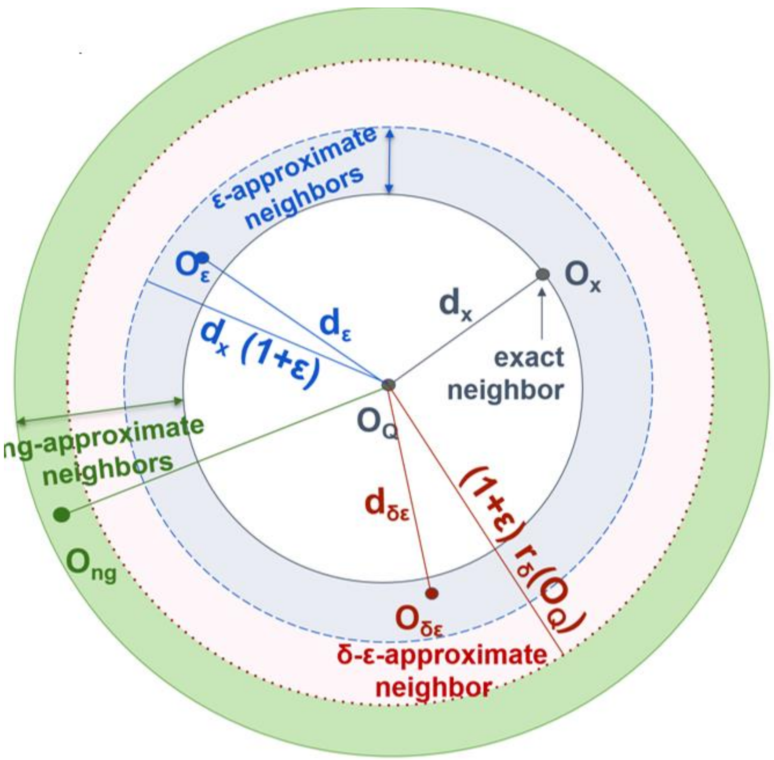


$d_\varepsilon <= d_x (1+\varepsilon)$

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



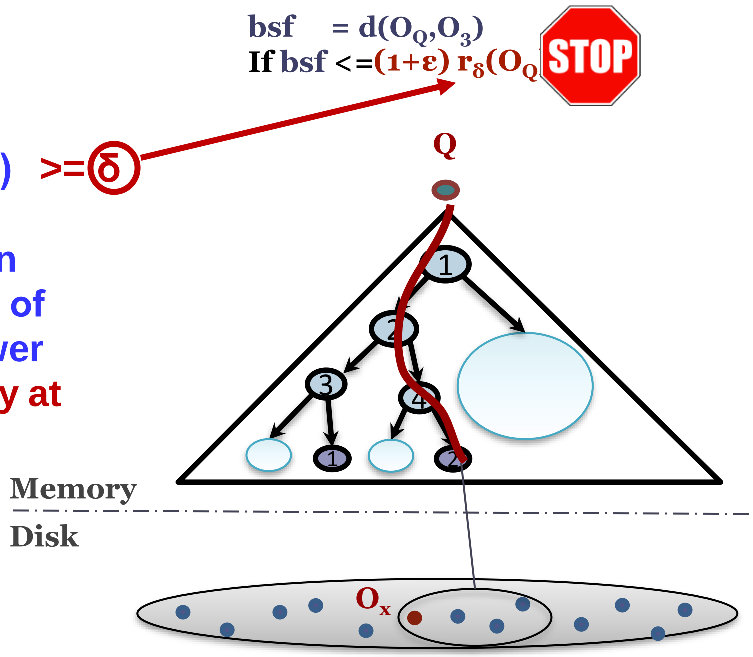
Extensions: Indexes



P{d_ε <= d_x (1+ε) >= δ

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

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



Publications

Echihabi-EDBT'21

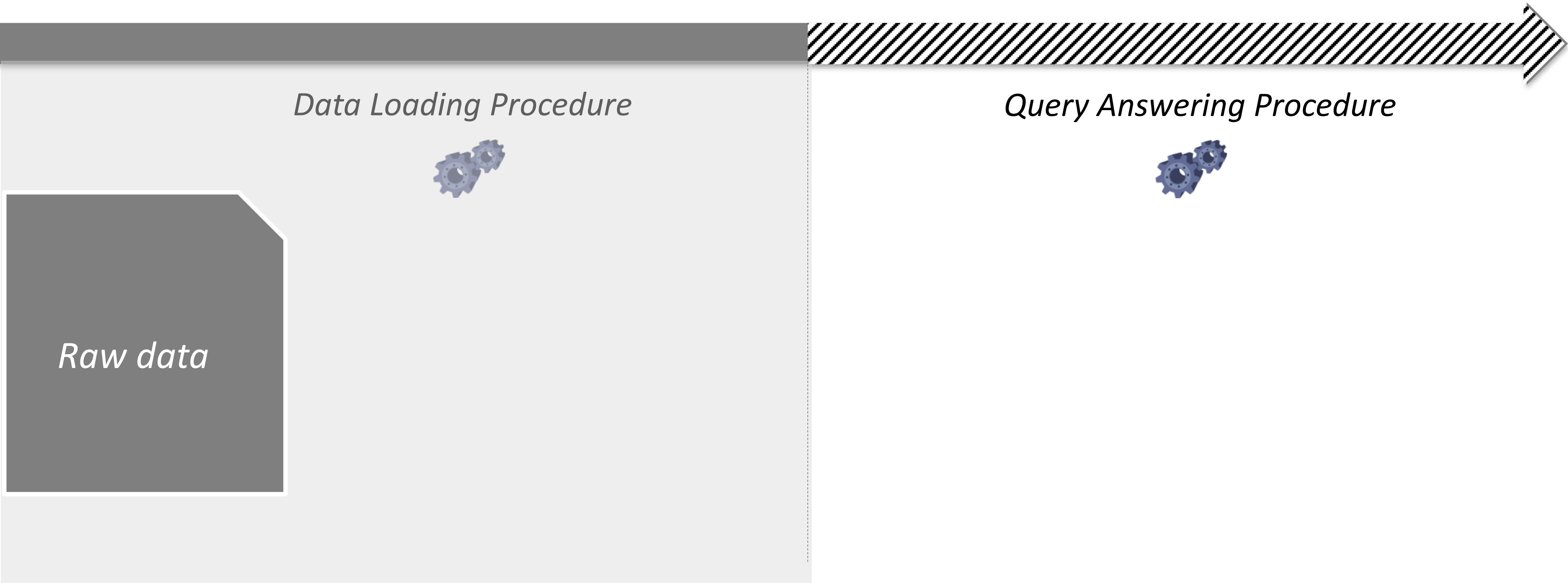
Data Series Similarity Search State-of-the-Art Methods

for a more complete and detailed presentation, see tutorial:

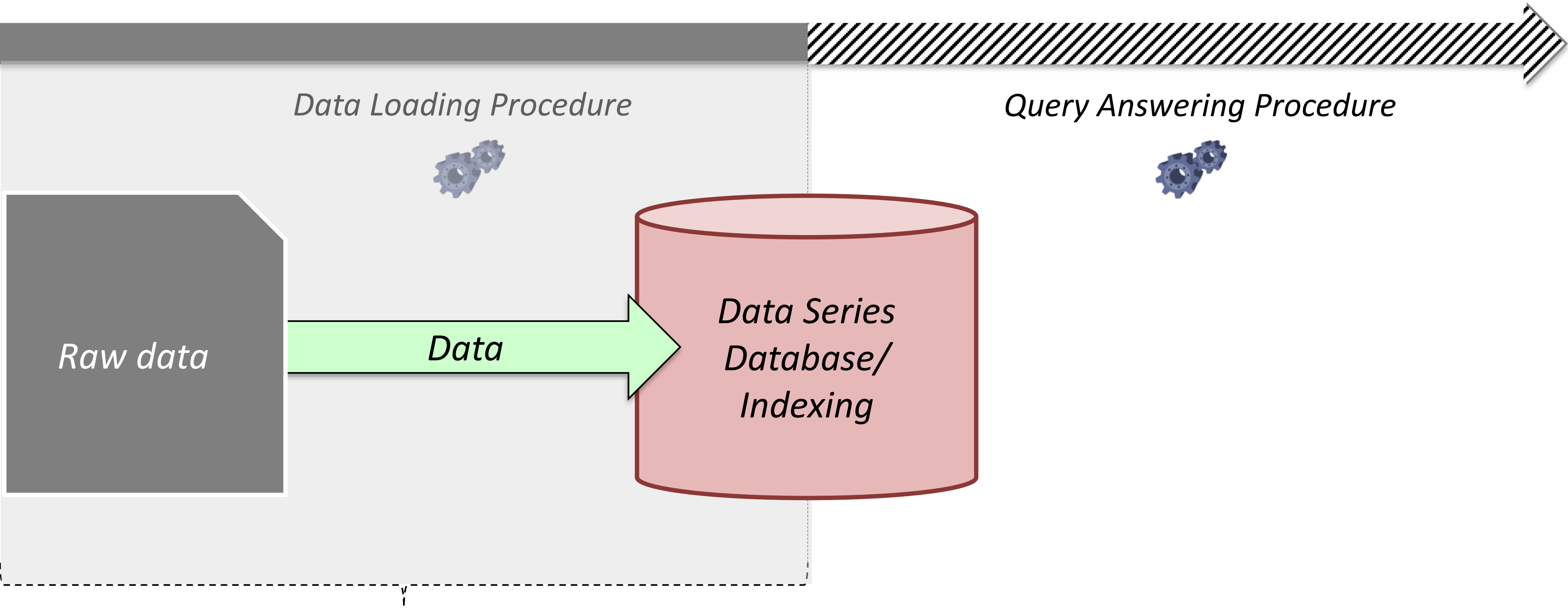
Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas. Big Sequence Management: Scaling Up and Out. EDBT 2021

http://helios.mi.parisdescartes.fr/~themisp/publications.html#tutorials

Query answering process

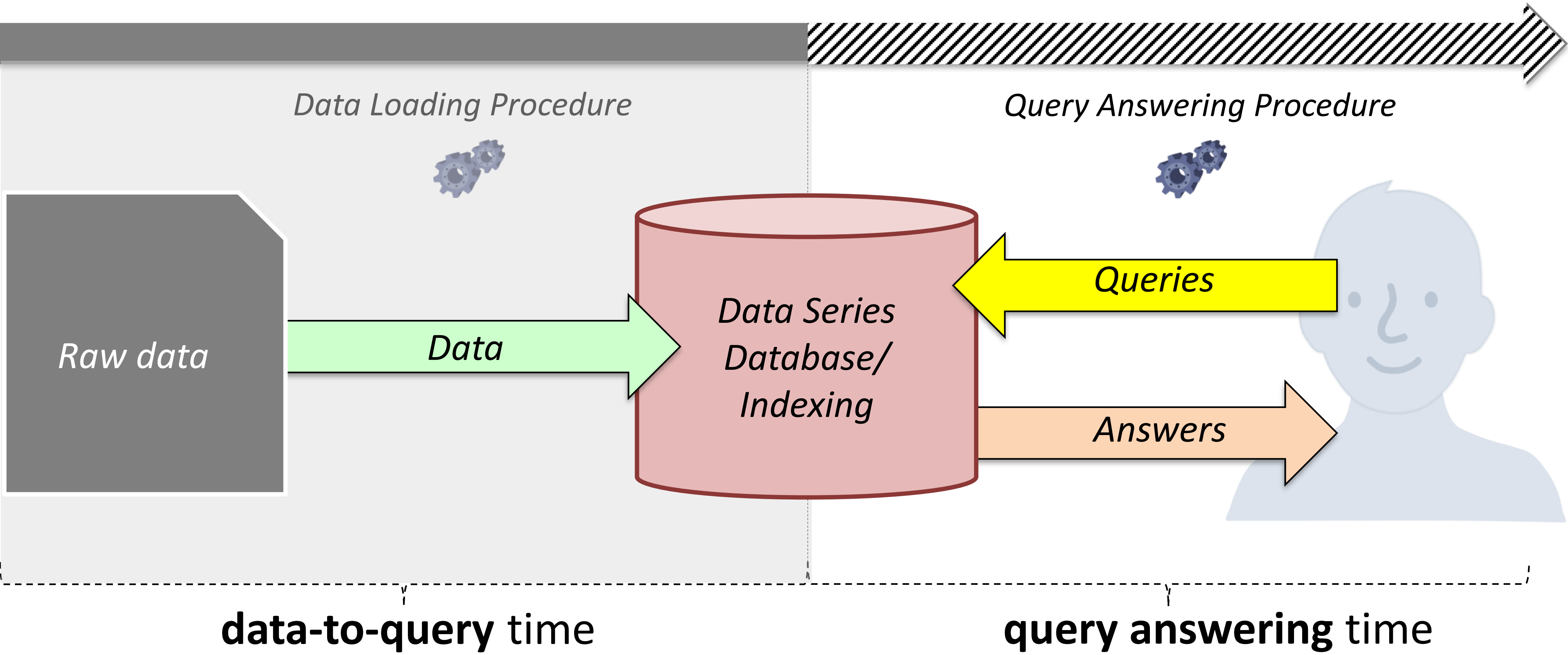


Query answering process

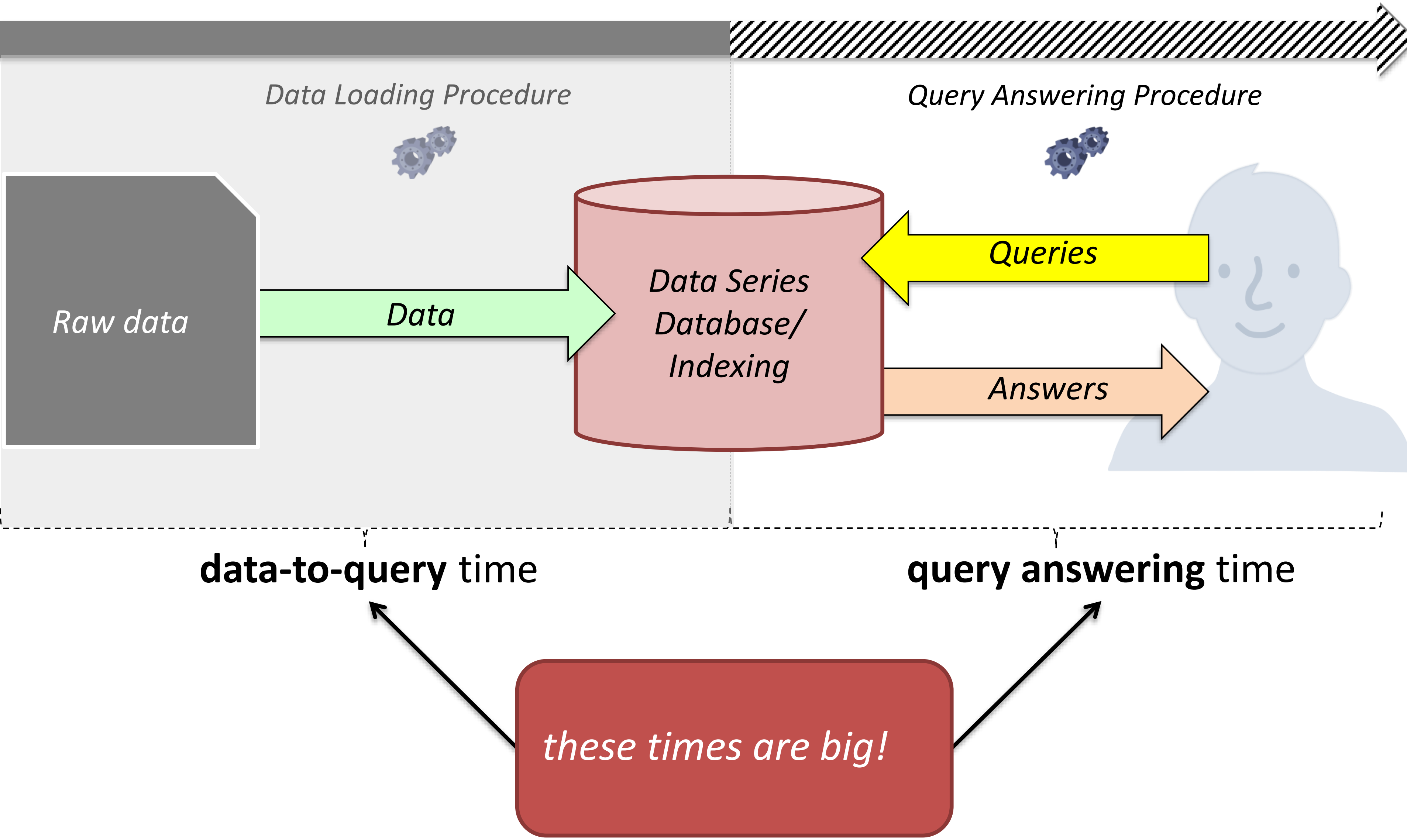


data-to-query time

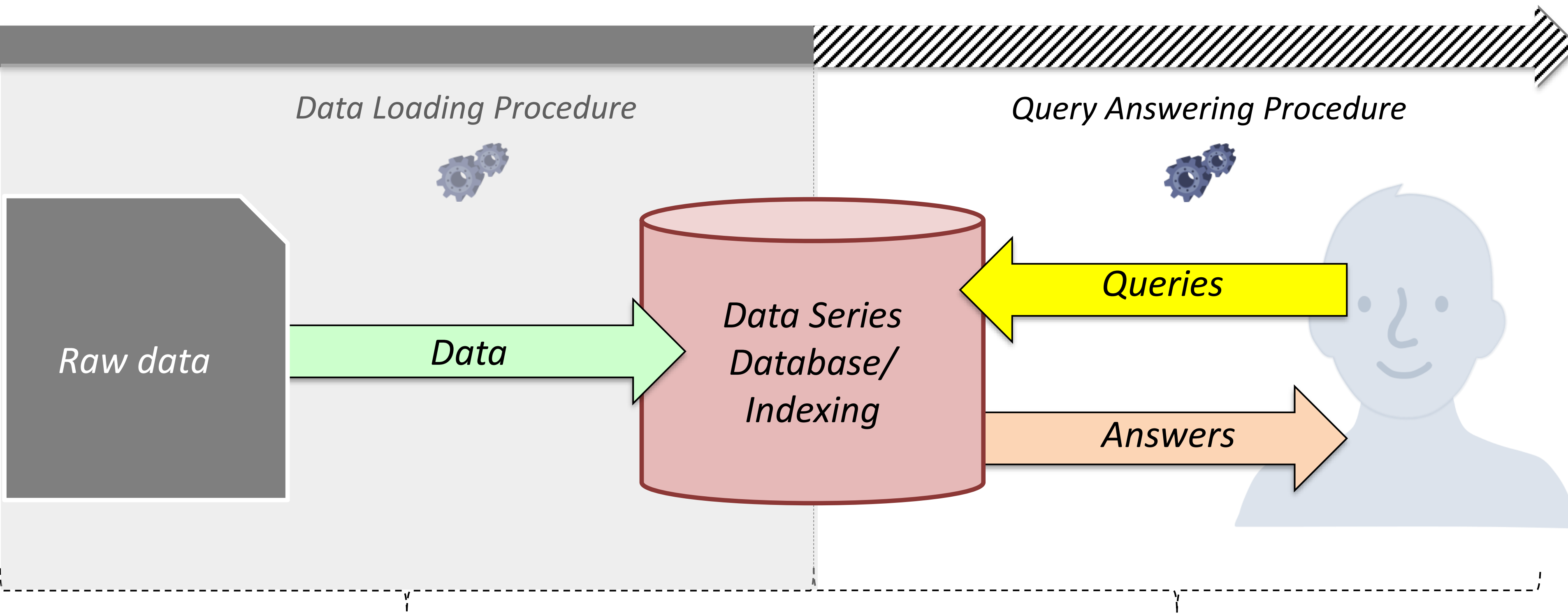
Query answering process



Query answering process



Query answering process



data-to-query time

query answering time

we have proposed the state-of-the-art solutions for both problems!

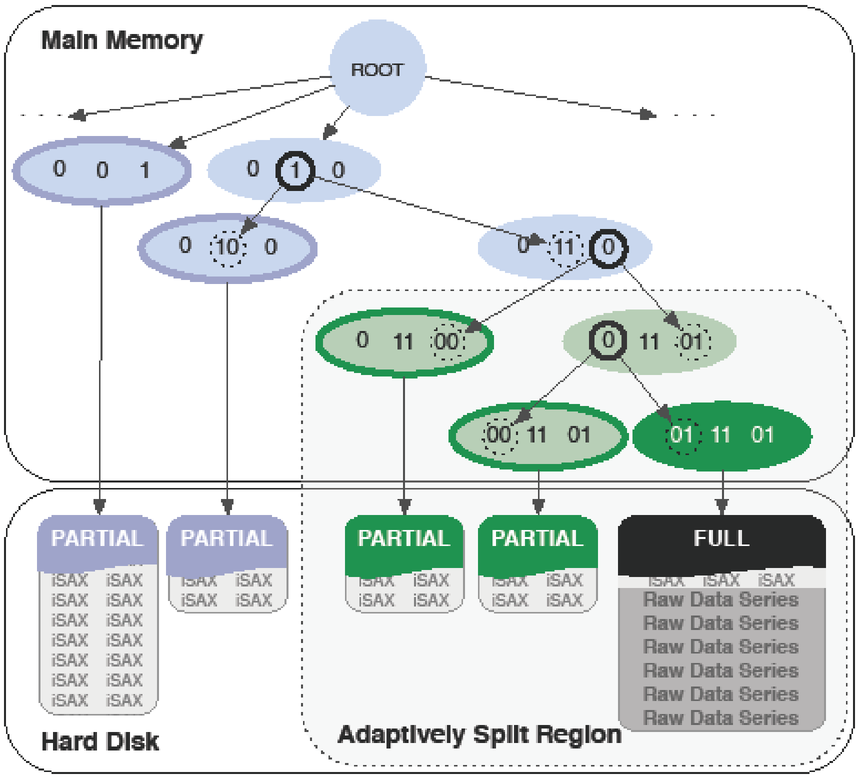
The ADS+ Solution

Publications

SIGMOD'14

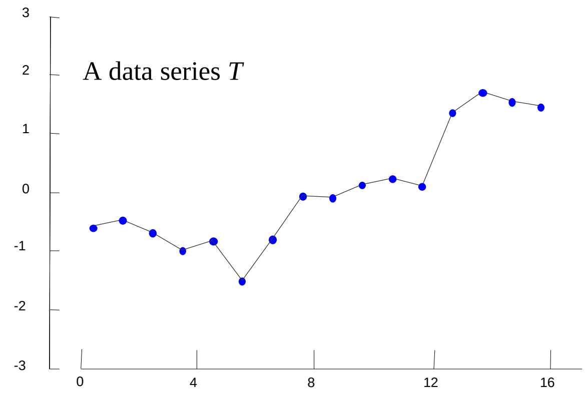
PVLDB'15

VLDBJ'16



SAX Representation

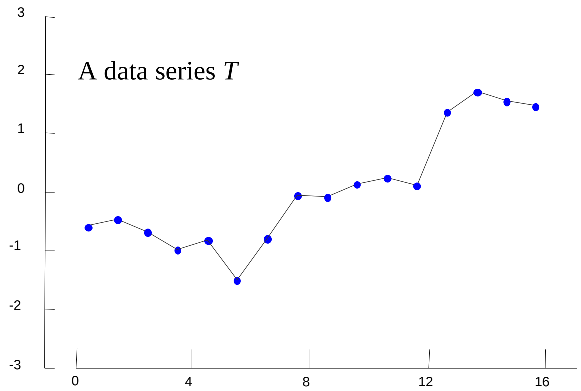
- **S**ymbolic **A**ggregate appro**X**imation (SAX)
  - **(1)** Represent data series *T* of length ***n*** with ***w*** segments using Piecewise Aggregate Approximation (PAA)

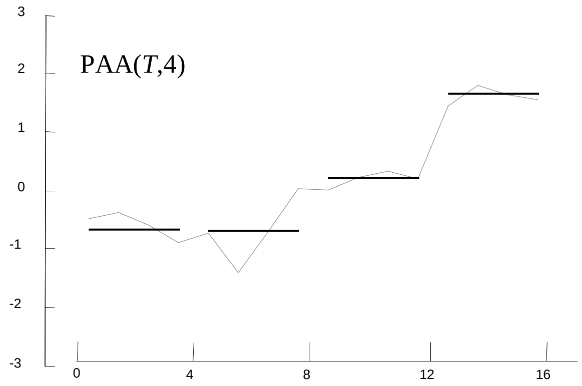


SAX Representation

- **S**ymbolic **A**ggregate appro**X**imation (SAX)
  - **(1)** Represent data series *T* of length ***n*** with ***w*** segments using Piecewise Aggregate Approximation (PAA)
    - *T* typically normalized to $\mu = 0$, $\sigma = 1$
    - PAA(*T*,*w*) $= \bar{T} = \bar{t}_1, \ldots, \bar{t}_w$

      where $\bar{t}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} T_j$

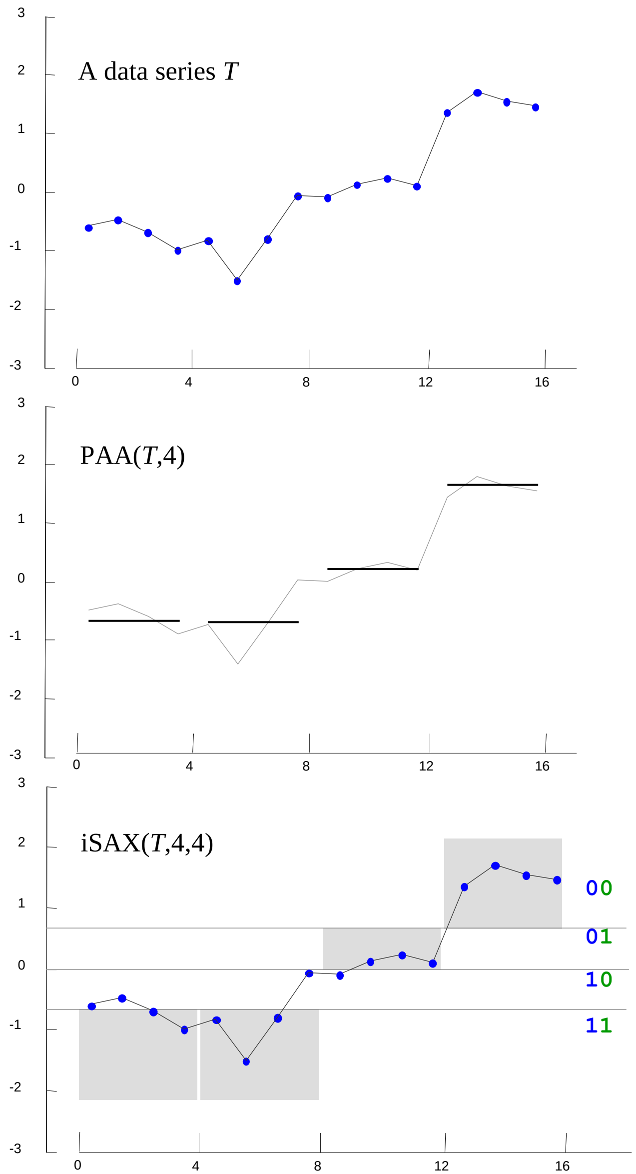




SAX Representation

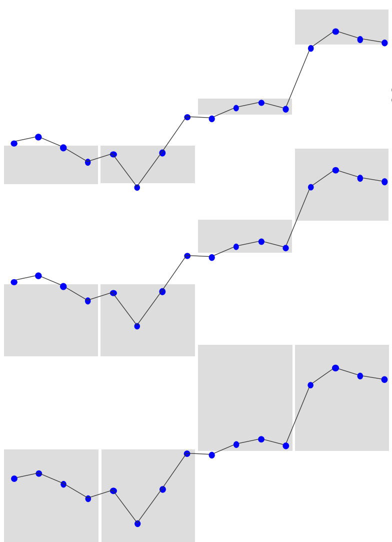
- **S**ymbolic **A**ggregate appro**X**imation (SAX)
  - **(1)** Represent data series T of length $\boldsymbol{n}$ with $\boldsymbol{w}$ segments using Piecewise Aggregate Approximation (PAA)
    - T typically normalized to $\mu = 0$, $\sigma = 1$
    - $\mathrm{PAA}(T,w) = \bar{T} = \bar{t}_1, \ldots, \bar{t}_w$

      where $\bar{t}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} T_j$
  - **(2)** Discretize into a vector of symbols
    - Breakpoints map to small alphabet $\boldsymbol{a}$ of symbols



*i*SAX Representation

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



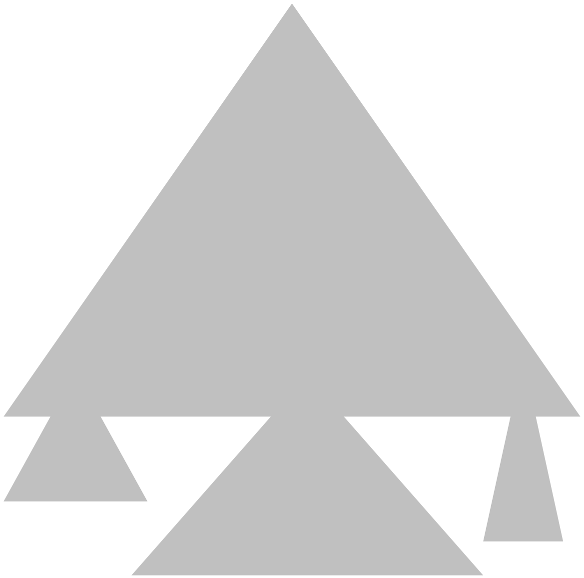
= { 6, 6, 3, 0} = {110 ,110 ,0111 ,000}

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

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

*i*SAX Index

- non-balanced tree-based index with non-overlapping regions, and controlled fan-out rate
    - base cardinality ***b*** (optional), segments ***w***, threshold ***th***
    - hierarchically subdivides SAX space until num. entries ≤ ***th***

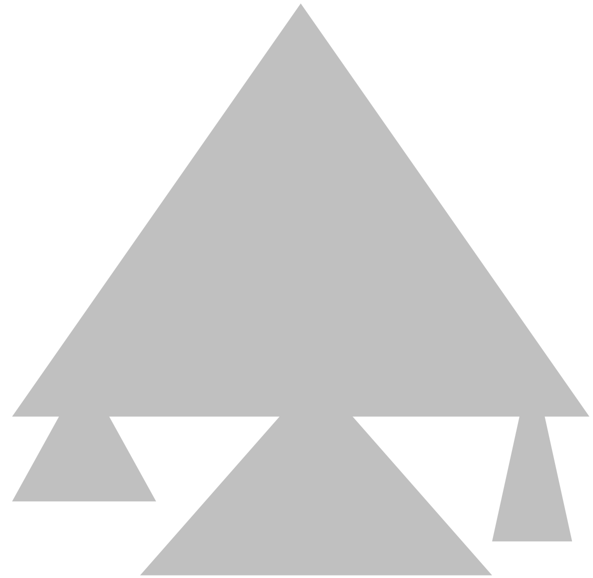


*i*SAX Index

- non-balanced tree-based index with non-overlapping regions, and controlled fan-out rate
  - base cardinality ***b*** (optional), segments ***w***, threshold ***th***
  - hierarchically subdivides SAX space until num. entries ≤ ***th***

e.g., th=4, w=4, b=1

$$\left[\begin{array}{cccc} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{array}\right.$$



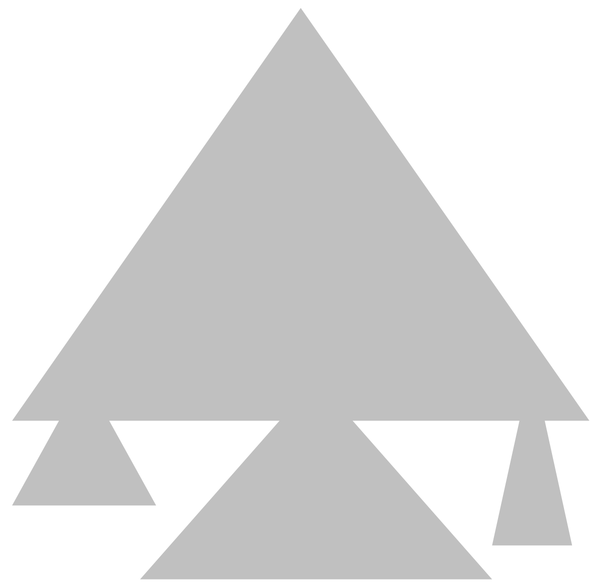
*i*SAX Index

- non-balanced tree-based index with non-overlapping regions, and controlled fan-out rate
  - base cardinality ***b*** (optional), segments ***w***, threshold ***th***
  - hierarchically subdivides SAX space until num. entries ≤ ***th***

e.g., th=4, w=4, b=1

Insert:
1 1 1 0

$$\rightarrow \left[\begin{array}{cccc} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{array}\right.$$



*i*SAX Index

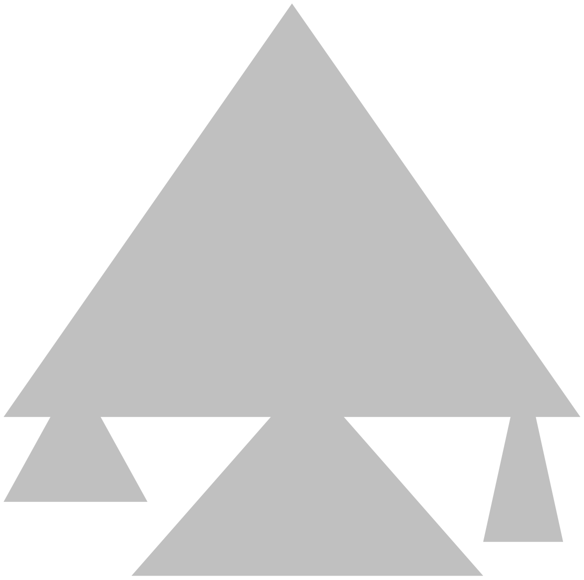
- non-balanced tree-based index with non-overlapping regions, and controlled fan-out rate
  - base cardinality ***b*** (optional), segments ***w***, threshold ***th***
  - hierarchically subdivides SAX space until num. entries ≤ ***th***

e.g., th=4, w=4, b=1

1 1 **1** 0

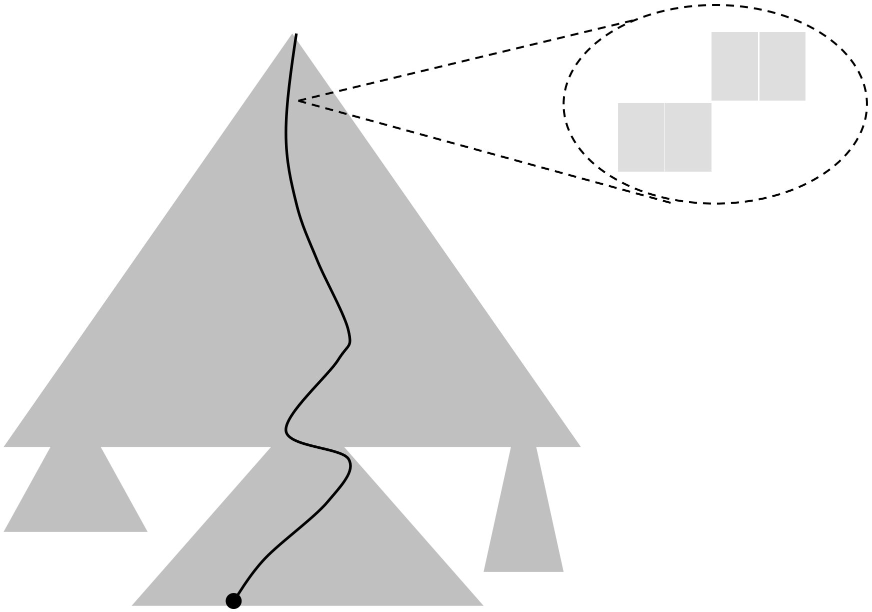
1 1 1**0** 0
1 1 1**0** 0

1 1 1**1** 0
1 1 1**1** 0
1 1 1**1** 0



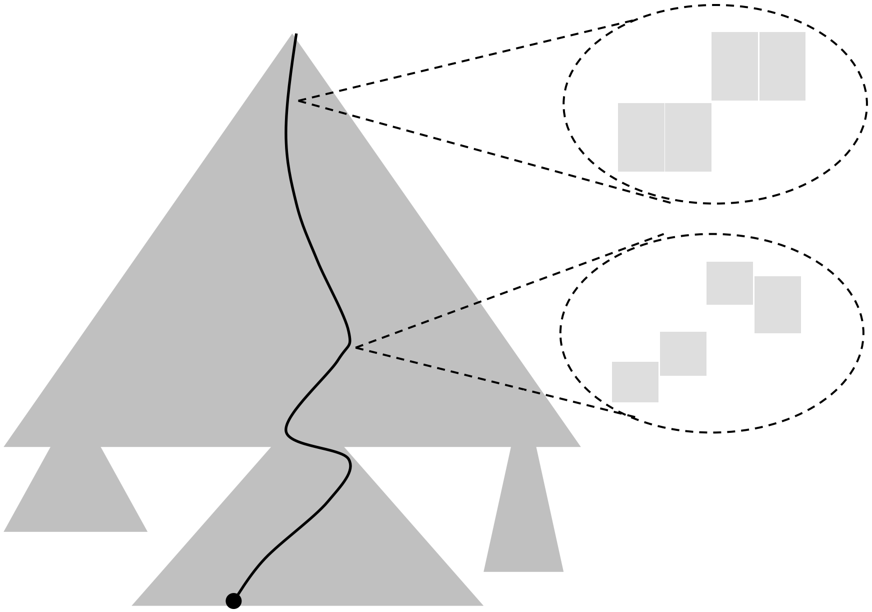
*i*SAX Index

- non-balanced tree-based index with non-overlapping regions, and controlled fan-out rate
  - base cardinality ***b*** (optional), segments ***w***, threshold ***th***
  - hierarchically subdivides SAX space until num. entries ≤ ***th***



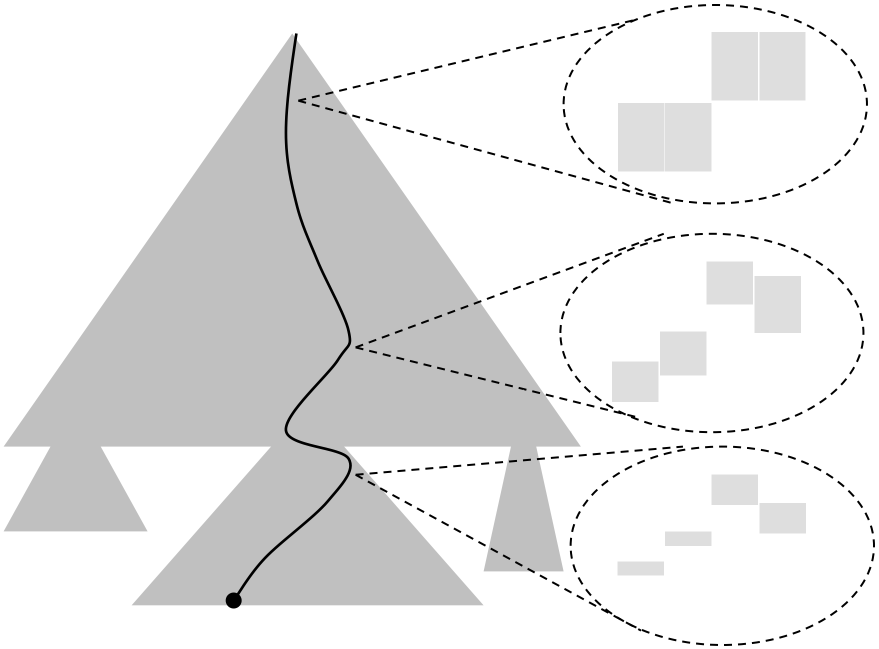
*i*SAX Index

- non-balanced tree-based index with non-overlapping regions, and controlled fan-out rate
  - base cardinality ***b*** (optional), segments ***w***, threshold ***th***
  - hierarchically subdivides SAX space until num. entries ≤ ***th***



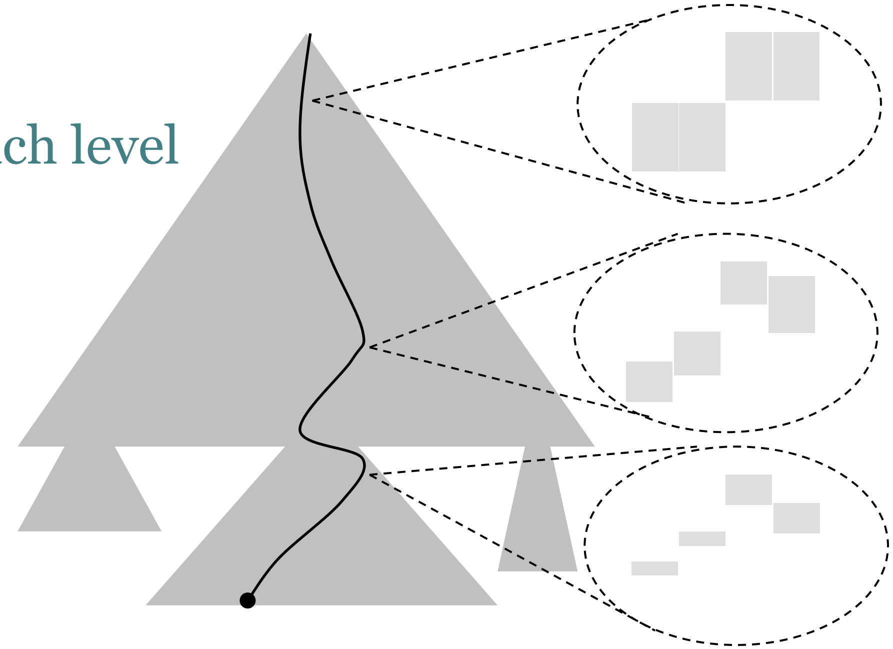
*i*SAX Index

- non-balanced tree-based index with non-overlapping regions, and controlled fan-out rate
  - base cardinality ***b*** (optional), segments ***w***, threshold ***th***
  - hierarchically subdivides SAX space until num. entries ≤ ***th***



*i*SAX Index

- non-balanced tree-based index with non-overlapping regions, and controlled fan-out rate
  - base cardinality ***b*** (optional), segments ***w***, threshold ***th***
  - hierarchically subdivides SAX space until num. entries ≤ ***th***
- Approximate Search
  - Match *i*SAX representation at each level
- Exact Search
  - Leverage approximate search
  - Prune search space
    - Lower bounding distance



Adaptive Data Series Index: ADS+

- novel paradigm for building a data series index
    - do not build entire index and then answer queries
    - start answering queries by building the part of the index needed by those queries
- still guarantee correct answers

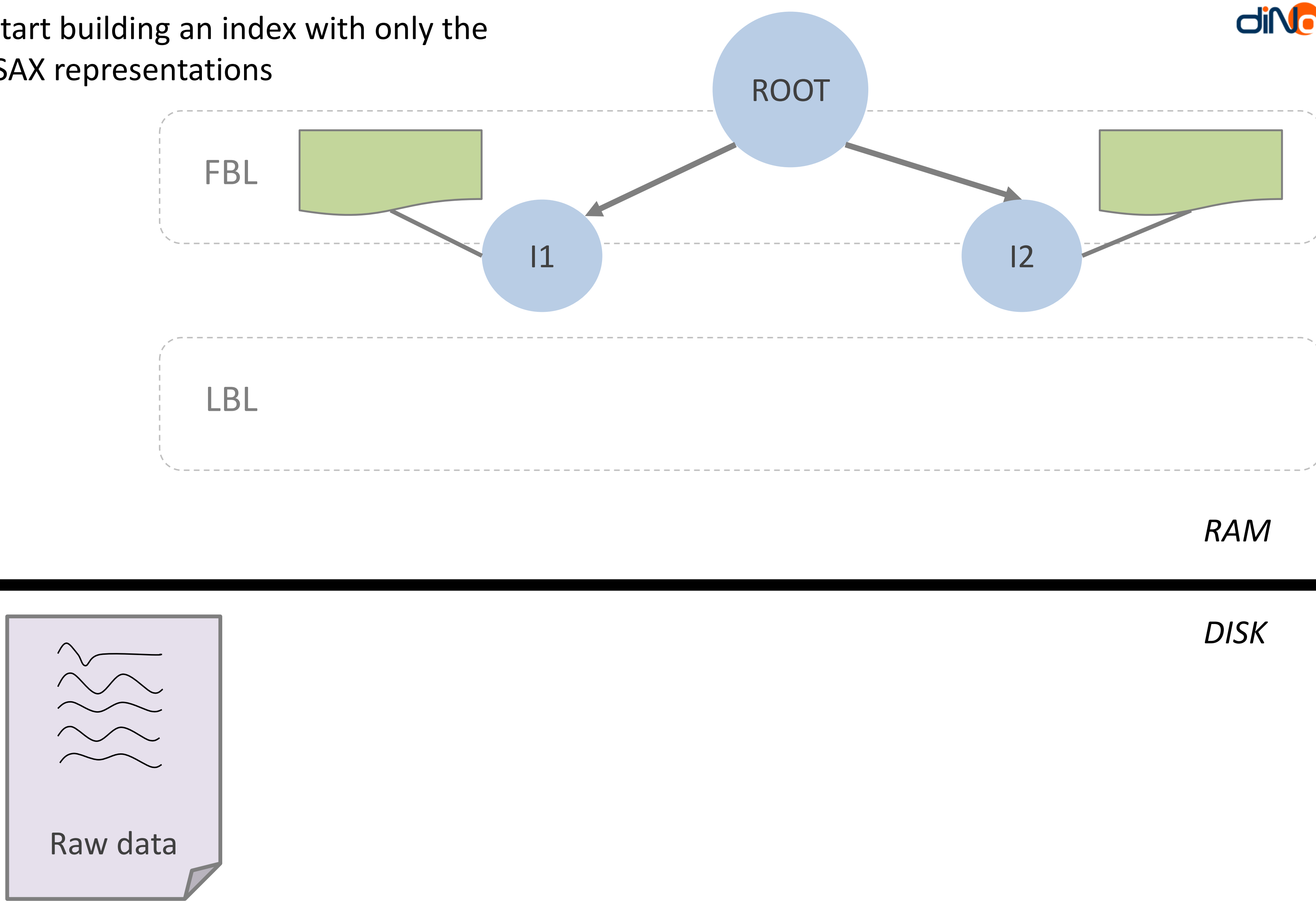
Publications

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

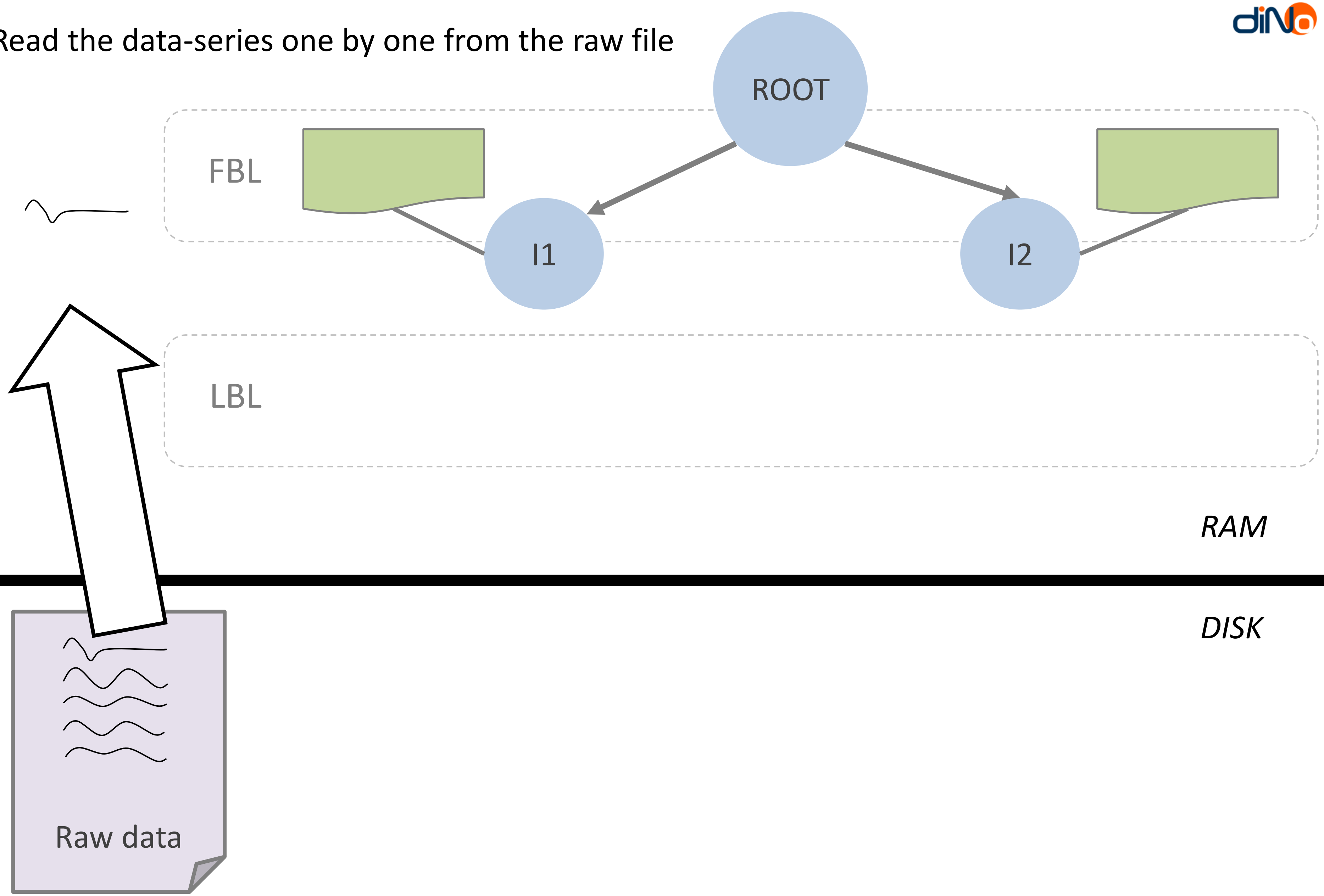
Adaptive Data Series Index: ADS+

- intuition for proposed solution
  - build the iSAX index using the iSAX representations
    - just like iSAX2+
  - but start with a large leaf size
    - minimize initial cost
  - postpone 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)

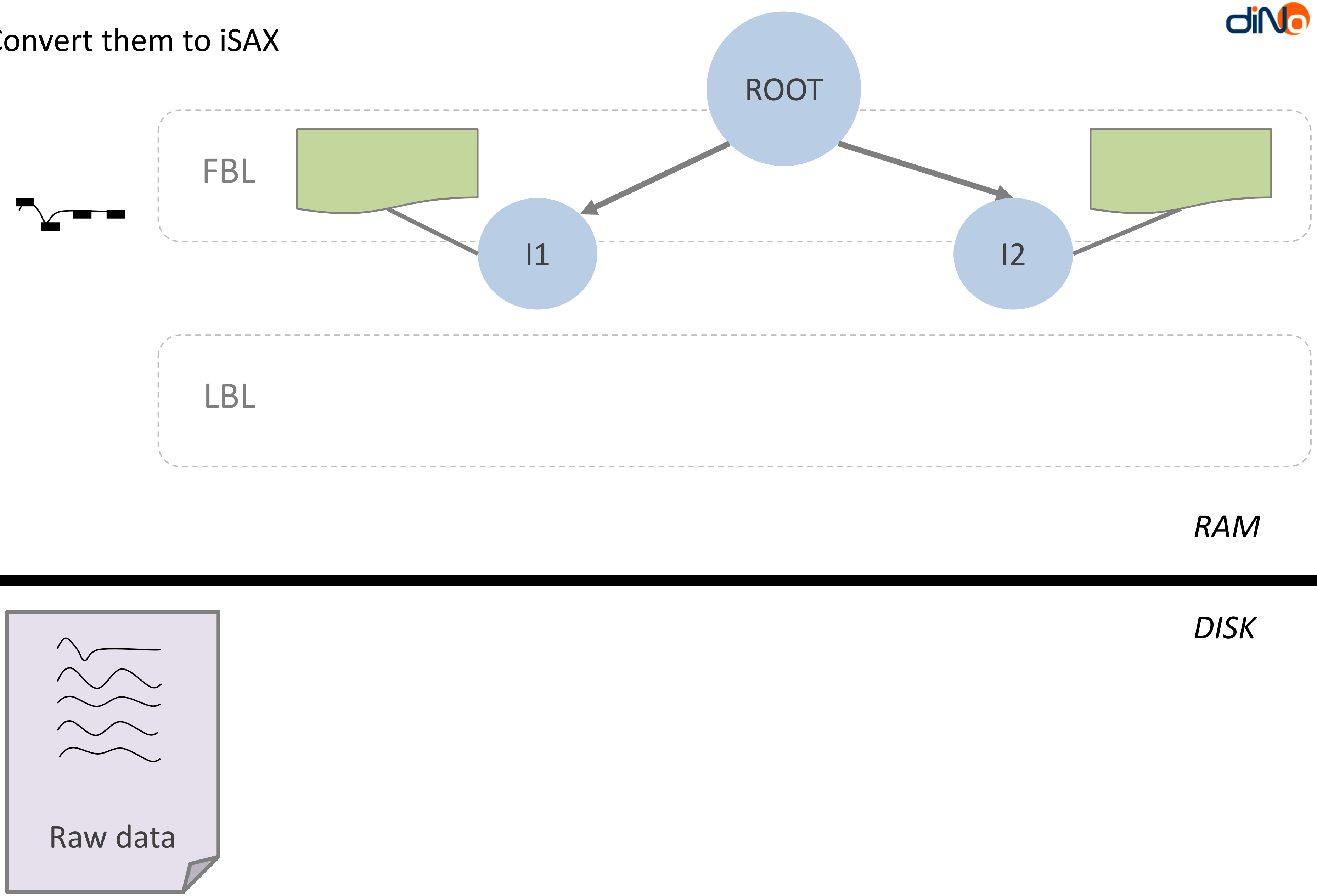
Start building an index with only the iSAX representations



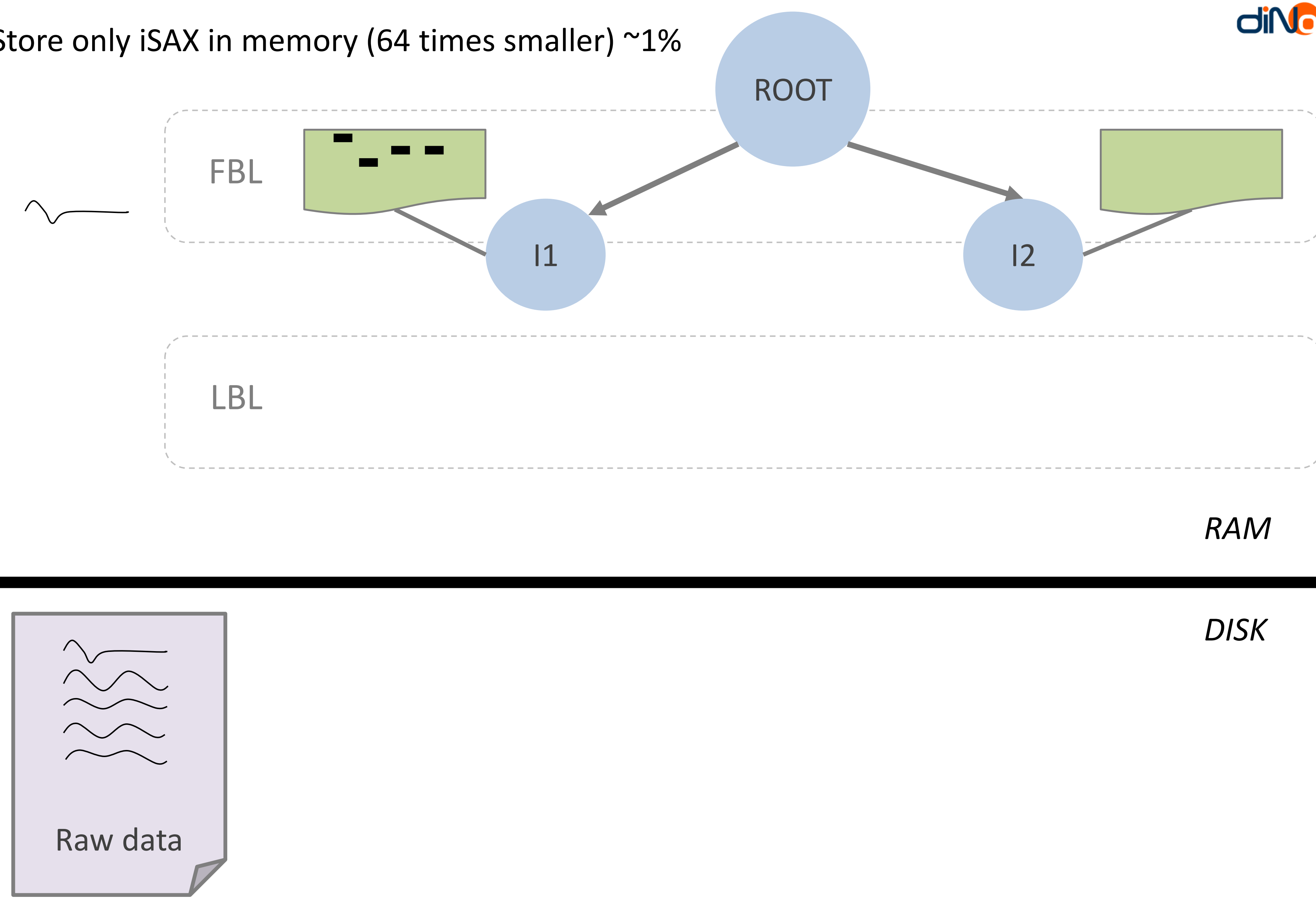
Read the data-series one by one from the raw file



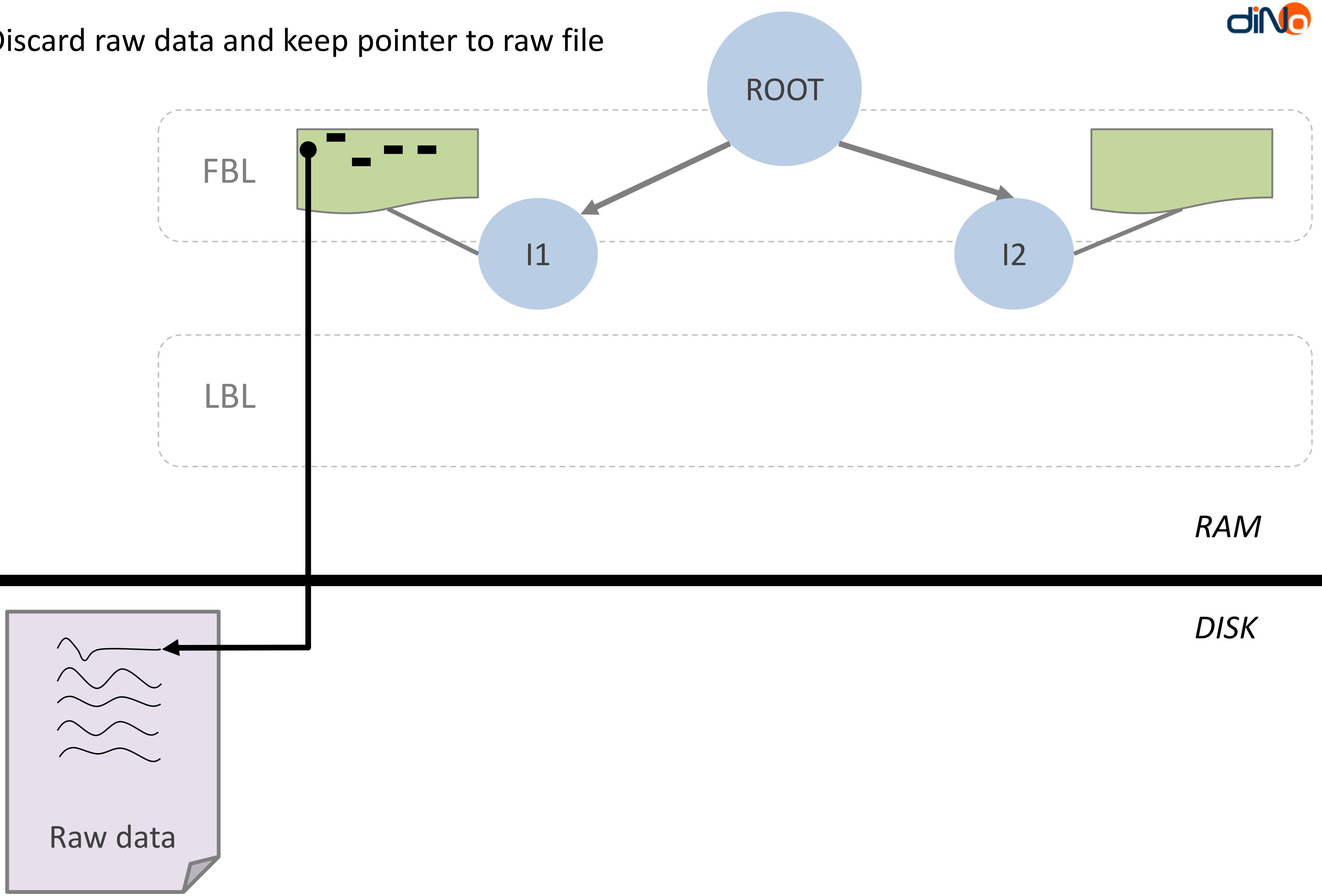
Convert them to iSAX



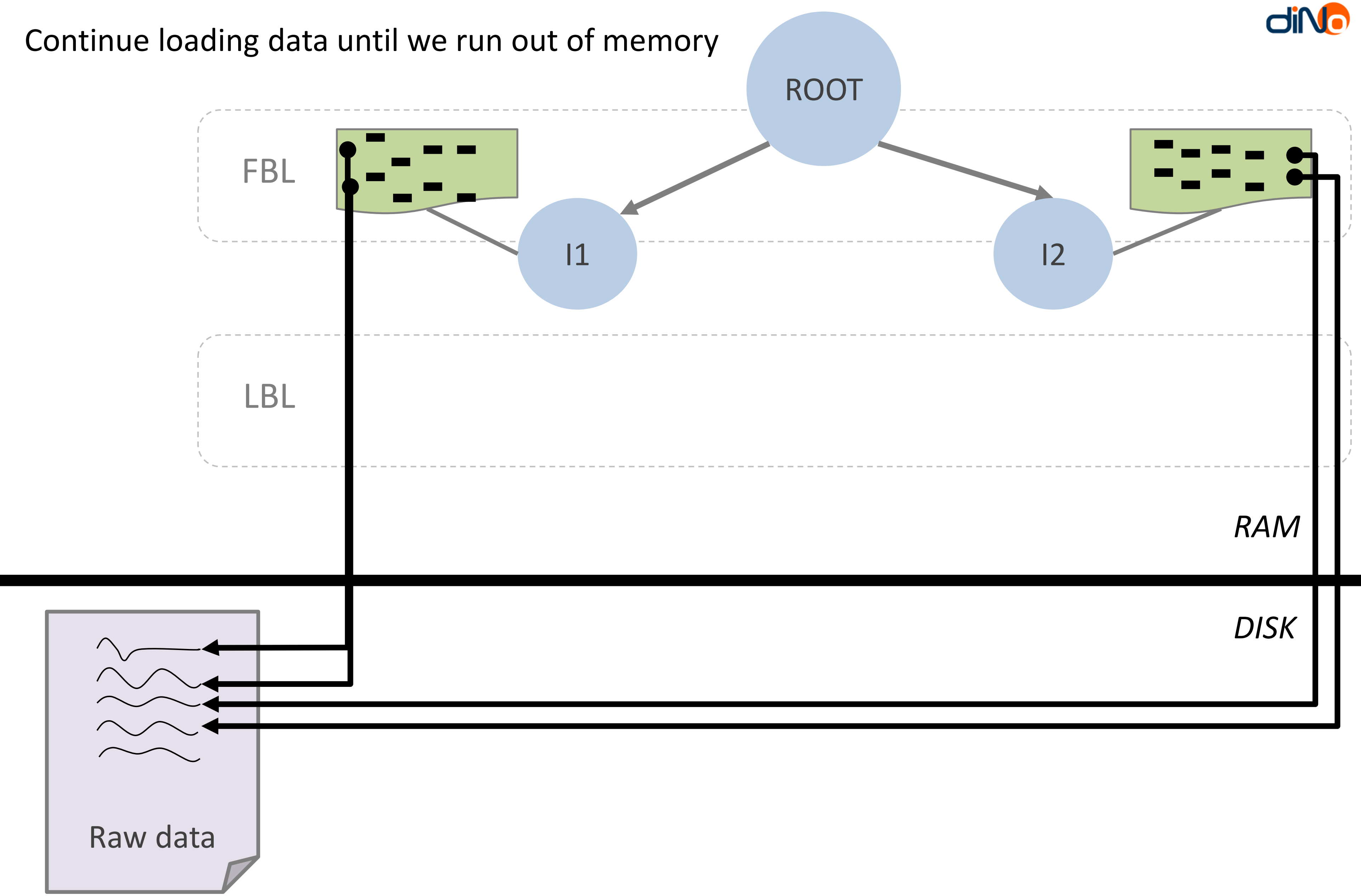
Store only iSAX in memory (64 times smaller) ~1%



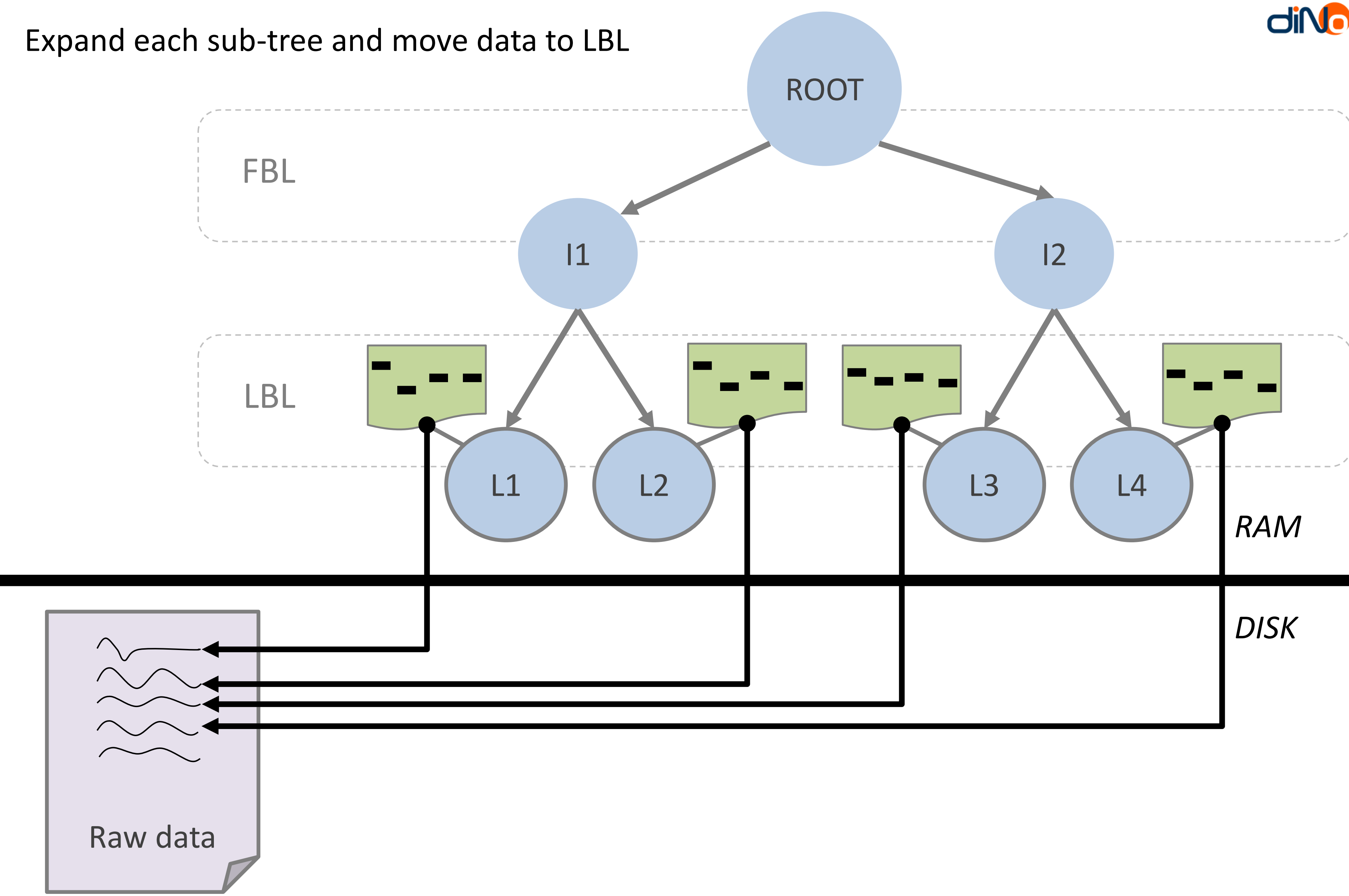
Discard raw data and keep pointer to raw file

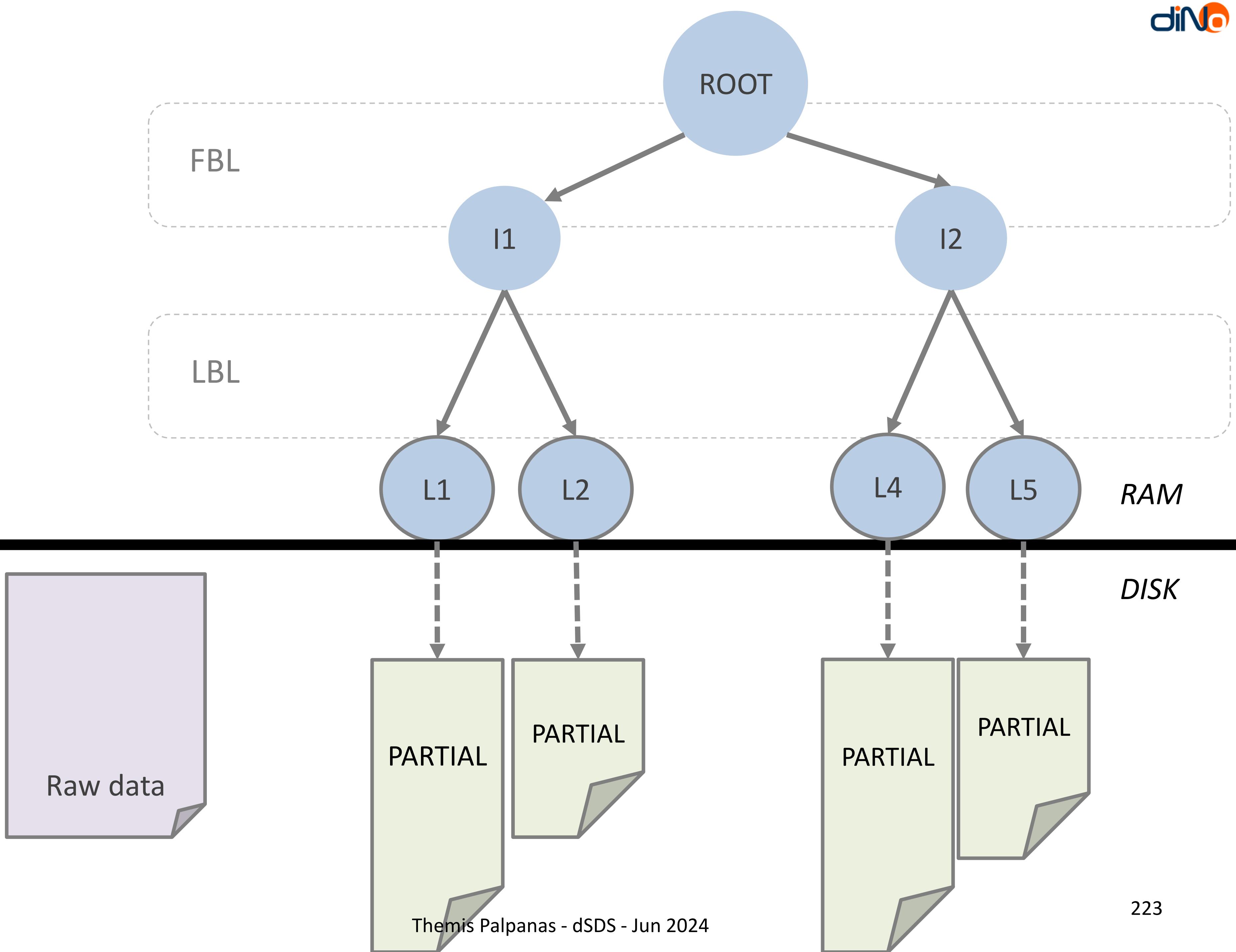


Continue loading data until we run out of memory

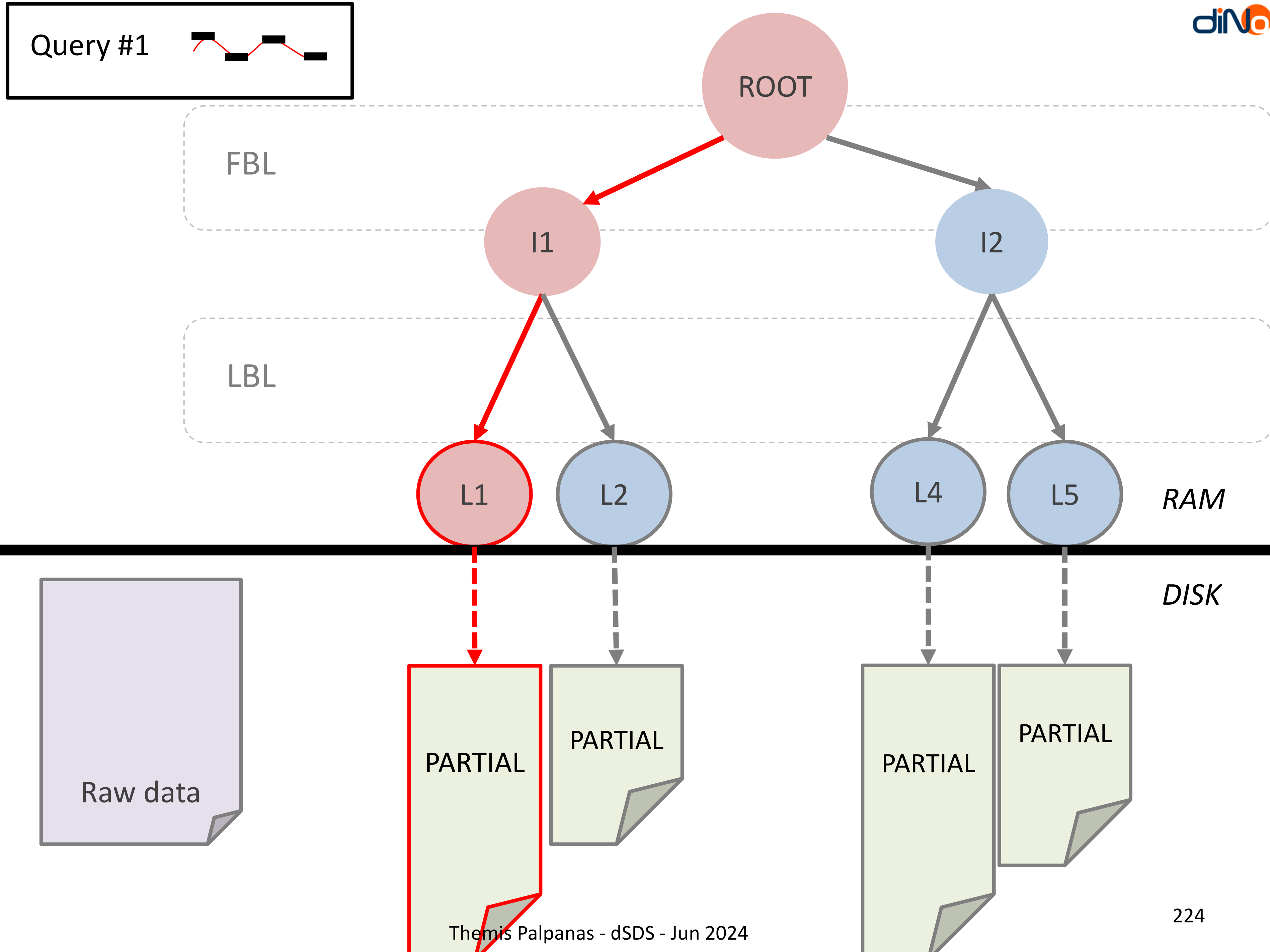


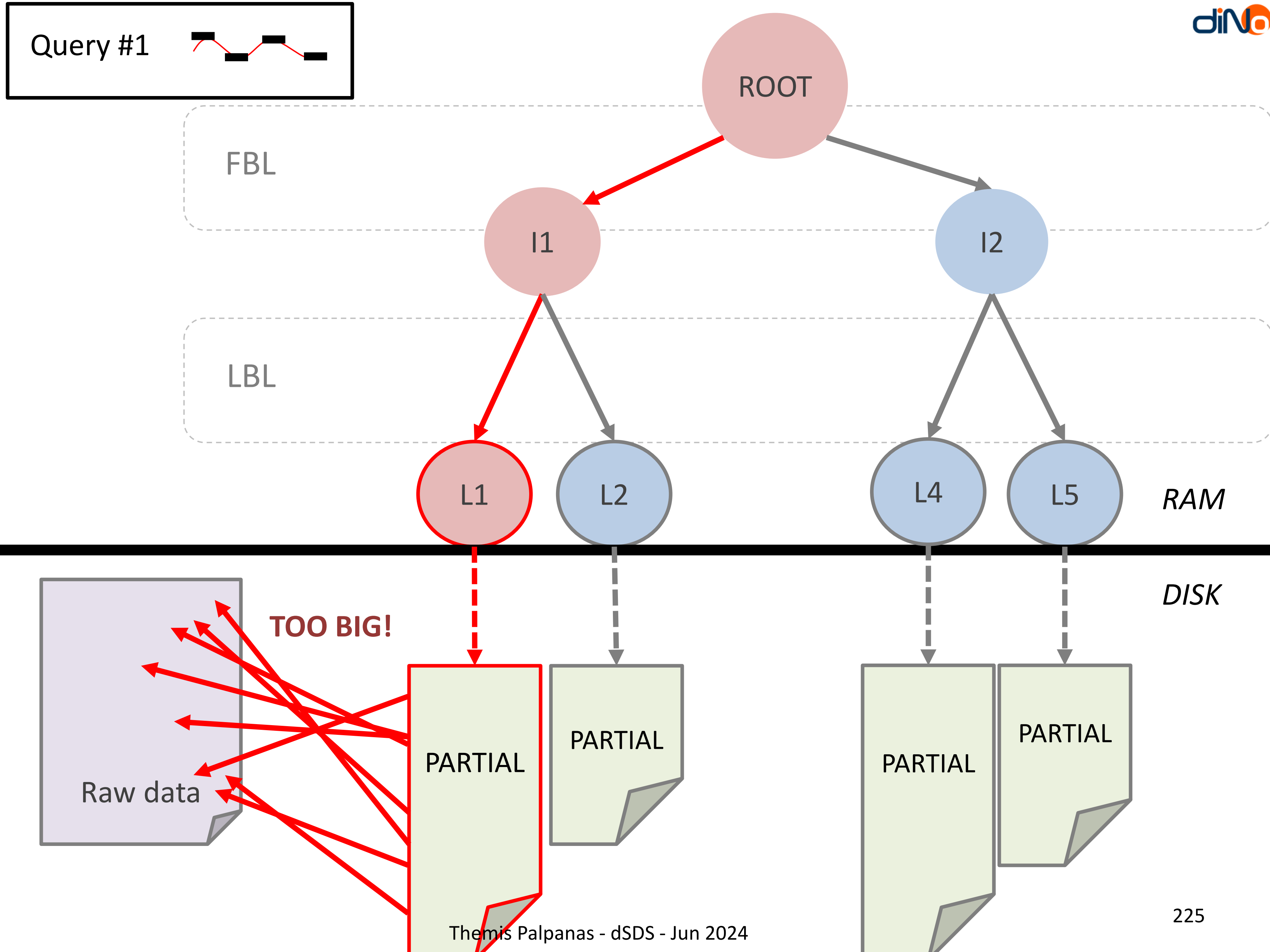
Expand each sub-tree and move data to LBL





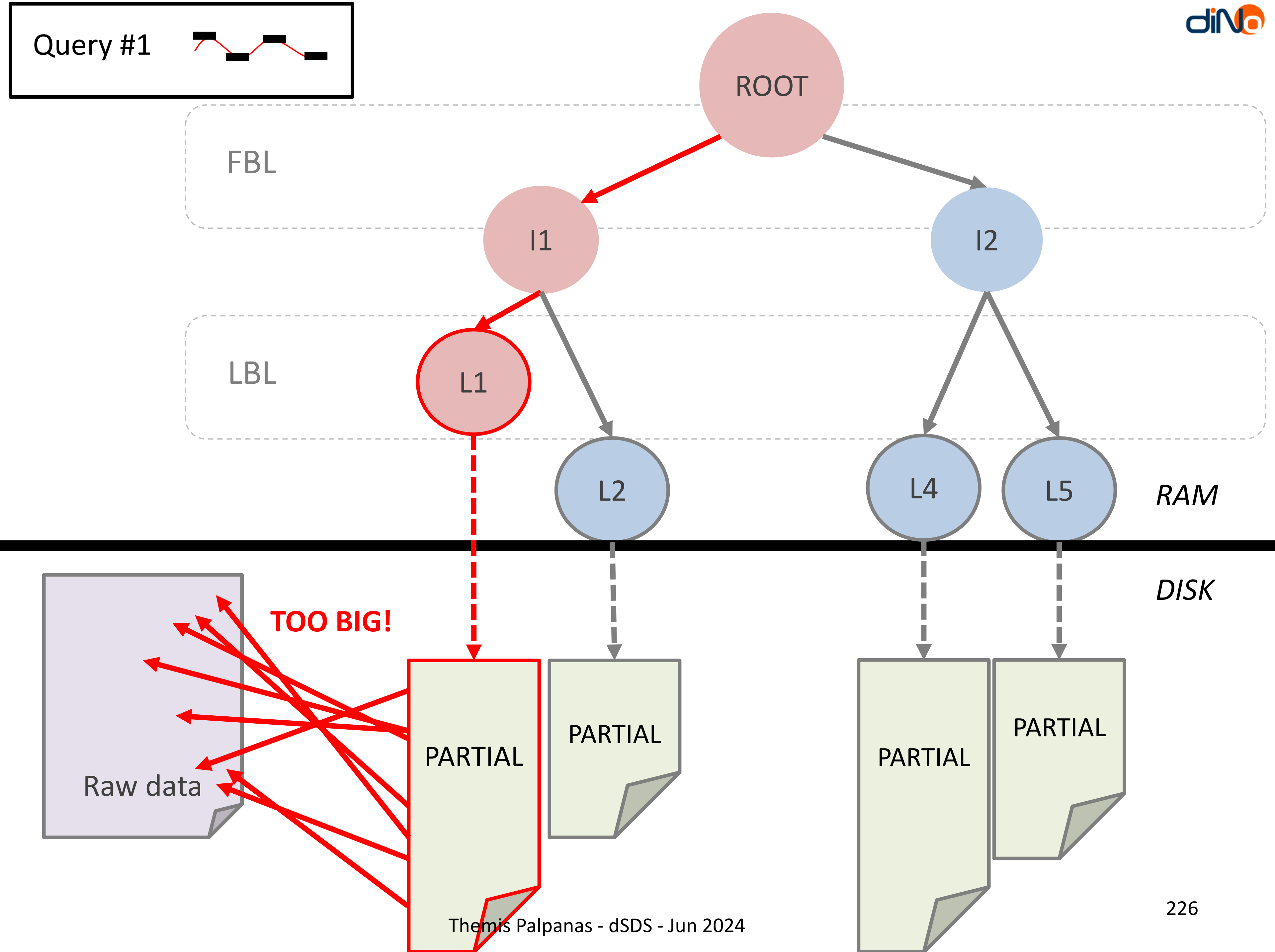
ROOT
FBL
I1
I2
LBL
L1
L2
L4
L5
RAM
DISK
Raw data
PARTIAL
PARTIAL
PARTIAL
PARTIAL


Query #1
ROOT
FBL
I1
I2
LBL
L1
L2
L4
L5
RAM
DISK
Raw data
PARTIAL
PARTIAL
PARTIAL
PARTIAL



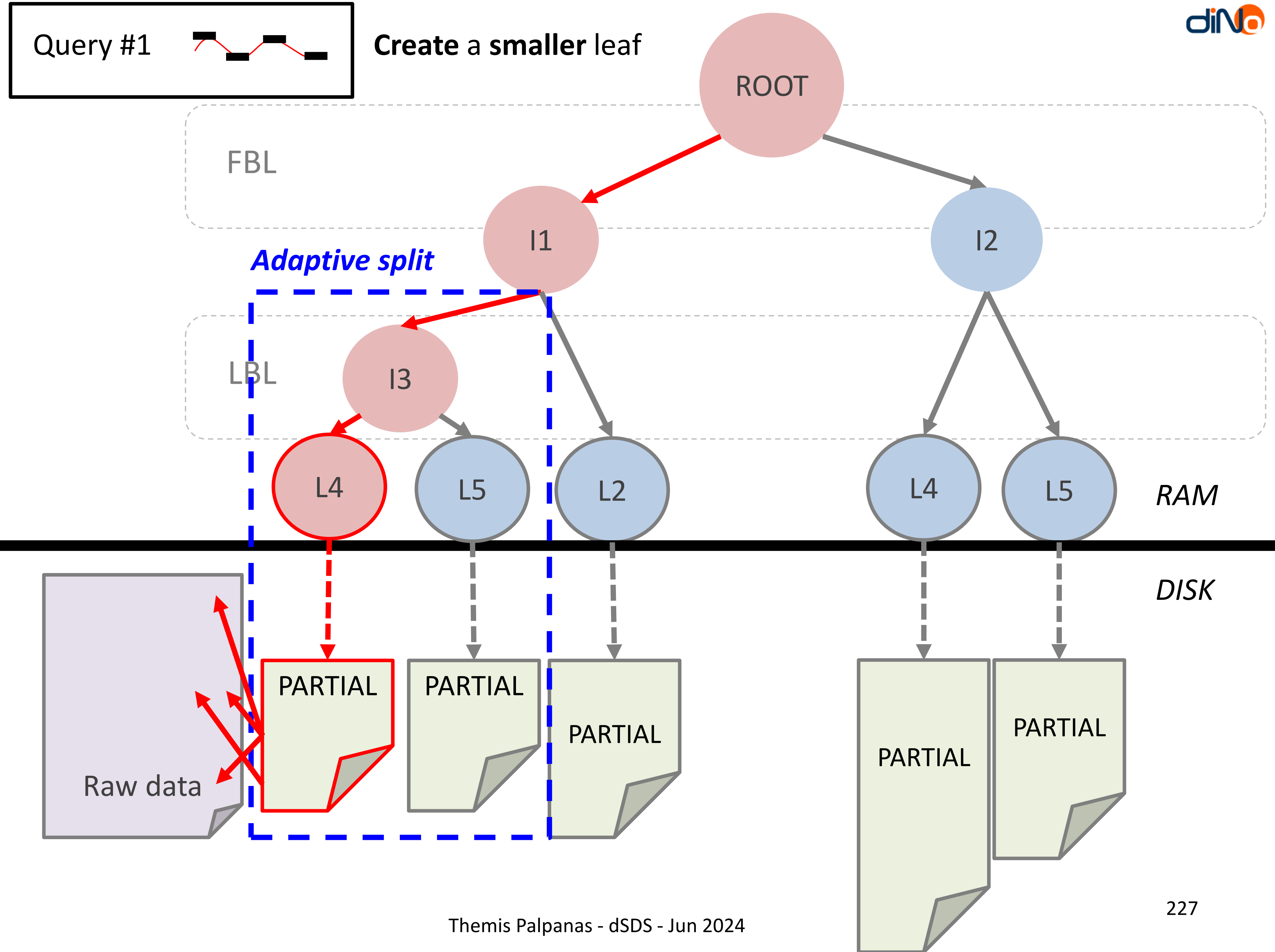
Query #1
ROOT
FBL
I1
I2
LBL
L1
L2
L4
L5
RAM
DISK
TOO BIG!
Raw data
PARTIAL
PARTIAL
PARTIAL
PARTIAL

Query #1


ROOT
FBL
I1
I2
LBL
L1
L2
L4
L5
RAM
DISK
TOO BIG!
Raw data
PARTIAL
PARTIAL
PARTIAL
PARTIAL

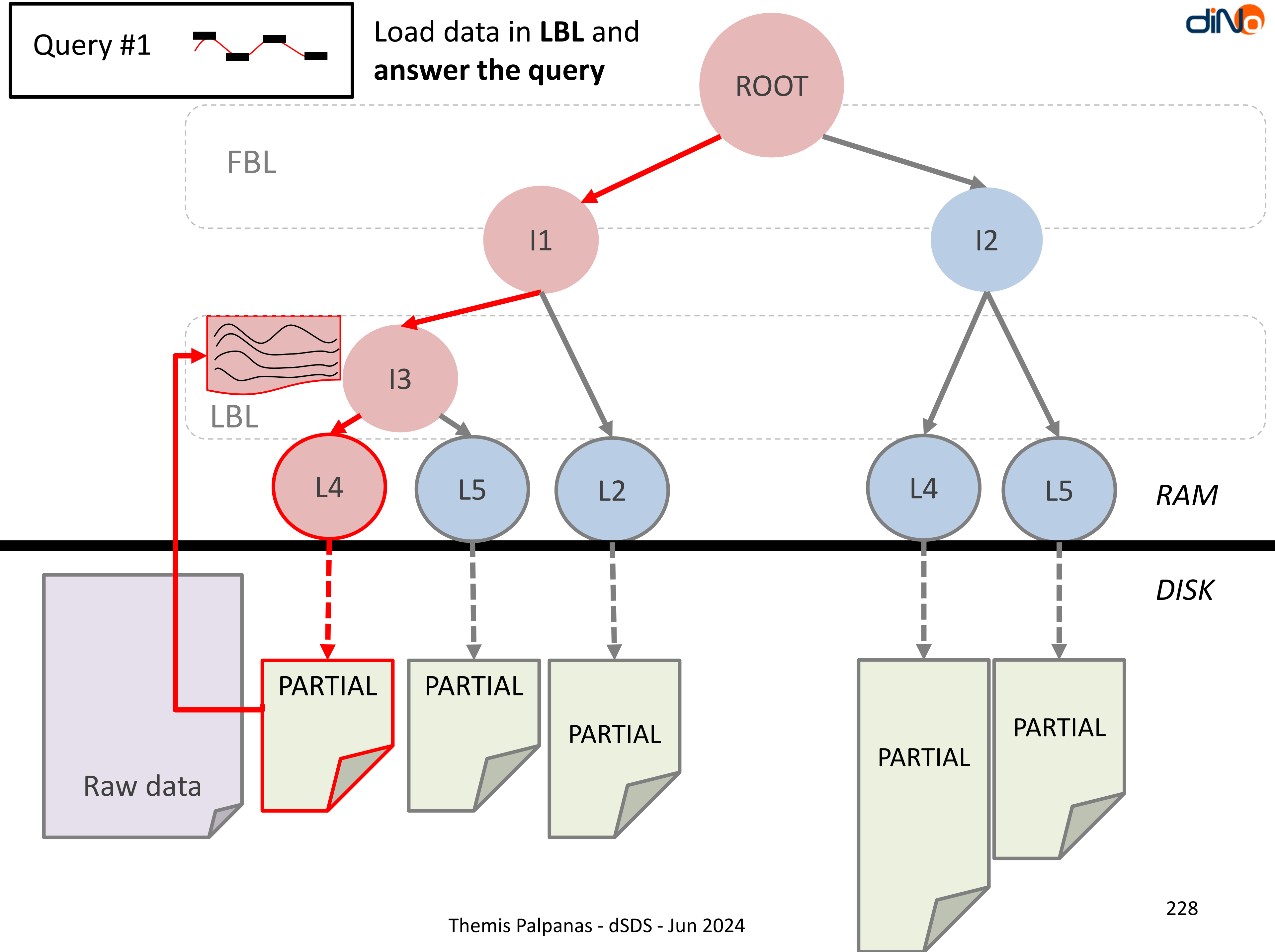
Query #1

Create a **smaller** leaf

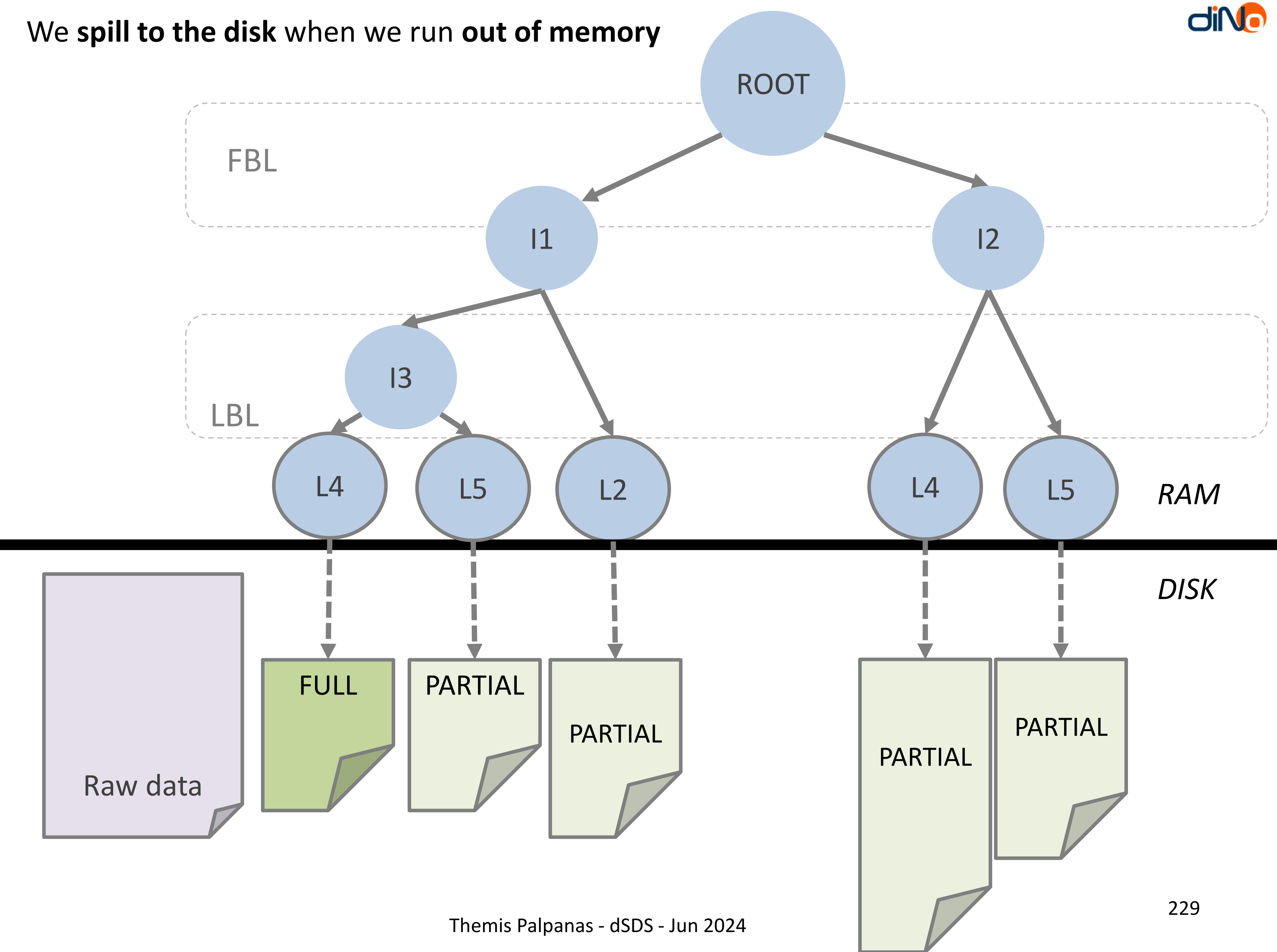


Query #1

Load data in **LBL** and **answer the query**



We **spill to the disk** when we run **out of memory**



Parallelization/Distribution?

Parallelization/Distribution?

- discussion so far assumed serial execution in a single core
  - focus on efficient resource utilization
  - squeeze the most out of a single core
  - produce scalable solutions at lowest possible cost
    - also suitable for analysts with no access to/expertise for clusters

Publications

HPCS'17

Need for Parallelization/Distribution

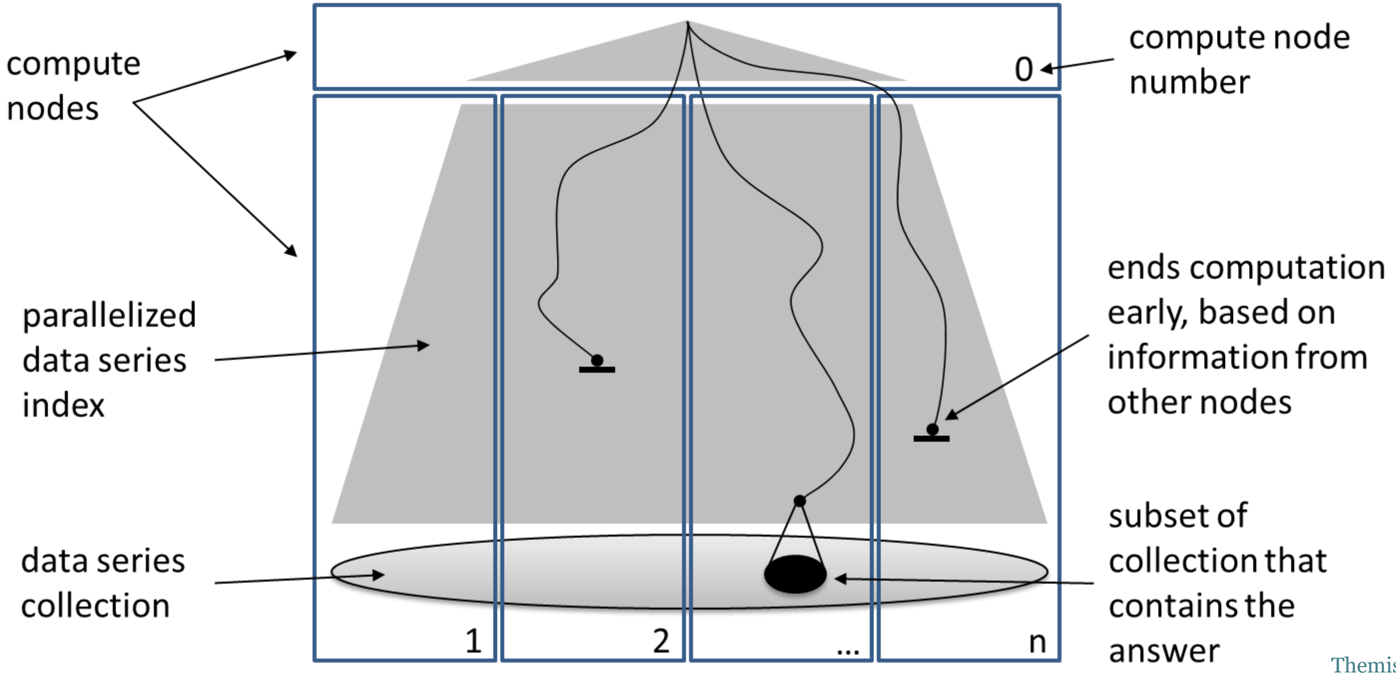
- take advantage of modern hardware!
  - Single Instruction Multiple Data (SIMD)
    - natural for data series operations
  - multi-tier CPU caches
    - design data structures aligned to cache lines
  - multi-core and multi-socket architectures
    - use parallelism inside each computation server
 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

Publications

HPCS'17

Need for Parallelization/Distribution

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



Need for Parallelization/Distribution

Publications

ICDM'17

TKDE'18

PKDD'19

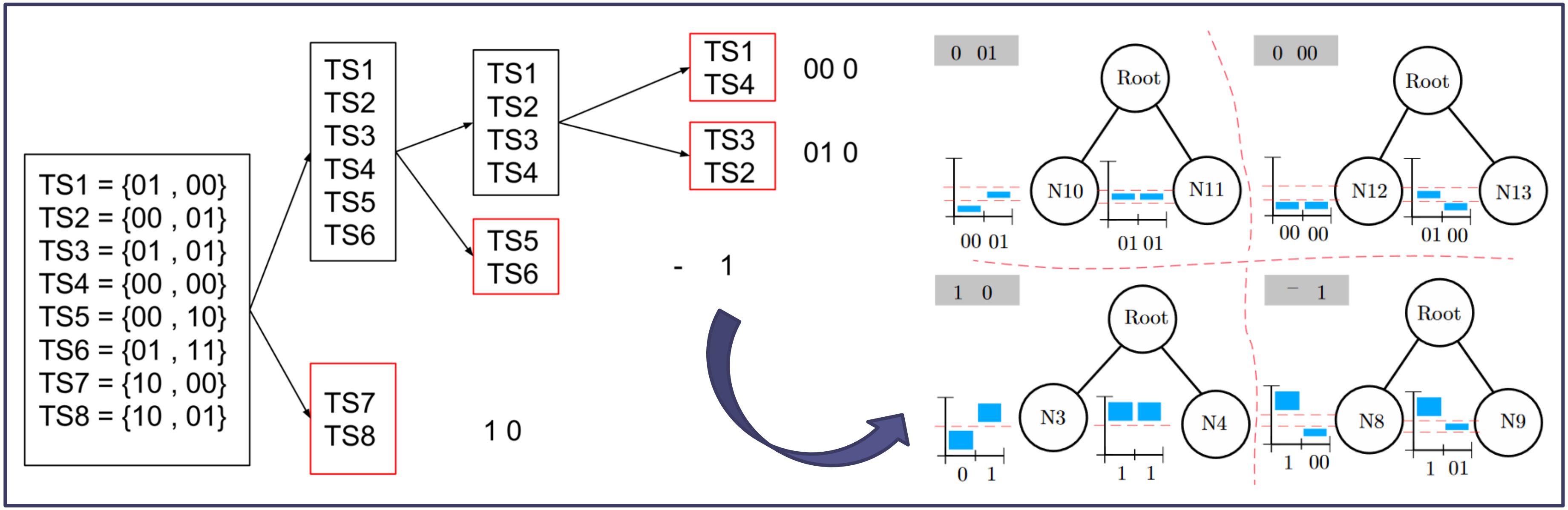
- DPiSAX: current solution for distributed processing (Spark)
  - balances work of different worker nodes

Need for Parallelization/Distribution

Publications

- ICDM'17
- TKDE'18
- PKDD'19

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



Publications

ICDM'17

TKDE'18

PKDD'19

Need for Parallelization/Distribution

- DPiSAX: current solution for distributed processing (Spark)
  - balances work of different worker nodes
  - performs 2 orders of magnitude faster than centralized solution

Need for Parallelization/Distribution

Publications

- ICDM'17
- TKDE'18
- PKDD'19
- BigData'18

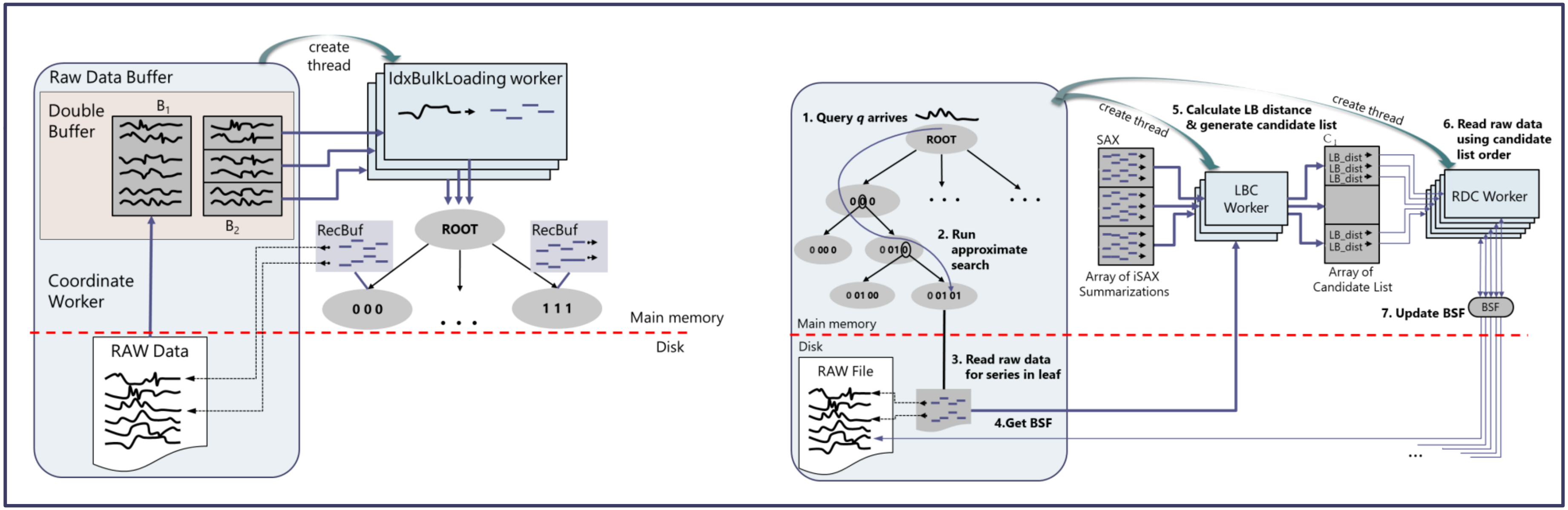
- DPiSAX: current solution for distributed processing (Spark)
  - balances work of different worker nodes
  - performs 2 orders of magnitude faster than centralized solution
- ParIS: current single-node parallel solution
  - masks out the CPU cost

Need for Parallelization/Distribution

Publications

- ICDM'17
- TKDE'18
- PKDD'19
- BigData'18

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



Need for Parallelization/Distribution

Publications

- ICDM'17
- TKDE'18
- PKDD'19
- BigData'18

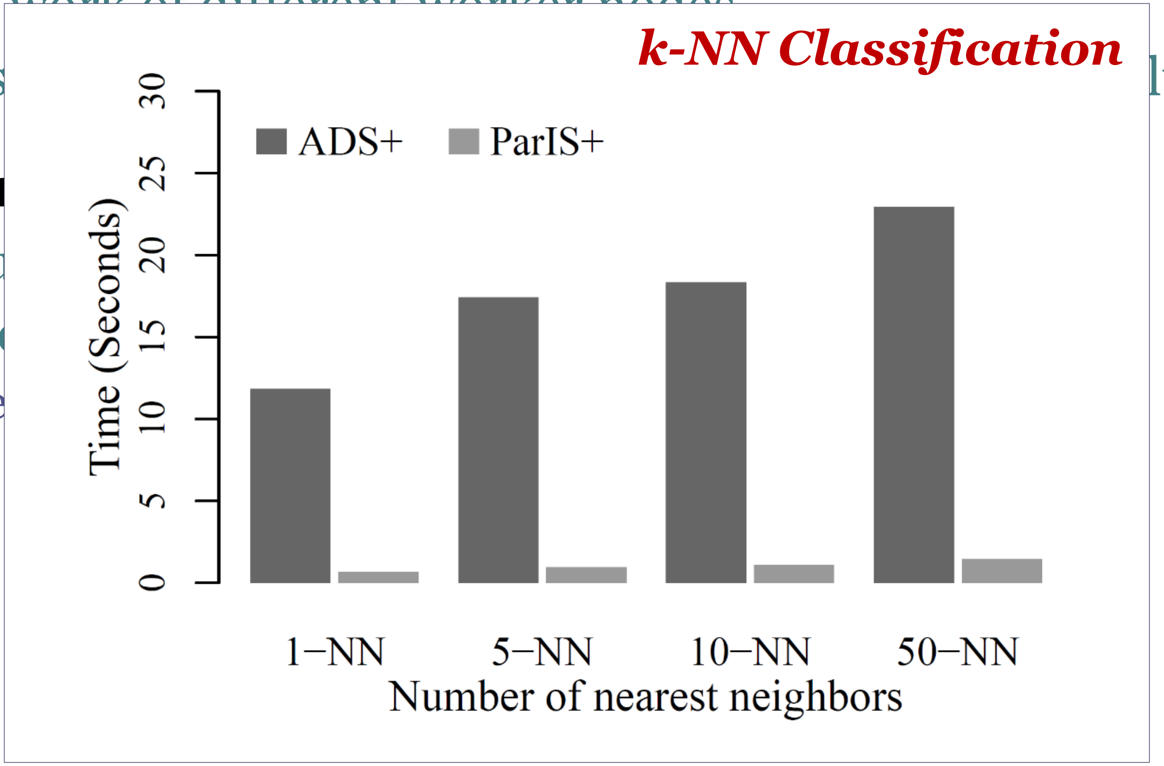
- DPiSAX: current solution for distributed processing (Spar
  - balances work of different worker nodes
  - performs 2 orders of magnitude faster than centralized solution
- ParIS: current single-node parallel solution
  - masks out the CPU cost
  - answers exact queries in the order of a few secs
    - >1 order of magnitude faster then single-core solutions

Need for Parallelization/Distribution

Publications

- ICDM'17
- TKDE'18
- PKDD'19
- BigData'18

- DPiSAX: current solution for distributed processing (Spar
  - balances work of different worker nodes
  - performs ... lution
- ParIS: curr
  - masks ou
  - answers
    - >1 orde

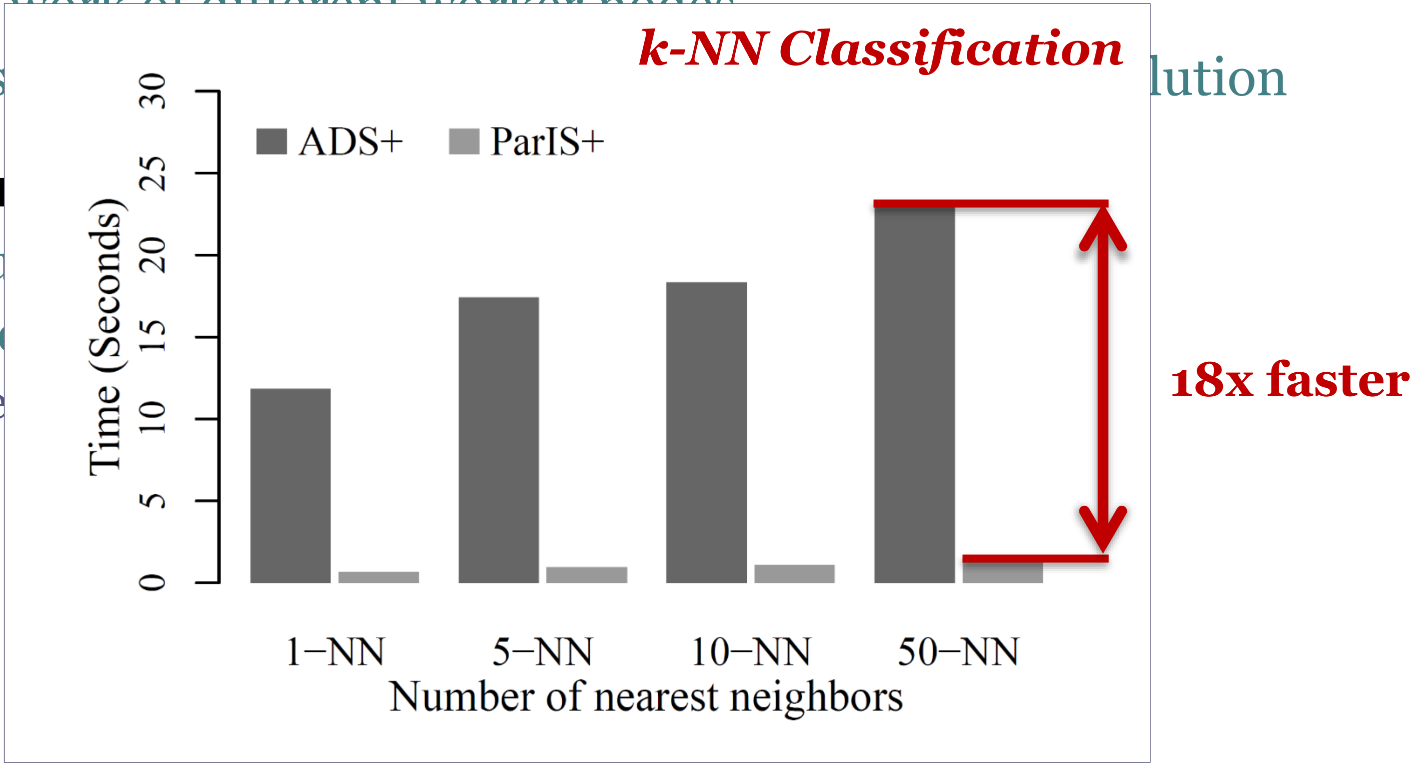


Need for Parallelization/Distribution

Publications

- ICDM'17
- TKDE'18
- PKDD'19
- BigData'18

- DPiSAX: current solution for distributed processing (Spar
  - balances
  - performs … lution
- ParIS: curr
  - masks ou
  - answers
    - >1 orde

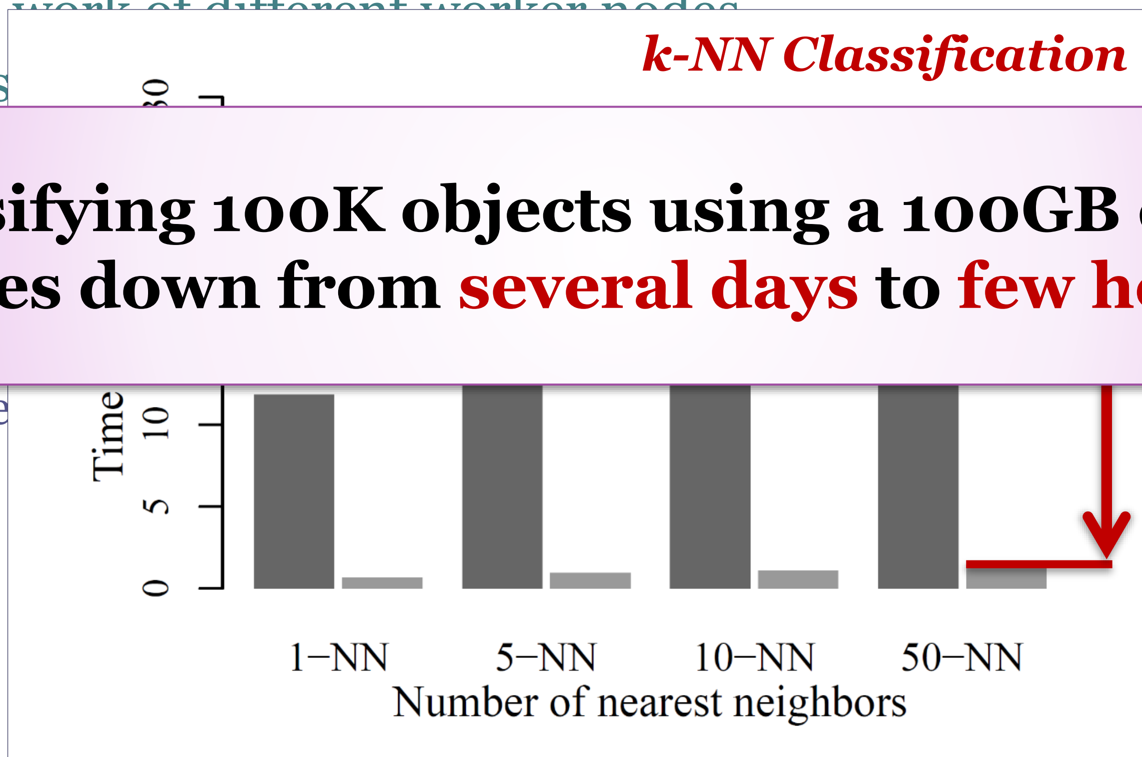


Need for Parallelization/Distribution

Publications

- ICDM'17
- TKDE'18
- PKDD'19
- BigData'18

- DPiSAX: current solution for distributed processing (Spar
  - balances
  - performs … lution
- Pa
  -
  -
    - >1 orde



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

Need for Parallelization/Distribution

Publications

- ICDM'17
- TKDE'18
- PKDD'19
- BigData'18
- ICDE'20

- DPiSAX: current solution for distributed processing (Spark)
  - balances work of different worker nodes
  - performs 2 orders of magnitude faster than centralized solution
- ParIS: current single-node parallel solution
  - masks out the CPU cost
  - answers exact queries in the order of a few secs
    - >1 order of magnitude faster then single-core solutions
- MESSI: current single-node parallel solution + in-memory data
  - answers exact queries at interactive speeds: ~50msec on 100GB
- SING: current single-node parallel solution + GPU + in-memory data
  - answers exact queries at interactive speeds: ~32msec on 100GB

Publications

Chatzakis-PVLDB'23

Odyssey
Distributed, Parallel, In-Memory Indexing

- Distributed, in-memory solution for SIMD, multi-core, multi-socket architectures
  - allows in-memory processing (across machines) of very large datasets

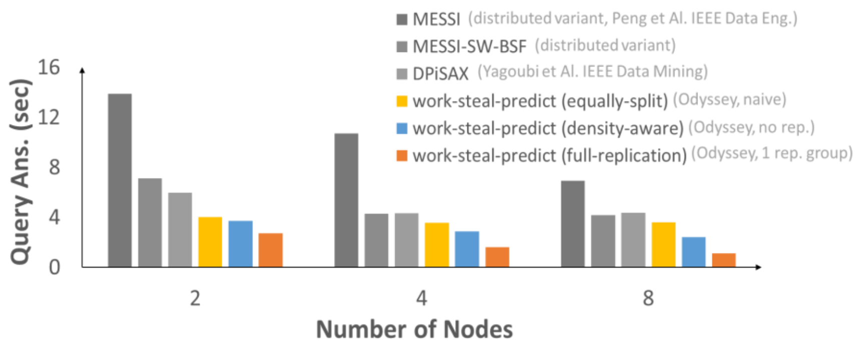
- Odyssey addresses the following challenges
  - **Query Scheduling**: Which queries to which nodes?
    - **Query Execution Time estimations**
    - **Flexible Replication Schemes**
    - **Dynamic Scheduling**
  - **Load Balancing**: Enable nodes to perform useful and equal work
    - **Density-aware Data Distribution**
    - **Efficient work-stealing**

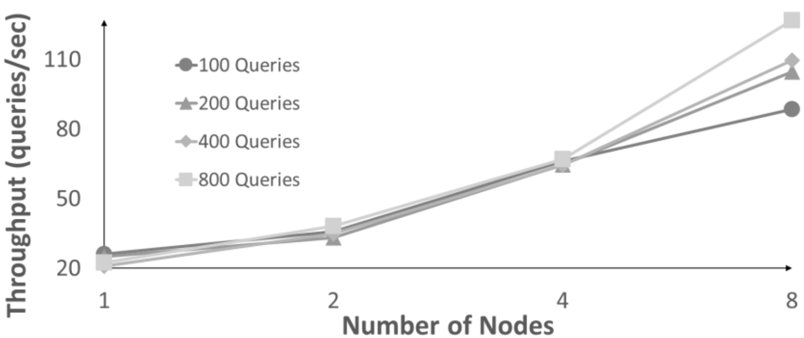
Odyssey

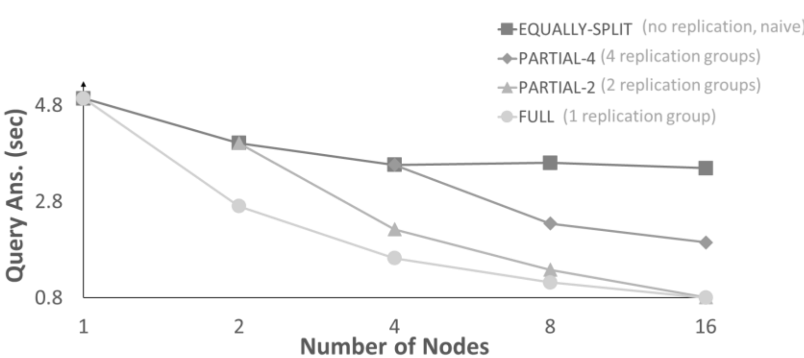
- achieves all goals

Publications

Chatzakis-PVLDB’23



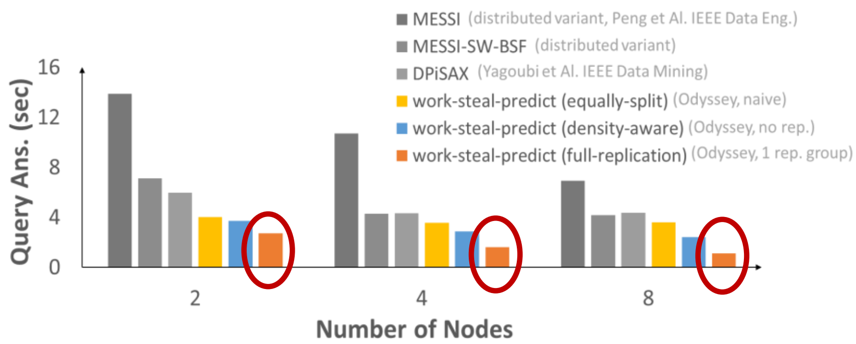


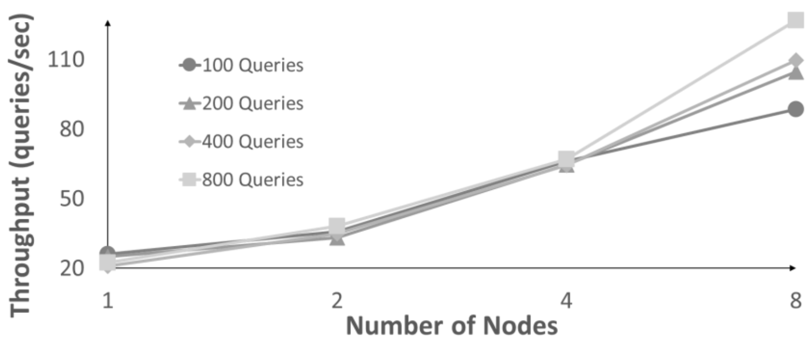


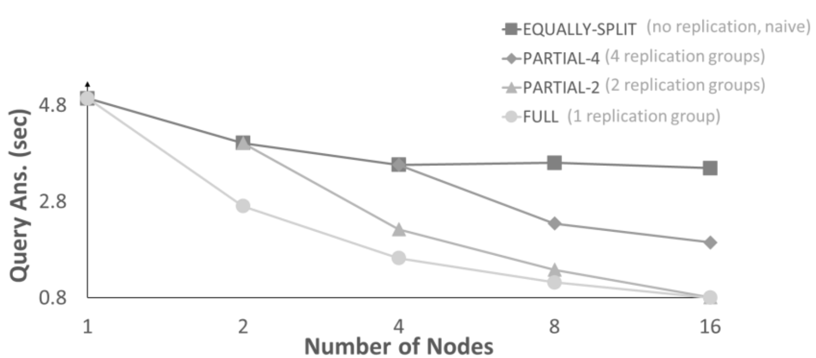
Odyssey

- achieves all goals

Publications
Chatzakis-PVLDB'23







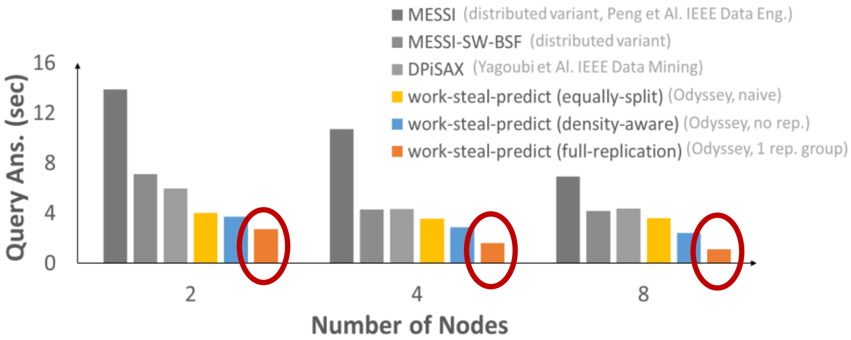
up to 3x faster
than best competitor

Odyssey

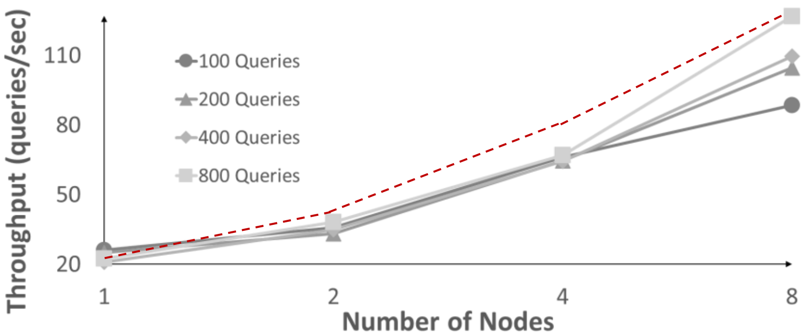
Publications

Chatzakis-PVLDB'23

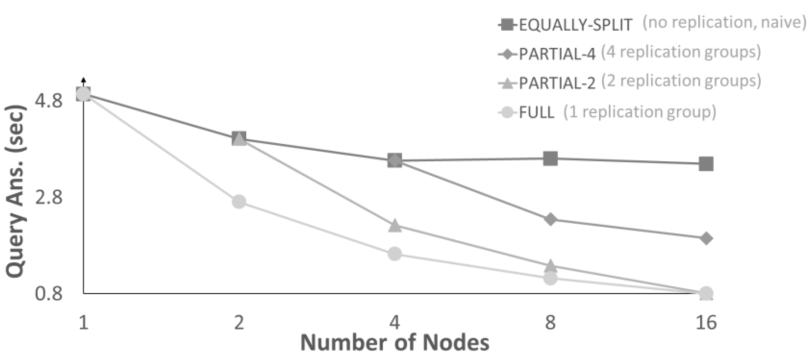
- achieves all goals



up to 3x faster than best competitor



scalable query answering (almost linear)

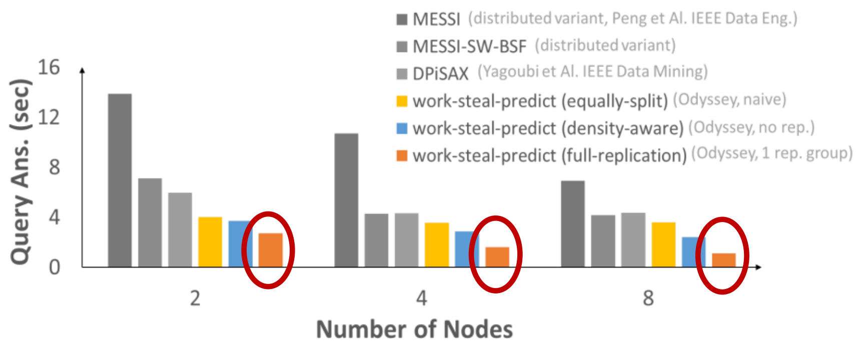


Odyssey

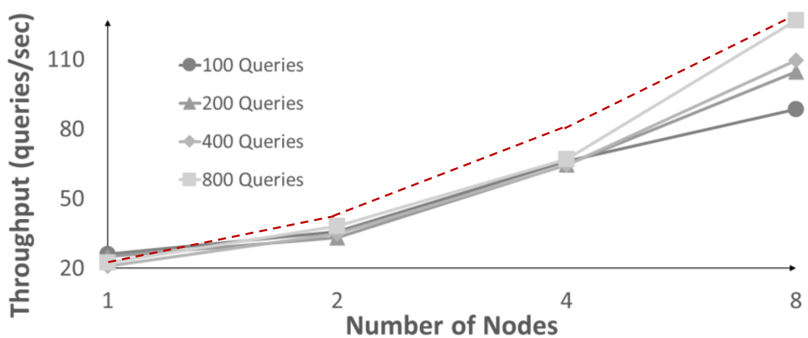
- achieves all goals

Publications

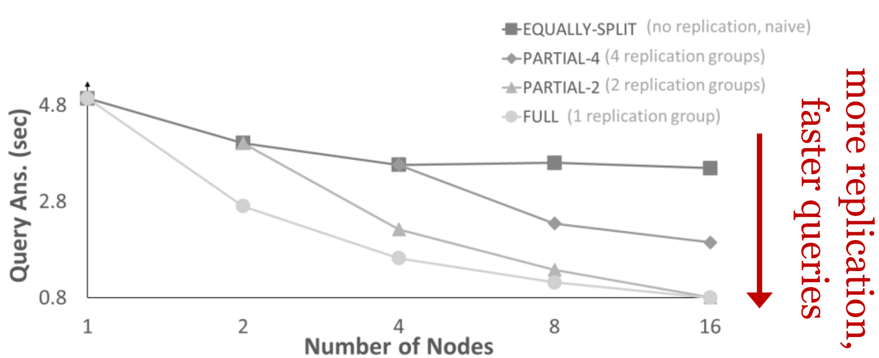
Chatzakis-PVLDB'23



up to 3x faster than best competitor



scalable query answering (almost linear)

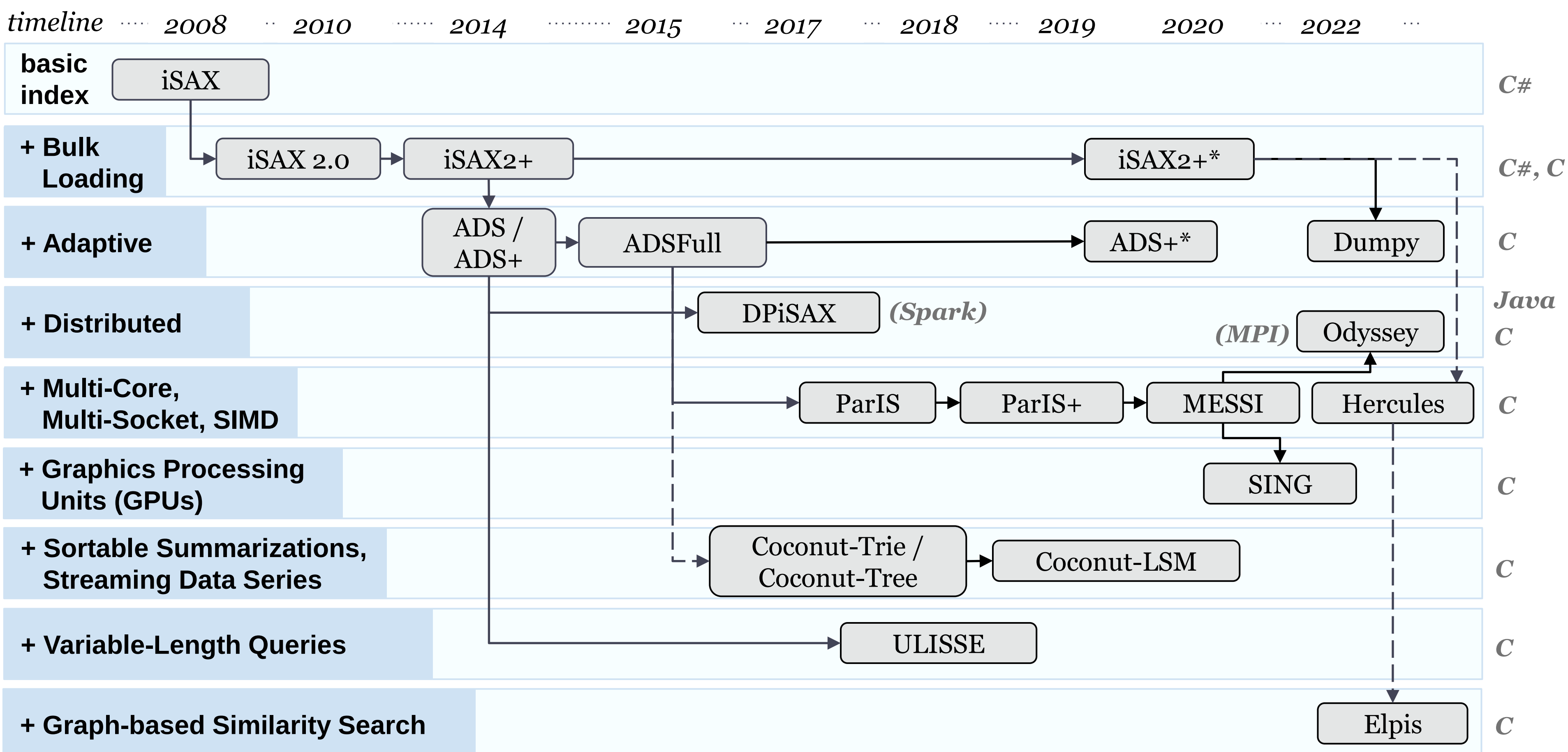


more replication leads to faster query answering

iSAX Index Family Lineage Tree

Publications

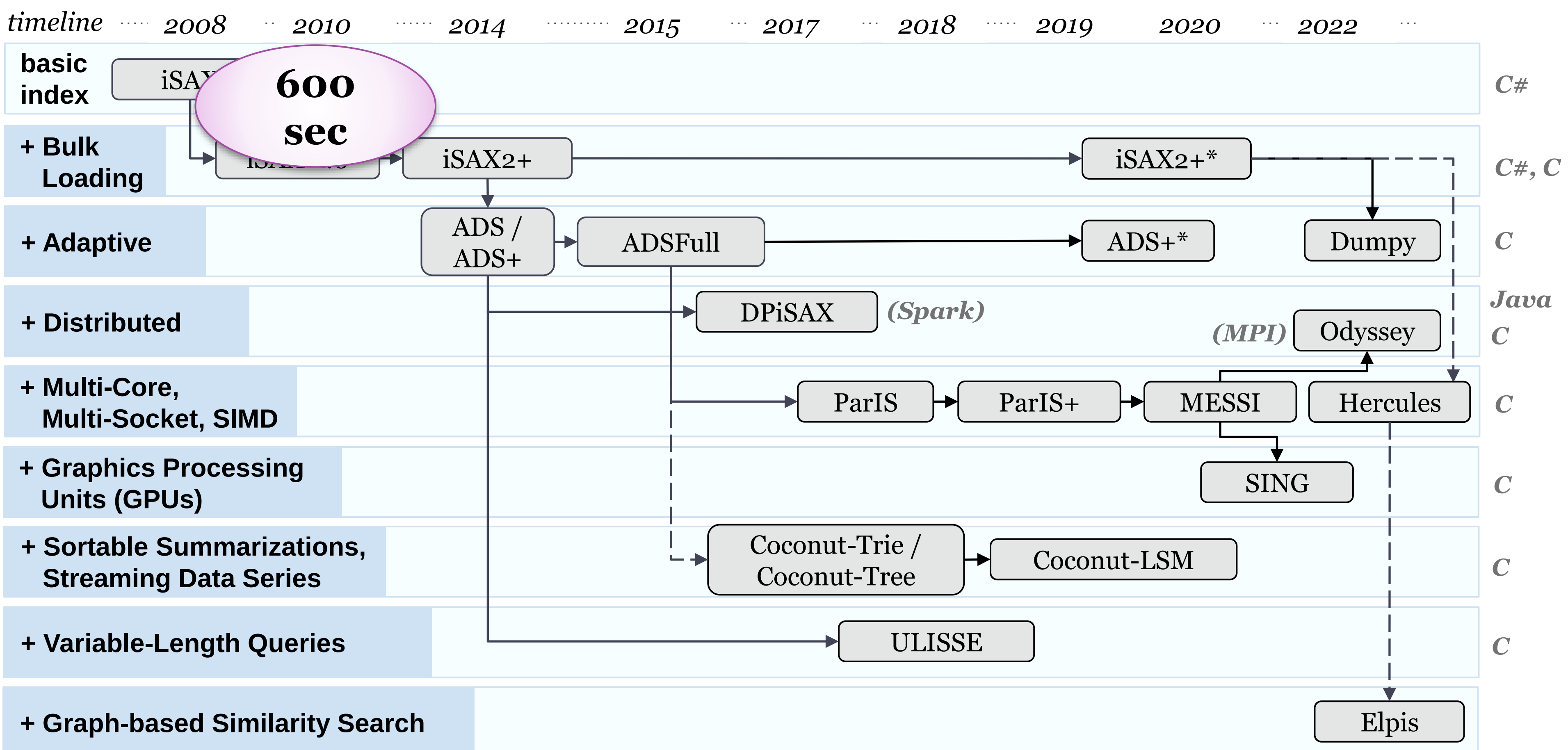
Palpanas-ISIP'19



iSAX Index Family Lineage Tree

Publications

Palpanas-ISIP‘19

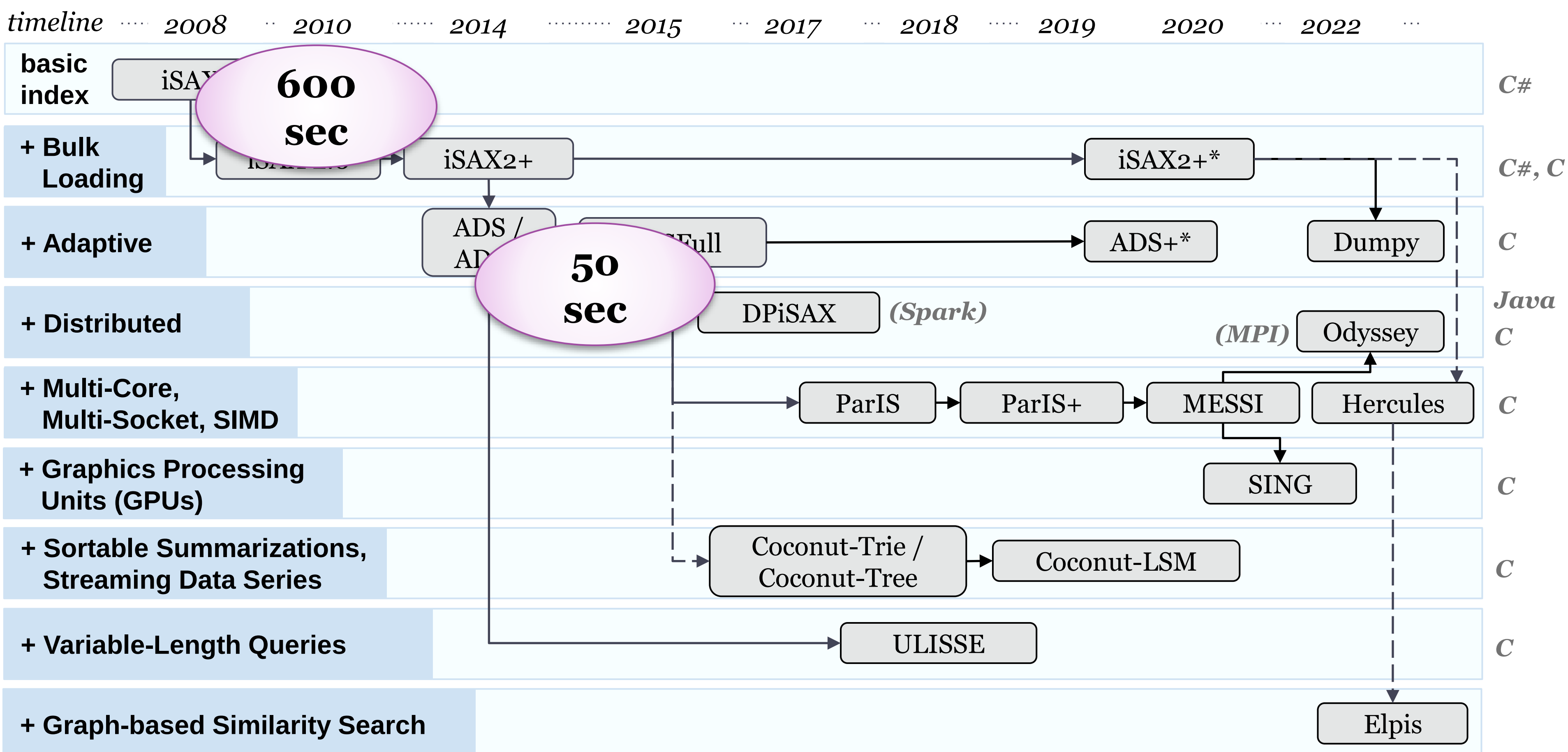


execution time for **1 similarity search query on a 100GB dataset _on disk_**

iSAX Index Family Lineage Tree

Publications

Palpanas-ISIP'19

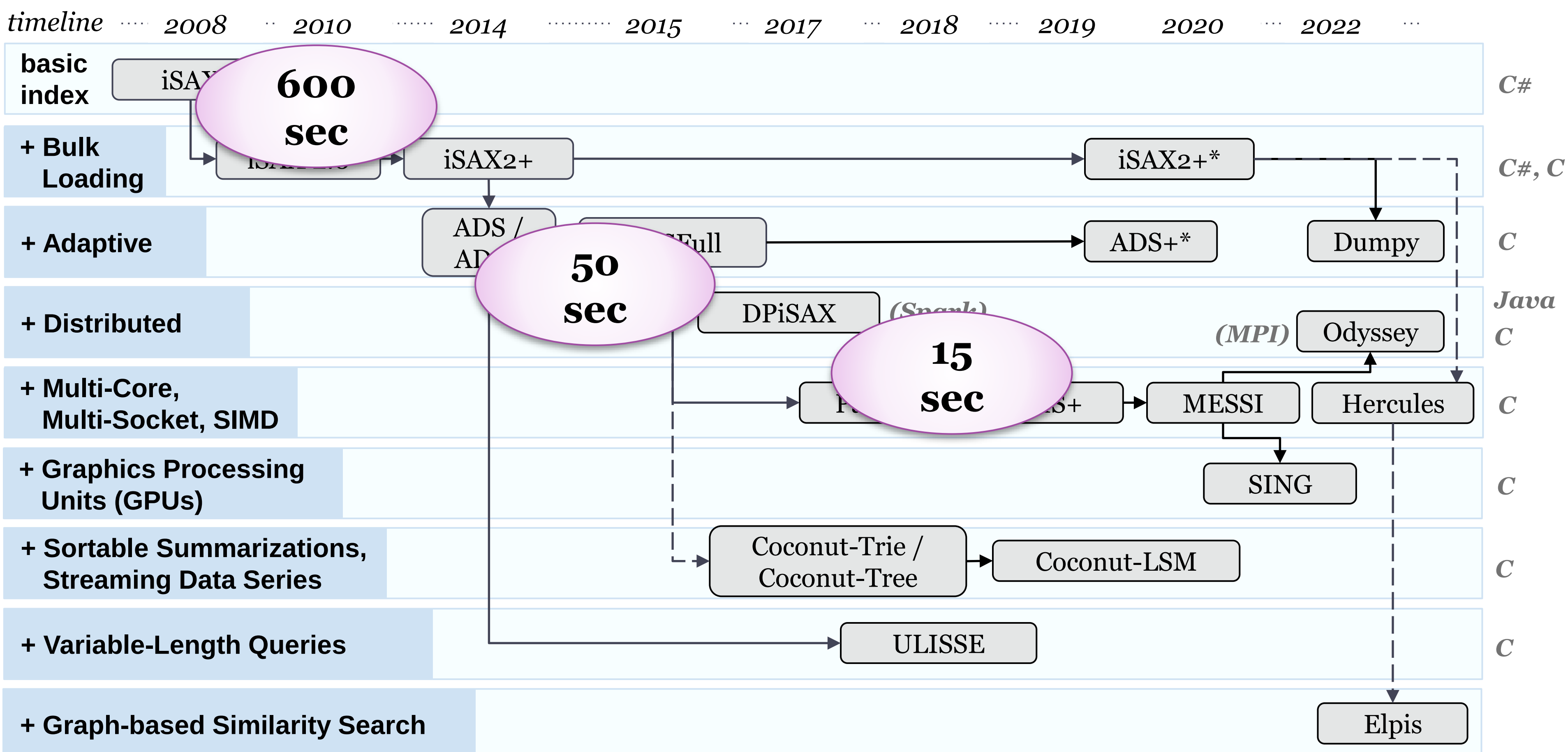


execution time for **1 similarity search query on a 100GB dataset *on disk***

iSAX Index Family Lineage Tree

Publications

Palpanas-ISIP‘19

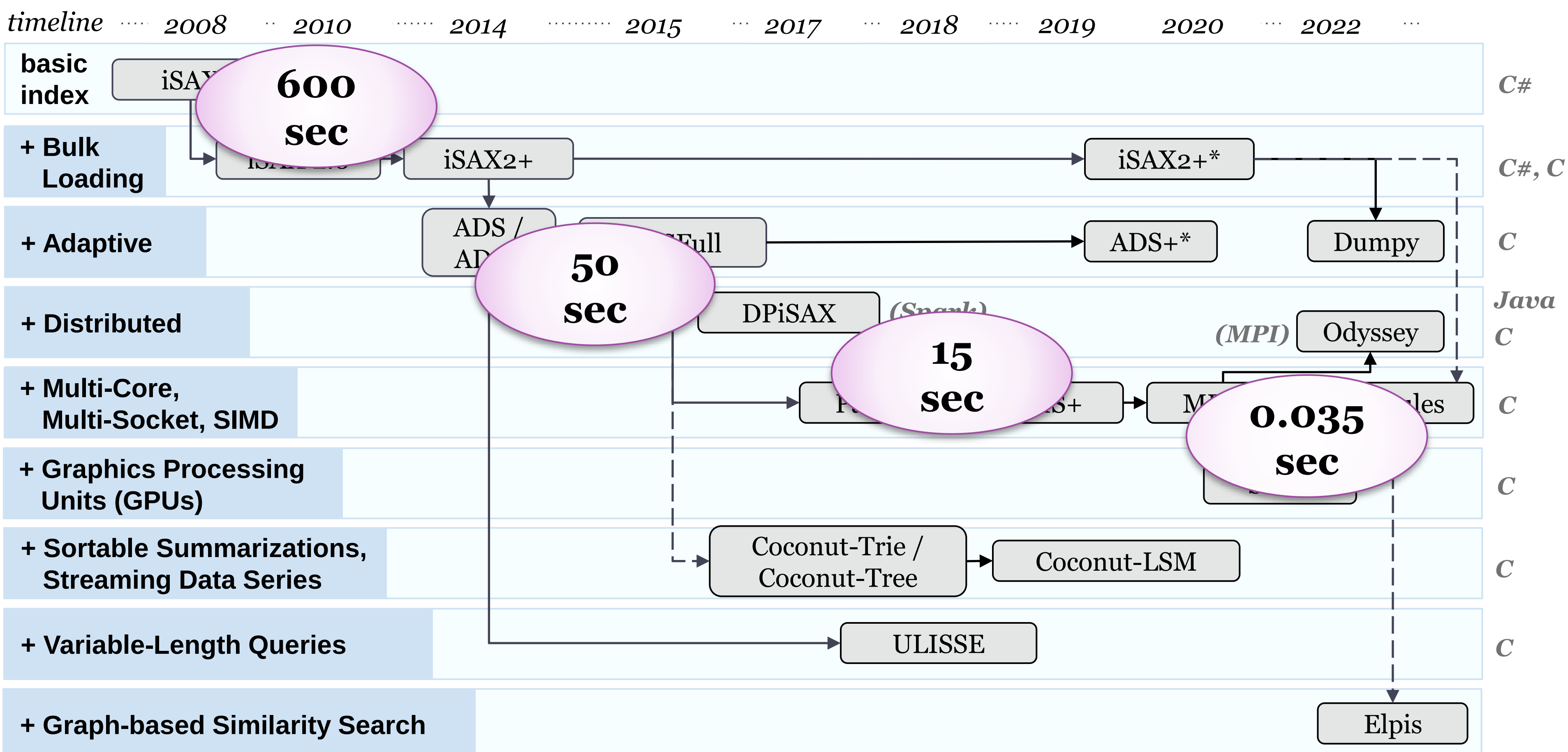


execution time for **1 similarity search query on a 100GB dataset *on disk***

iSAX Index Family Lineage Tree

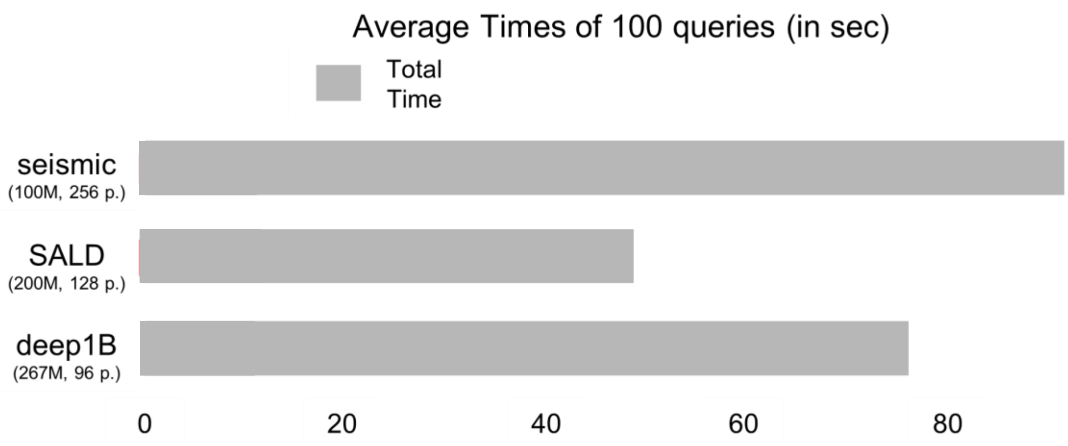
Publications

Palpanas-ISIP‘19

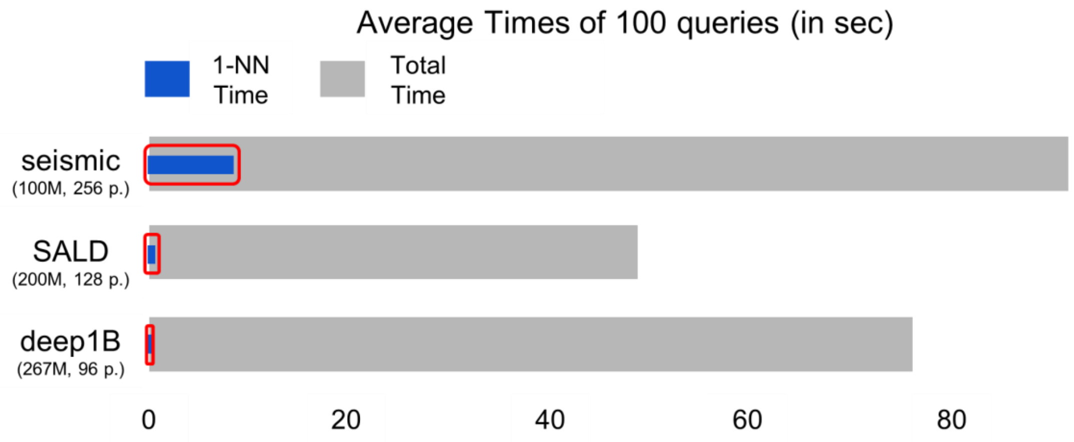


execution time for **1 similarity search query on a 100GB dataset *in memory***

Further Advances

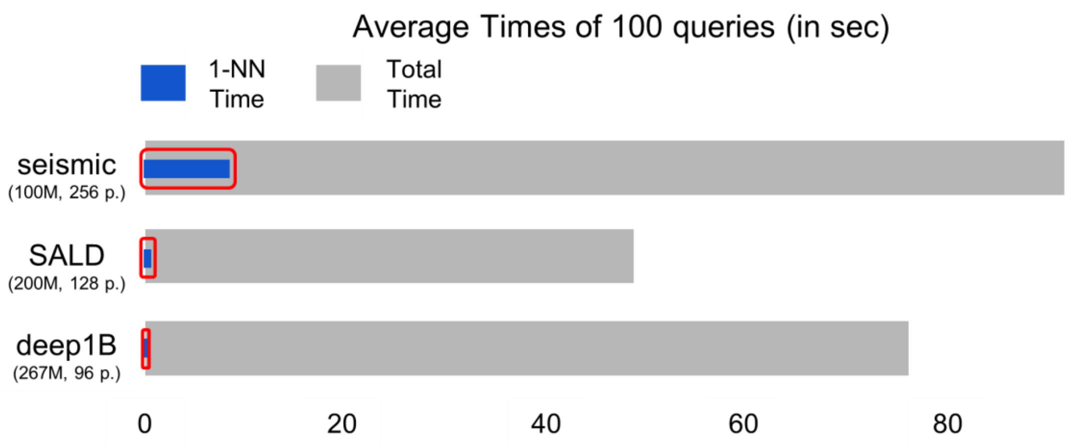


Further Advances



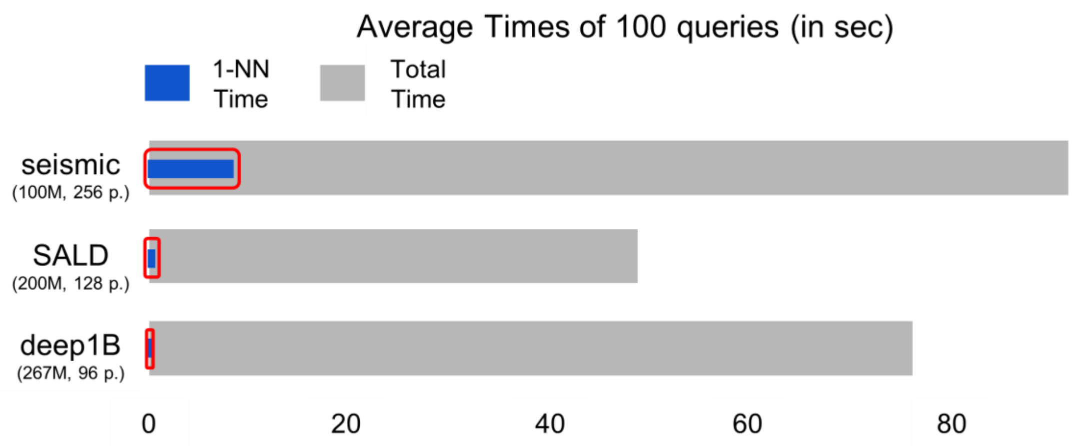
- how do we further reduce the wasted (gray) effort?

Further Advances



- how do we further reduce the wasted (gray) effort?
  - **progressive query answering**
    - produce intermediate answers with (probabilistic) quality guarantees

Further Advances



- how do we further reduce the wasted (gray) effort?
  - **progressive query answering**
    - produce intermediate answers with (probabilistic) quality guarantees
  - **learned summarizations + index structures**
    - adapt to data characteristics
    - build more efficient indexes
    - perform more effective pruning

Publications

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

Publications

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
- everything we discussed applicable to high-d vectors, too!

Publications

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
- everything we discussed applicable to high-d vectors, too!
- several techniques for similarity search in high-d vectors
  - using LSH (SRS), space quantization (IMI), k-NN graphs (HNSW)

Publications

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
- everything we discussed applicable to high-d vectors, too!
- several techniques for similarity search in high-d vectors
  - using LSH (SRS), space quantization (IMI), k-NN graphs (HNSW)

sequences
text
images
video
graphs
…

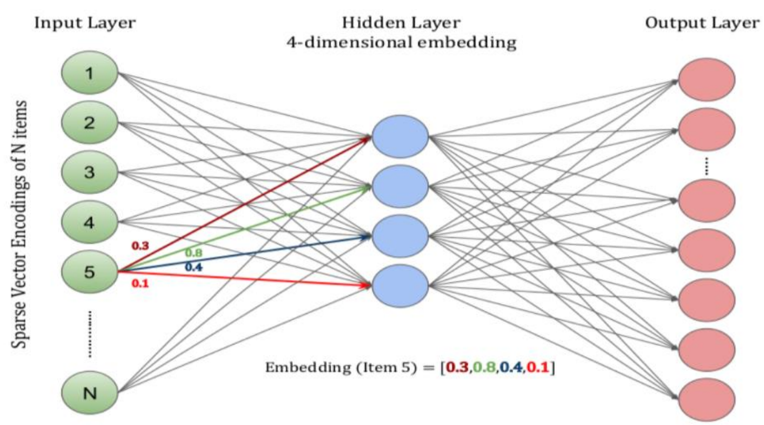
Publications

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
- everything we discussed applicable to high-d vectors, too!
- several techniques for similarity search in high-d vectors
  - using LSH (SRS), space quantization (IMI), k-NN graphs (HNSW)

sequences
text
images
video
graphs
...



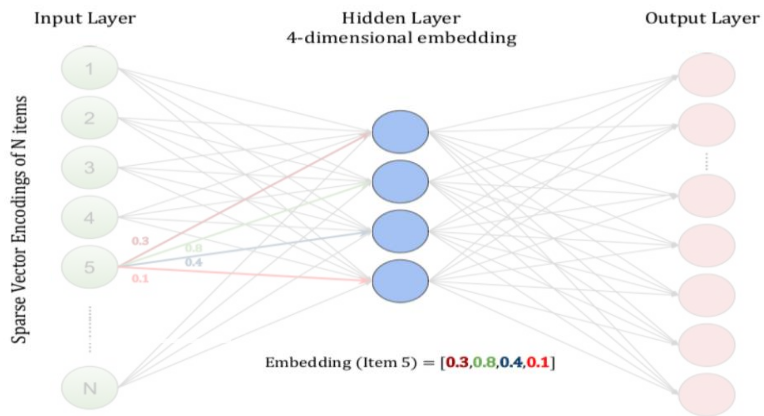
Data Series vs. high-d Vectors

Publications

WIMS'20

- two sides of the same(?) coin
  - data series as multidimensional points
  - for a specific ordering of the dimensions
- everything we discussed applicable to high-d vectors, too!
- several techniques for similarity search in high-d vectors
  - using LSH (SRS), space quantization (IMI), k-NN graphs (HNSW)

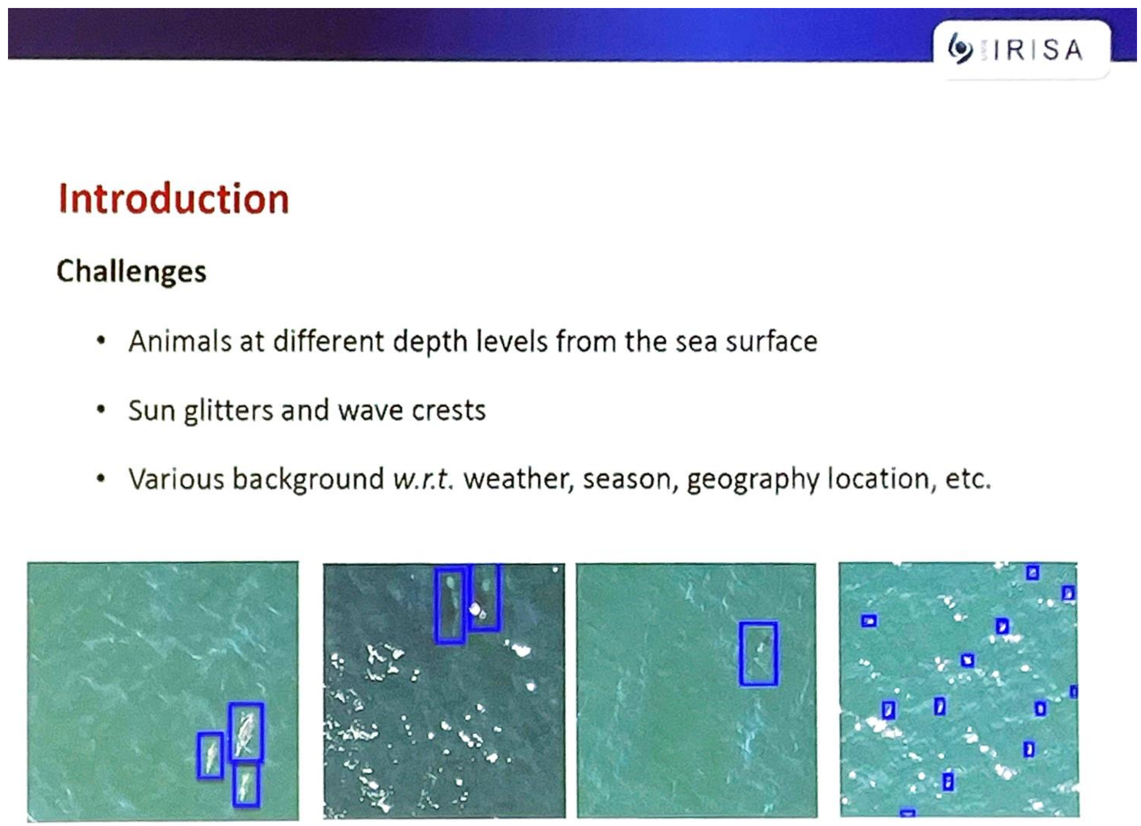
sequences
text
images
video
graphs
...



deep embeddings
high-d vectors learned using a DNN

High-d Vector Similarity Search Applications

- ocean life monitoring



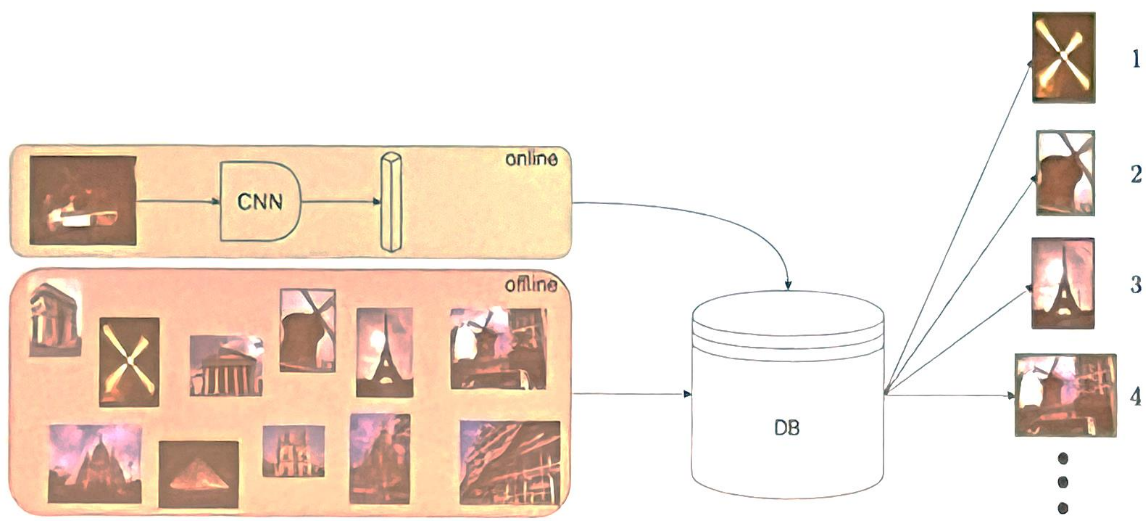
Allain and Pham, MACLEAN'22

High-d Vector Similarity Search Applications

- ocean life monitoring
- image retrieval

Image Retrieval: the task

Given a query image, rank images of a database from most to least similar.



Ramzi et al., Cap&RFIAP'22

High-d Vector Similarity Search Applications

- ocean life monitoring
- image retrieval
- recommendations
- entity matching
- fraud detection
- …

Ramzi et al., Cap&RFIAP'22

Publications

WIMS'20

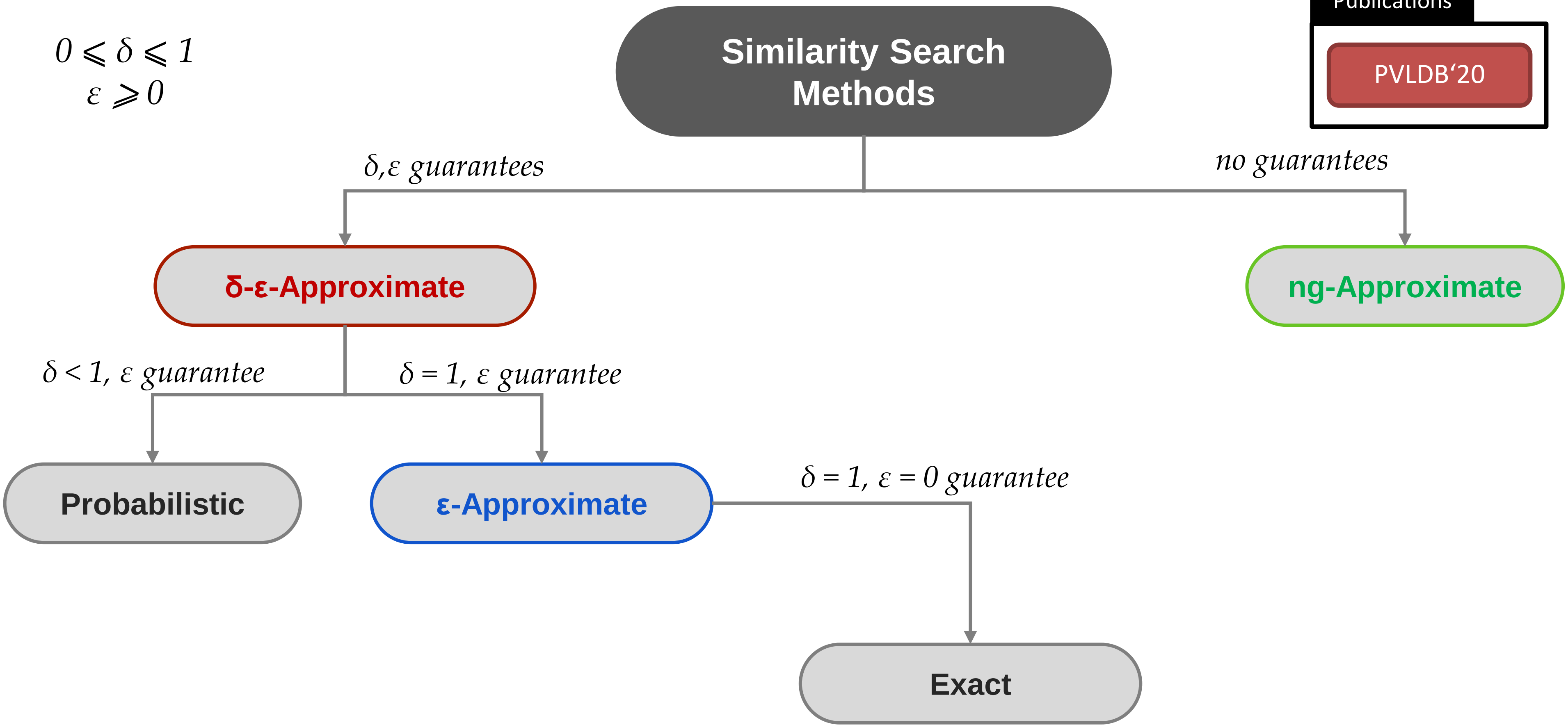
Data Series vs. high-d Vectors

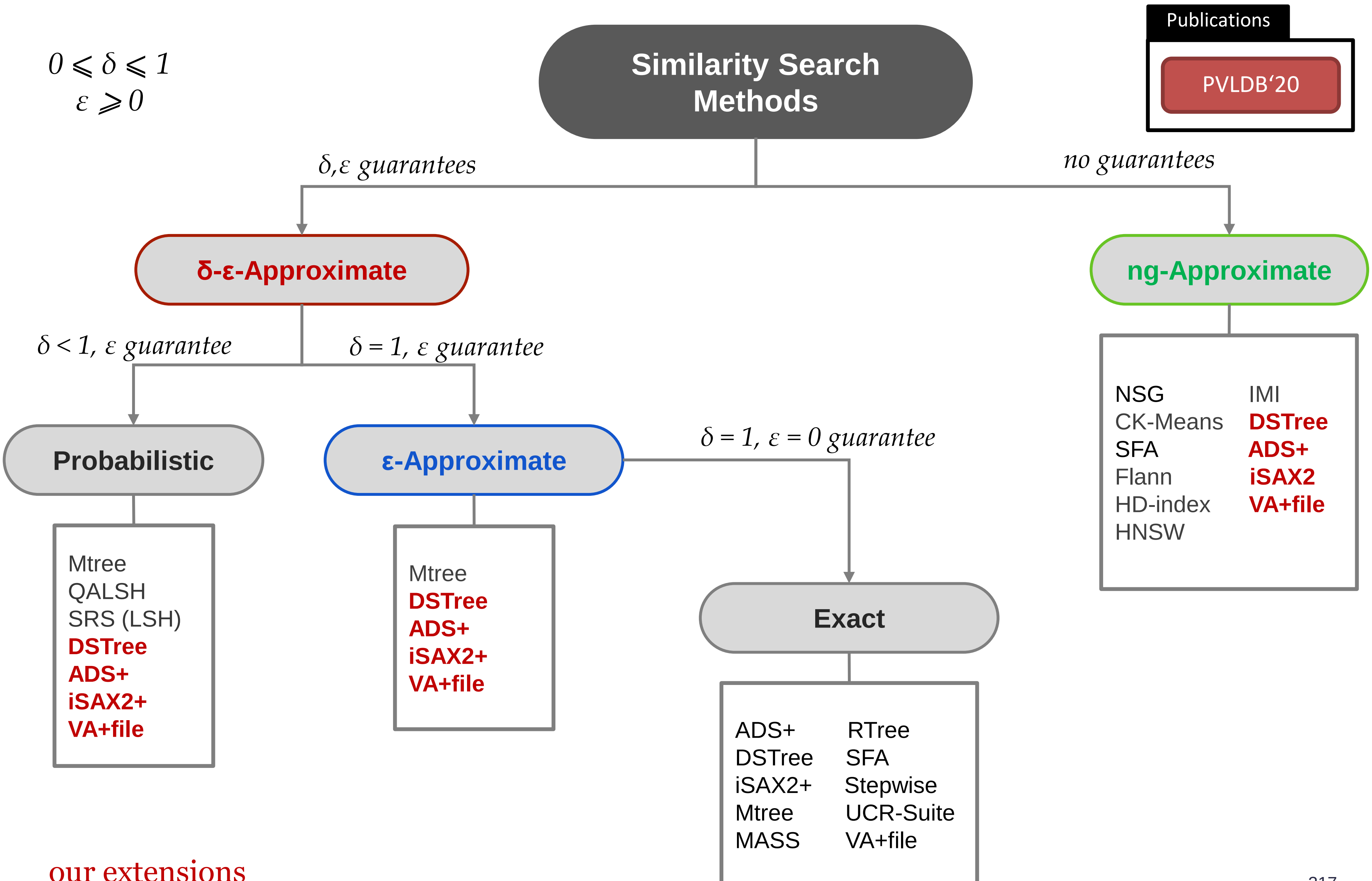
- two sides of the same(?) coin
  - data series as multidimensional points
  - for a specific ordering of the dimensions
- everything we discussed applicable to high-d vectors, too!
- several techniques for similarity search in high-d vectors
  - using LSH (SRS), space quantization (IMI), k-NN graphs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

Publications

PVLDB'20

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





our extensions

Publications

PVLDB'20

Data Series vs. high-d Vectors

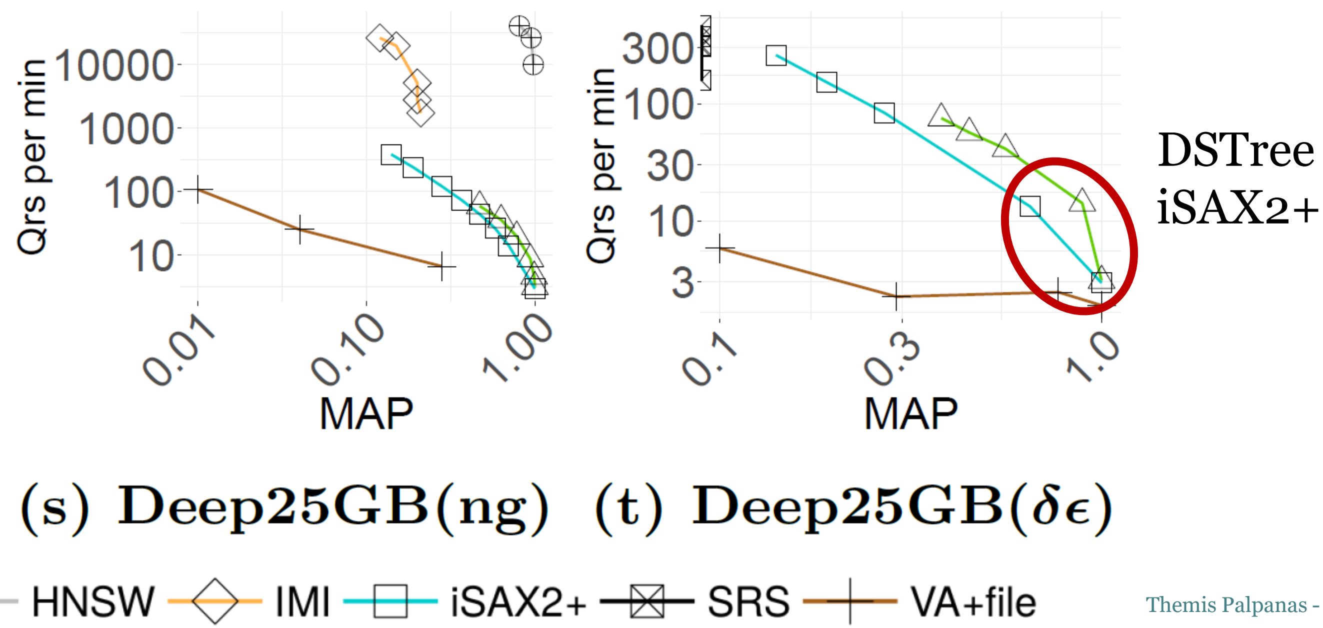
- **data series techniques** are the **overall winners**, even on **general high-d vector** data

Publications

PVLDB‘20

Data Series vs. high-d Vectors

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



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

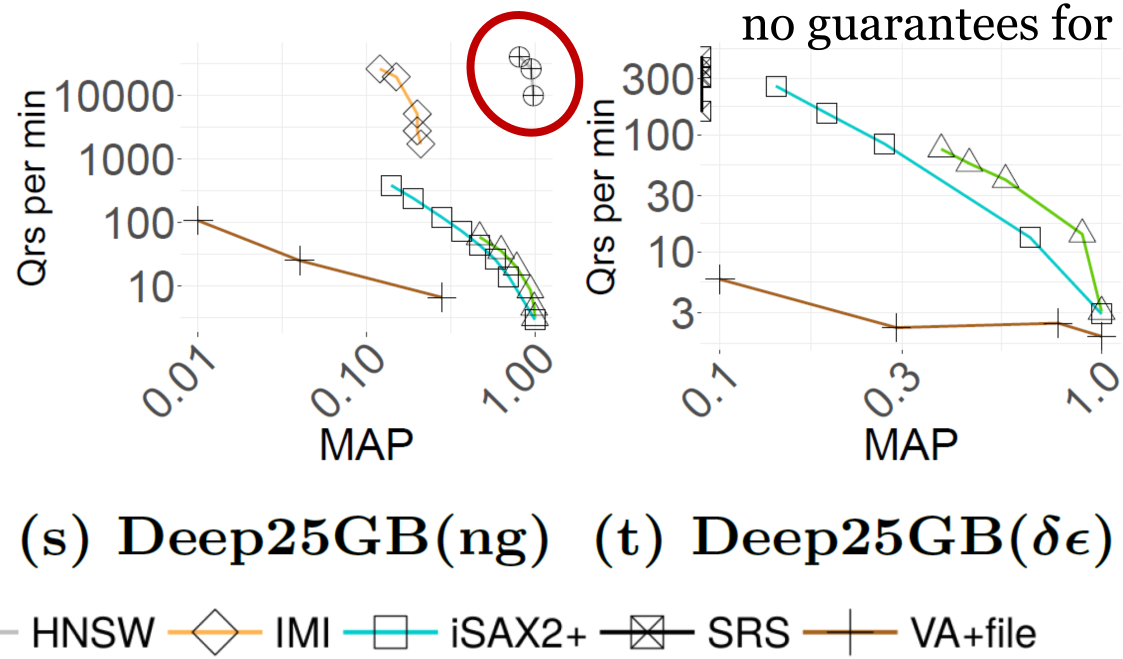
Data Series vs. high-d Vectors

Publications

PVLDB‘20

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

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

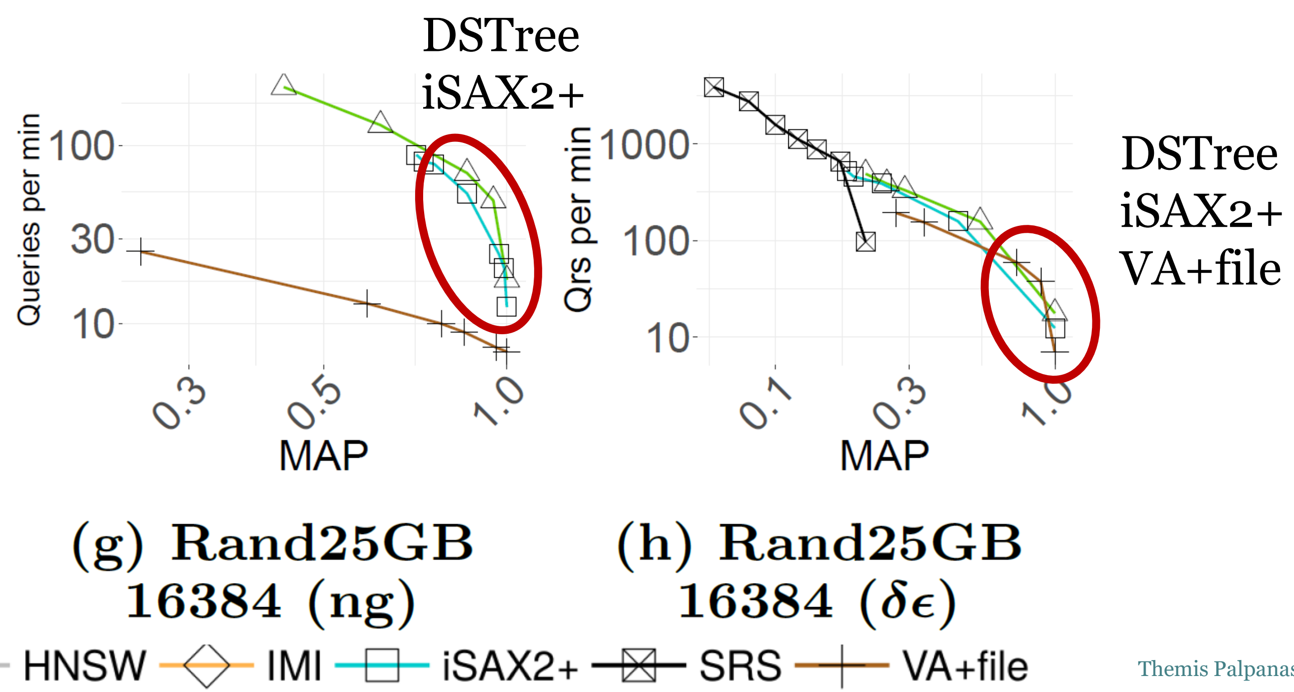


Publications

PVLDB'20

Data Series vs. high-d Vectors

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



(g) Rand25GB 16384 (ng)

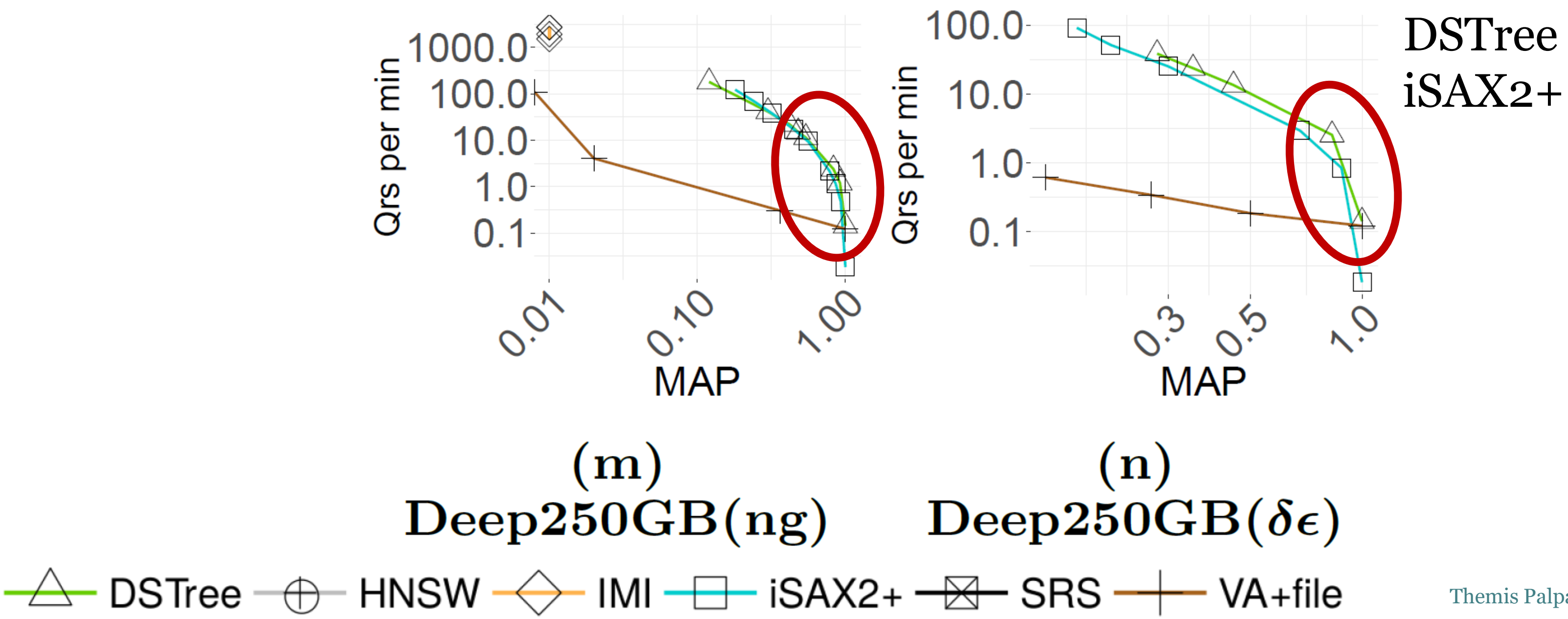
(h) Rand25GB 16384 ($\delta\epsilon$)

Publications

PVLDB'20

Data Series vs. high-d Vectors

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

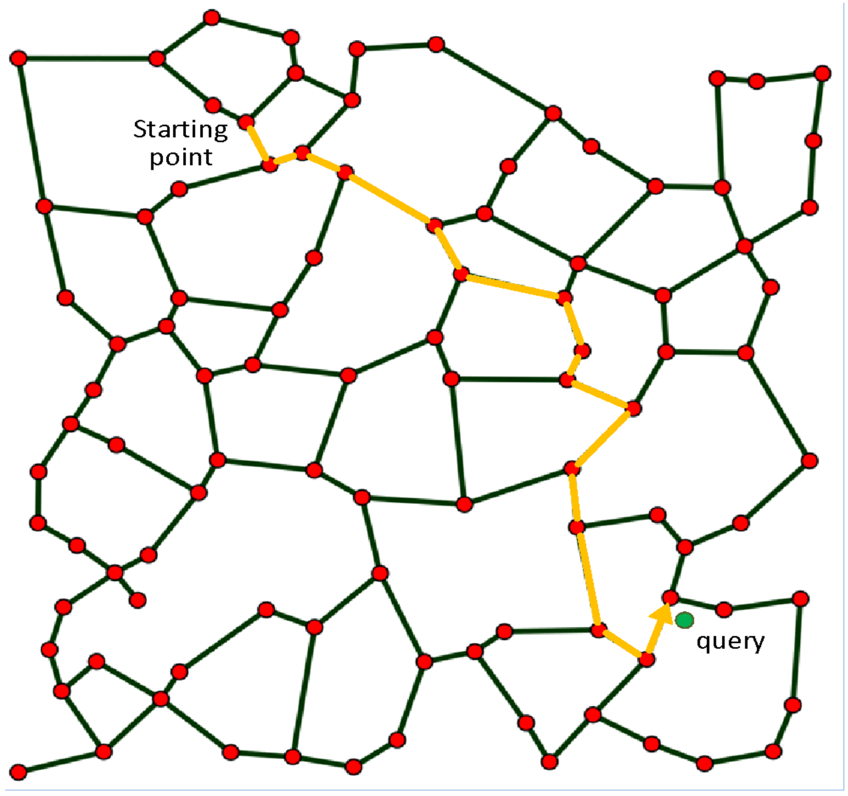


Publications

Kleinberg
STOC' 00

NSW Graphs

- Augment approximate kNN graphs with long range links:
  - Milgram experiment
  - Shorten the greedy algorithm path to log(N)

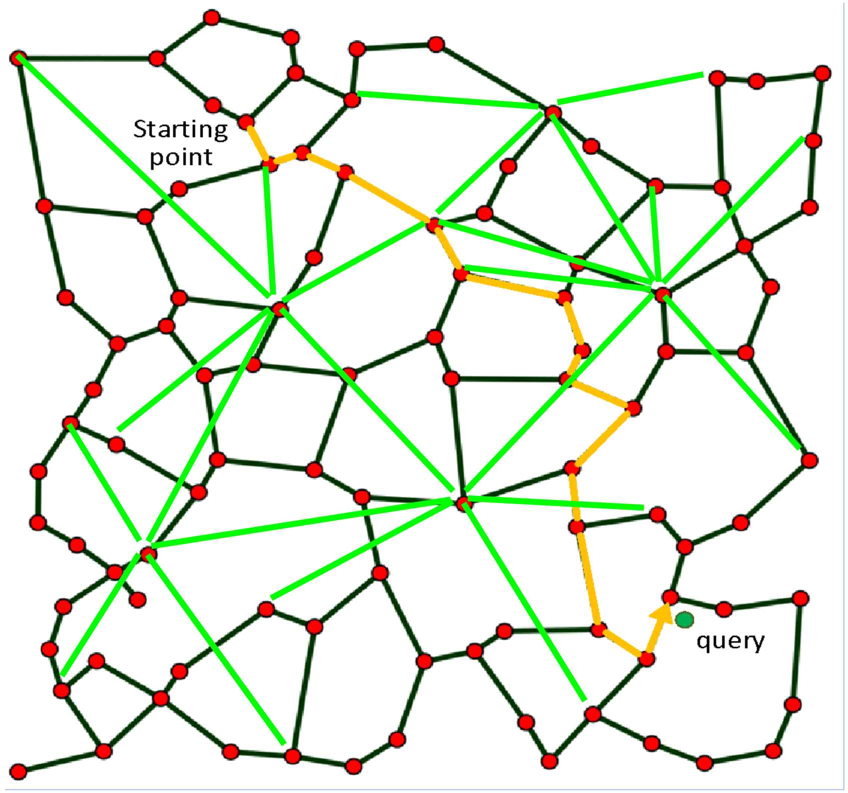


NSW Graphs

Publications

Kleinberg STOC' 00

- Augment approximate kNN graphs with long range links:
  - Milgram experiment
  - Shorten the greedy algorithm path to log(N)

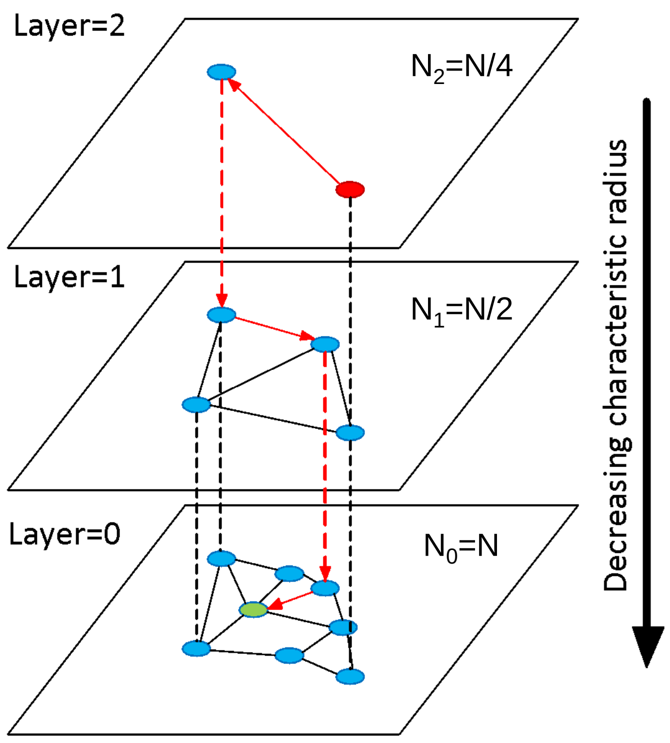


Publications

Malkov et al.
TPAMI' 20
Arxiv'16

HNSW

- In HNSW we split the graph into layers (fewer elements at higher levels)
- Search starts for the top layer. Greedy routing at each level and descend to the next layer.
- Maximum degree is capped while paths ~ log(N) → log(N) complexity scaling.
- Incremental construction
- ng-approximate search



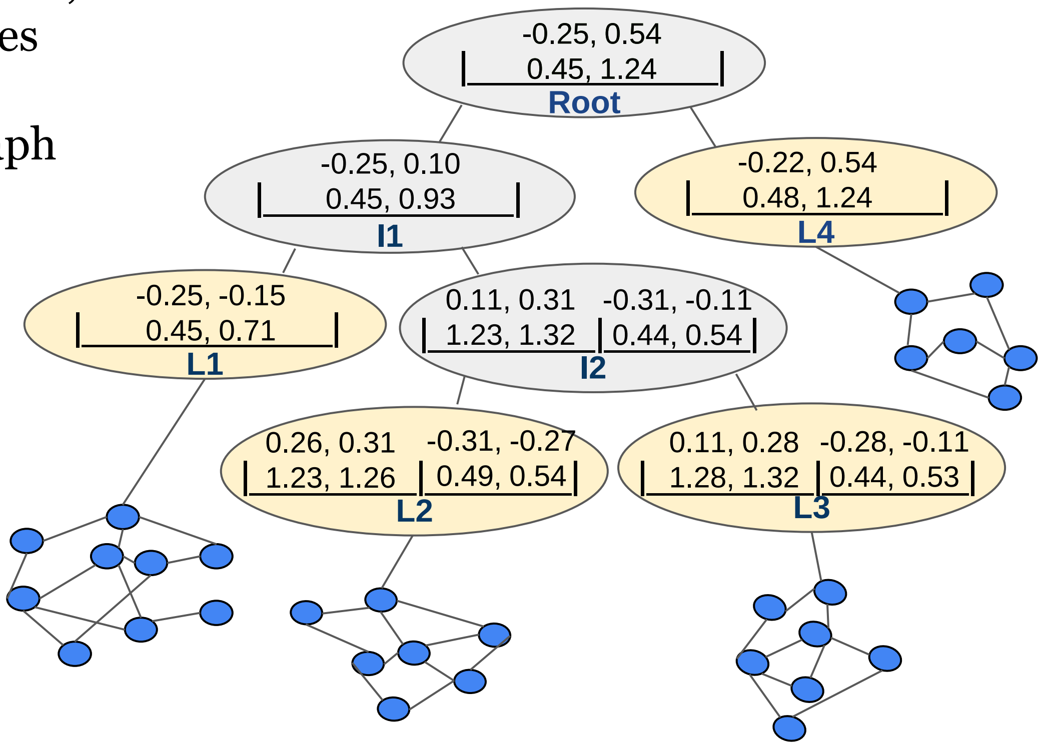
Slides by Malkov

ELPIS
Parallel, In-Memory Indexing of Sequences

Publications

Azizi-PVLDB'23

- In-memory solution for SIMD, multi-core, multi-socket architectures
- **ELPIS** combines tree and graph structures for efficient in-memory ng-approximate vector similarity search.

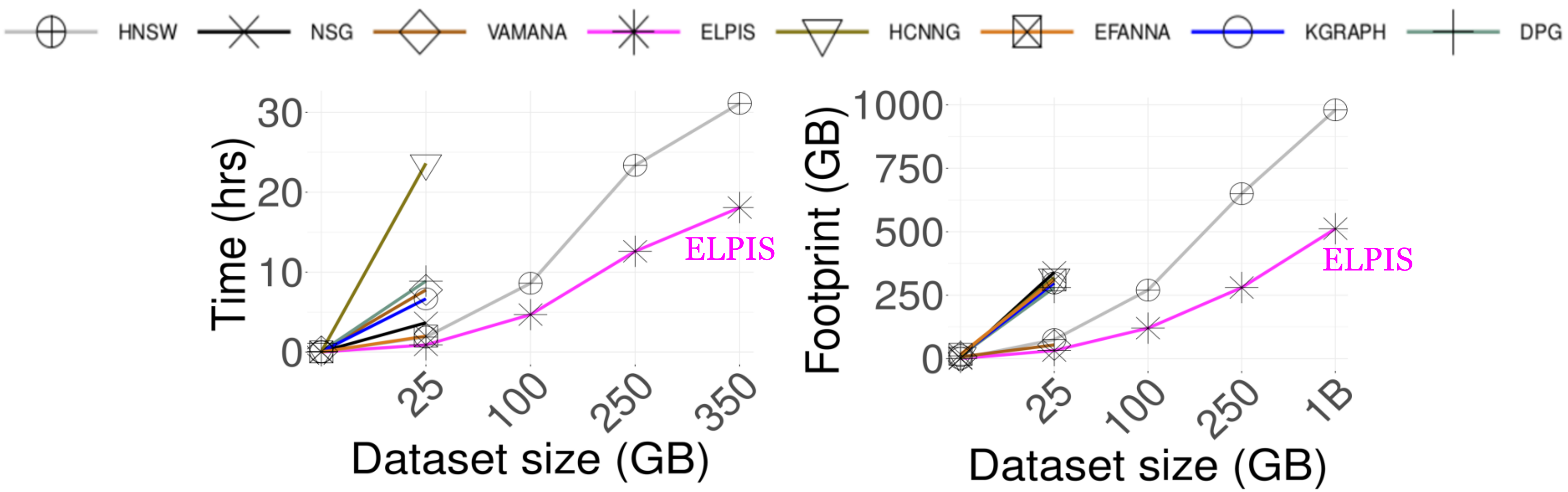


Publications

Azizi-PVLDB'23

ELPIS
Parallel, In-Memory Indexing of Sequences

- Scalability of indexing time and memory footprint with dataset size (Deep)



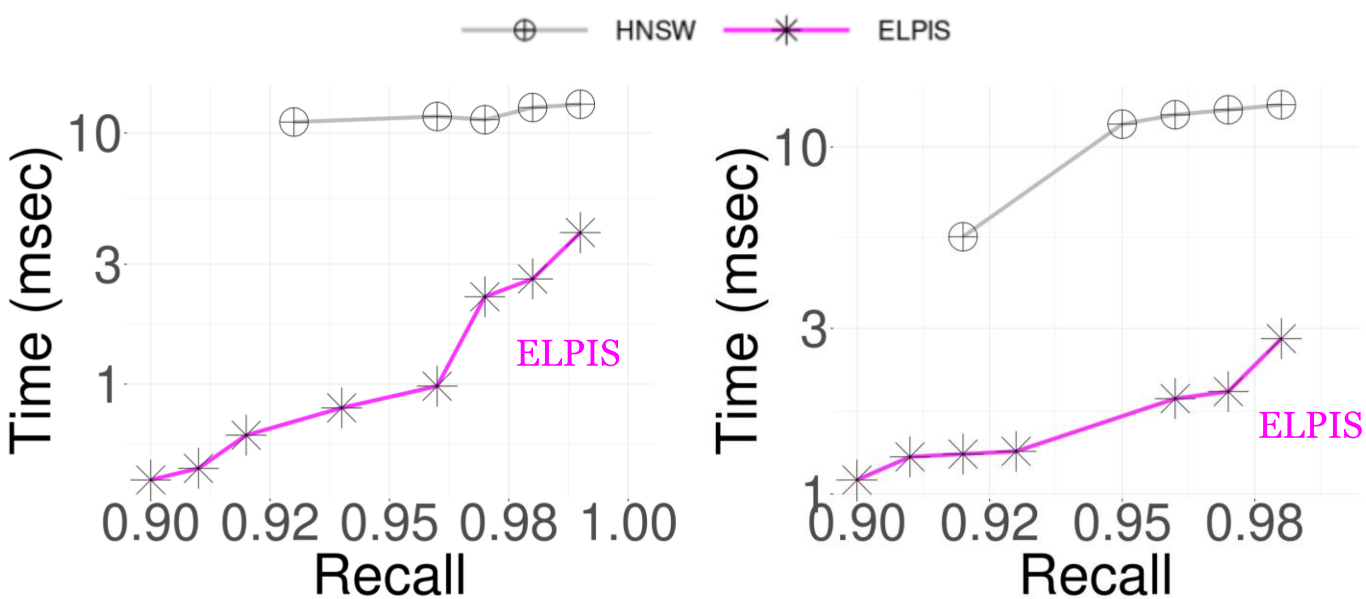
ELPIS builds the index up to **8x faster,**
using **40% less memory**

Publications

Azizi-PVLDB'23

ELPIS
Parallel, In-Memory Indexing of Sequences

- Query Performance on 1B vectors datasets (Sift, Deep)



Sift Dataset

Deep Dataset

ELPIS answers **10-NN queries** in **~3 msec** for a dataset of **1 billion vectors** with **recall 0.99**

Available Solutions

- libraries
  - ELPIS
  - HNSW
  - FAISS (META/Facebook)
- vector databases
  - Pinecone
  - milvus
- general databases
  - Postgres with HNSW and IVF algorithms (open souree)
  - AlloyDB with SCaNN algorithm (Google)
  - Oracle

Conclusions

- high-d vectors is a very common data type
  - across several different domains and applications

Conclusions

- high-d vectors is a very common data type
  - across several different domains and applications
- complex high-d vector analytics are challenging
  - have very high complexity

Conclusions

- high-d vectors is a very common data type
  - across several different domains and applications
- complex high-d vector analytics are challenging
  - have very high complexity
- data series management/indexing techniques provide much needed scalability
  - work for data series and general high-d vectors (and embeddings)
  - lead to fast complex analytics and machine learning

Conclusions

- high-d vectors is a very common data type
  - across several different domains and applications
- complex high-d vector analytics are challenging
  - have very high complexity

- data series management/indexing techniques provide much needed scalability
  - work for data series and general high-d vectors (and embeddings)
  - lead to fast complex analytics and machine learning
- several exciting research opportunities

thank you!

google: **Themis Palpanas**

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

References (chronological order)

- Delaunay, Boris (1934). "Sur la sphère vide". Bulletin de l'Académie des Sciences de l'URSS, Classe des Sciences Mathématiques et Naturelles. **6**: 793–800.
- 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.
- Jon Louis Bentley. 1975. Multidimensional binary search trees used for associative searching. Commun. ACM 18, 9 (Sept. 1975), 509–517.
- Pavlidis, T. (1976). Waveform segmentation through functional approximation. *IEEE Transactions on Computers*.
- Godfried T. Toussaint, The relative neighbourhood graph of a finite planar set, Pattern Recognition, Volume 12, Issue 4, 1980, Pages 261-268,
- 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.
- K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "'nearest neighbor'" meaningful? In *ICDT*, 1999.
- 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.
- J. Kleinberg. The Small-world Phenomenon: An Algorithmic Perspective. In Proceedings of the Thirty- second Annual ACM Symposium on Theory of Computing, STOC '00, pages 163–170, New York, NY, USA, 2000. ACM

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.
- C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In *ICDT*, 2001
- 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)
- Sivic and Zisserman, "Video Google: a text retrieval approach to object matching in videos," *Proceedings Ninth IEEE International Conference on Computer Vision*, Nice, France, 2003, pp. 1470-1477 vol.2
- 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)
- Stephen Blott and Roger Weber. 2008. What's wrong with high-dimensional similarity search? Proc. VLDB Endow. 1, 1 (August 2008), 3.

References (chronological order)

- C. Silpa-Anan and R. Hartley, "Optimised KD-trees for fast image descriptor matching," 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 2008, pp. 1-8
- Alexandr Andoni and Piotr Indyk. 2008. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. Commun. ACM 51, 1 (January 2008), 117–122.
- 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)
- Hervé Jégou, Matthijs Douze, Cordelia Schmid: Product Quantization for Nearest Neighbor Search. IEEE Trans. Pattern Anal. Mach. Intell. 33(1): 117-128 (2011)
- Wei Dong, Charikar Moses, and Kai Li. 2011. Efficient k-nearest neighbor graph construction for generic similarity measures. In Proceedings of the 20th international conference on World wide web (WWW '11).
- P. Schafer and M. Hogvist. Sfa: A symbolic fourier approximation and index for similarity search in high dimensional datasets. ICDE 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.
- J. He, S. Kumar, and S.-F. Chang. On the difficulty of nearest neighbor search. In *ICML*, 2012.

References (chronological order)

- Junhao Gan, Jianlin Feng, Qiong Fang, and Wilfred Ng. 2012. Locality-sensitive hashing scheme based on dynamic collision counting. In SIGMOD.
- Babenko, Artem & Lempitsky, Victor. (2012). The Inverted Multi-Index. IEEE Transactions on Pattern Analysis and Machine Intelligence. 37. 3069-3076.
- 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.
- NSW IS'14: Y. Malkov, A. Ponomarenko, A. Logvinov, and V. Krylov: Approximate nearest neighbor algorithm based on navigable small world graphs, Inf. Syst., vol. 45, pp. 61–68, 2014.
- T. Ge, K. He, Q. Ke, and J. Sun. Optimized Product Quantization. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI), 36(4):744–755, Apr. 2014
- A. Babenko and V. Lempitsky. The Inverted MultiIndex. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 37(6):1247–1260, June 2015.

References (chronological order)

- 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
- David C. Anastasiu and George Karypis. 2015. L2Knng: Fast Exact K-Nearest Neighbor Graph Construction with L2-Norm Pruning. In CIKM '15.
- 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. ICDE 2017: 502-505
- Katsiaryna Mirylenka, Michele Dallachiesa, Themis Palpanas: Data Series Similarity Using Correlation-Aware Measures. SSDBM 2017: 11:1-11:12
- Kostas Zoumpatianos, Themis Palpanas: Data Series Management: Fulfilling the Need for Big Sequence Analytics. ICDE 2018: 1677-1678
- A. Arora, S. Sinha, P. Kumar, and A. Bhattacharya. HD-index: Pushing the Scalability-accuracy Boundary for Approximate kNN Search in High-dimensional Spaces. PVLDB, 11(8):906–919, 2018

References (chronological order)

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

References (chronological order)

- Oleksandra Levchenko, Boyan Kolev, Djamel Edine Yagoubi, Dennis E. Shasha, Themis Palpanas, Patrick Valduriez, Reza Akbarinia, Florent Masseglia: Distributed Algorithms to Find Similar Time Series. ECML/PKDD (3) 2019: 781-785
- Haridimos Kondylakis, Niv Dayan, Kostas Zoumpatianos, Themis Palpanas: Coconut: sortable summarizations for scalable indexes over static and streaming data series. VLDB J. 28(6): 847-869 (2019)
- Danila Piatov, Sven Helmer, Anton Dignös, Johann Gamper: Interactive and space-efficient multi-dimensional time series subsequence matching. Inf. Syst. 82: 121-135 (2019)
- Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, Houda Benbrahim: Return of the Lernaean Hydra: Experimental Evaluation of Data Series Approximate Similarity Search. Proc. VLDB Endow. 13(3): 403-420 (2019)
- C. Fu, C. Xiang, C. Wang, and D. Cai. Fast Approximate Nearest Neighbor Search with the Navigating Spreading-out Graph. PVLDB, 12(5):461–474, 2019.
- Anna Gogolou, Theophanis Tsandilas, Themis Palpanas, Anastasia Bezerianos: Comparing Similarity Perception in Time Series Visualizations. IEEE Trans. Vis. Comput. Graph. 25(1): 523-533 (2019)
- John Paparrizos, Michael J. Franklin: GRAIL: Efficient Time-Series Representation Learning. Proc. VLDB Endow. 12(11): 1762-1777 (2019)
- Cong Fu, Chao Xiang, Changxu Wang, and Deng Cai. 2019. Fast approximate nearest neighbor search with the navigating spreading-out graph. Proc. VLDB Endow. 12, 5 (January 2019), 461–474.
- 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
- Anna Gogolou, Theophanis Tsandilas, Karima Echihabi, Anastasia Bezerianos, Themis Palpanas: Data Series Progressive Similarity Search with Probabilistic Quality Guarantees. SIGMOD Conference 2020: 1857-1873
- Themis Palpanas. Evolution of a Data Series Index - The iSAX Family of Data Series Indexes. CCIS, 1197 (2020)
- Djamel Edine Yagoubi, Reza Akbarinia, Florent Masseglia, Themis Palpanas: Massively Distributed Time Series Indexing and Querying. IEEE Trans. Knowl. Data Eng. 32(1): 108-120 (2020)

References (chronological order)

- Botao Peng, Panagiota Fatourou, Themis Palpanas: MESSI: In-Memory Data Series Indexing. ICDE 2020: 337-348
- Kefeng Feng, Peng Wang, Jiaye Wu, Wei Wang: L-Match: A Lightweight and Effective Subsequence Matching Approach. IEEE Access 8: 71572-71583 (2020)
- Chen Wang, Xiangdong Huang, Jialin Qiao, Tian Jiang, Lei Rui, Jinrui Zhang, Rong Kang, Julian Feinauer, Kevin Mcgrail, Peng Wang, Diaohan Luo, Jun Yuan, Jianmin Wang, Jiaguang Sun: Apache IoTDB: Time-series database for Internet of Things. Proc. VLDB Endow. 13(12): 2901-2904 (2020)
- 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
- Kejing Lu, Hongya Wang, Wei Wang, Mineichi Kudo. VHP: Approximate Nearest Neighbor Search via Virtual Hypersphere Partitioning. PVLDB, 13(9): 1443-1455, 2020
- Yury A. Malkov, D. A. Yashunin: Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. IEEE Trans. Pattern Anal. Mach. Intell. 42(4): 824-836 (2020)
- Botao Peng, Panagiota Fatourou, Themis Palpanas. Paris+: Data series indexing on multi-core architectures. TKDE, 2021
- Botao Peng, Panagiota Fatourou, Themis Palpanas. SING: Sequence Indexing Using GPUs. ICDE, 2021
- Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas. Big Sequence Management: Scaling Up and Out. EDBT 2021
- Botao Peng, Panagiota Fatourou, Themis Palpanas. Fast Data Series Indexing for In-Memory Data. VLDBJ 2021