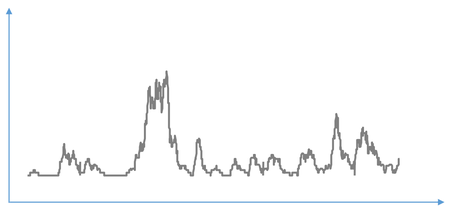
T-Store: Tunable Storage for Large Sequential Data

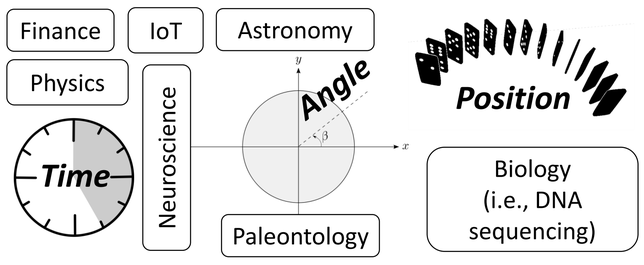
Kostas Zoumpatianos[1,2], Stratos Idreos[1] Themis Palpanas[2]

[1] Harvard University [2] Paris Descartes University

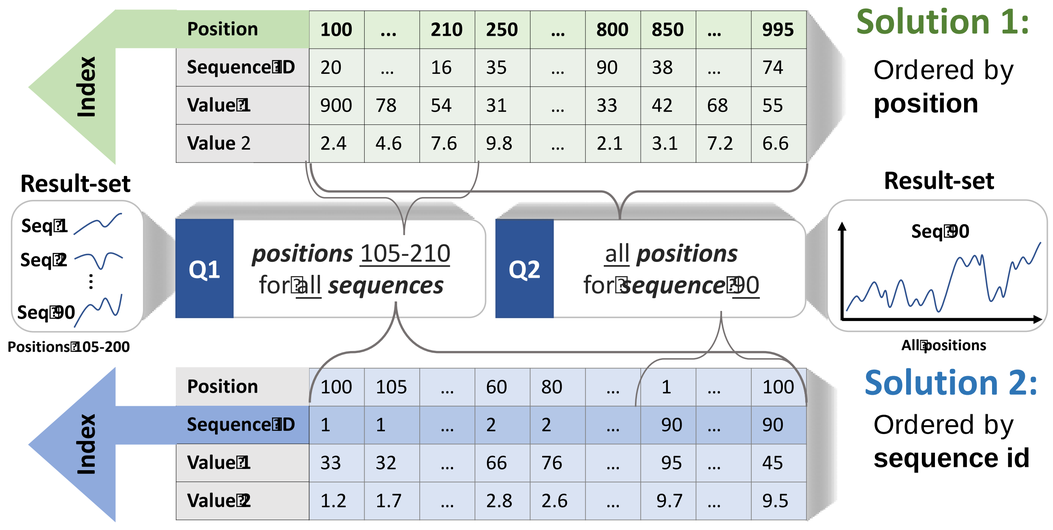
Data Series

Sequences of points **ordered** over a **dimension**

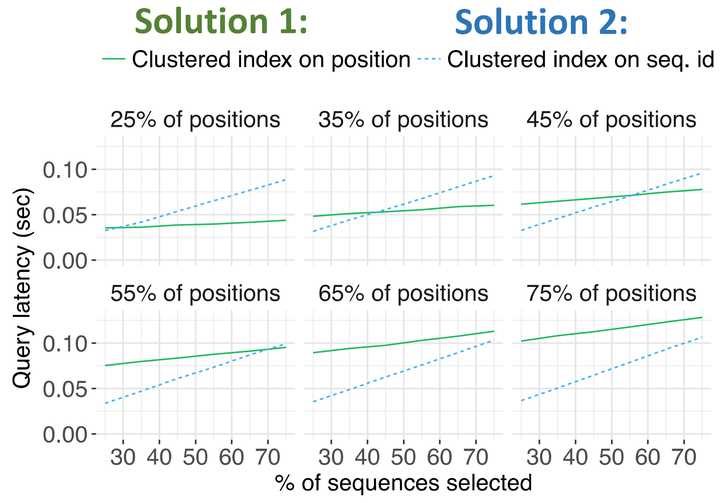




Data Layout



DBMS Performance



There is **no perfect** storage scheme

Design Continuum: From Fixed to Hybrid Layouts

Sequence first | Groups of series | Position first

How to achieve the best layout for a given **workload** and a given set of **data series**?

Current Systems

Choose a **fixed layout without considering** the **workload**

Moreover, **all series are stored in the same way** since **layouts are applied globally** to all series

Our Solution: Hybrid Layouts with T-Store

Hybrid store: supports three different data layouts simultaneously
- Position-first
- Sequence-first
- Groups-of-series (sorted position-first)

Storage advisor: choses the appropriate layout for each series
- Constructs a set of **features** that **describe a user workload**
- **Partitions by solving** an **optimization** problem using modeling

Smart grouping: Constructs groups-of-series for various applications
- **Data mining** (i.e., similarity search) by grouping **similar shapes**
- **Data ingestion** by grouping series that are **updated together**

Optimization

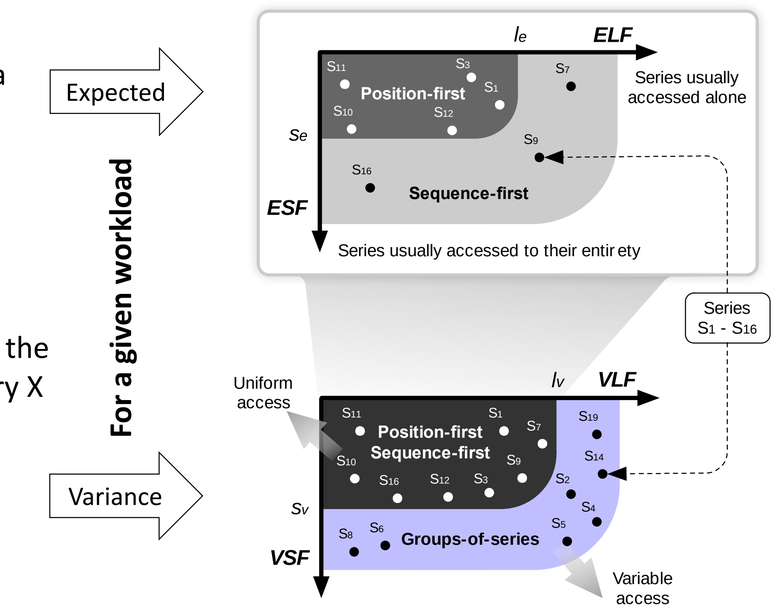
Loneliness Factor. Used to identify series usually queried in isolation. For a query Q that touches a set of series that also includes series X:

$$LF(Q, X) = \frac{\#\text{ series in database}}{\#\text{ series accessed by Q}}$$

Series Selectivity Factor. Captures how much of the total positions of a series are accessed by a query X that touches it. We compute this as follows:

$$SF(Q, X) = \frac{\#\text{ points of series X selected by Q}}{\#\text{ points of series X}}$$

For a given workload



Position-first data layout

$$c_{pfirst}(q_i, D) = \left(log_f \sum_{\forall s_i \in D} |s_i|\right) * c_{lat} * \alpha + \left\lceil sel^D_{pts}(q_i) * \frac{r_{size}}{p_{size}} \right\rceil * c_{thr} * \beta \quad (1)$$

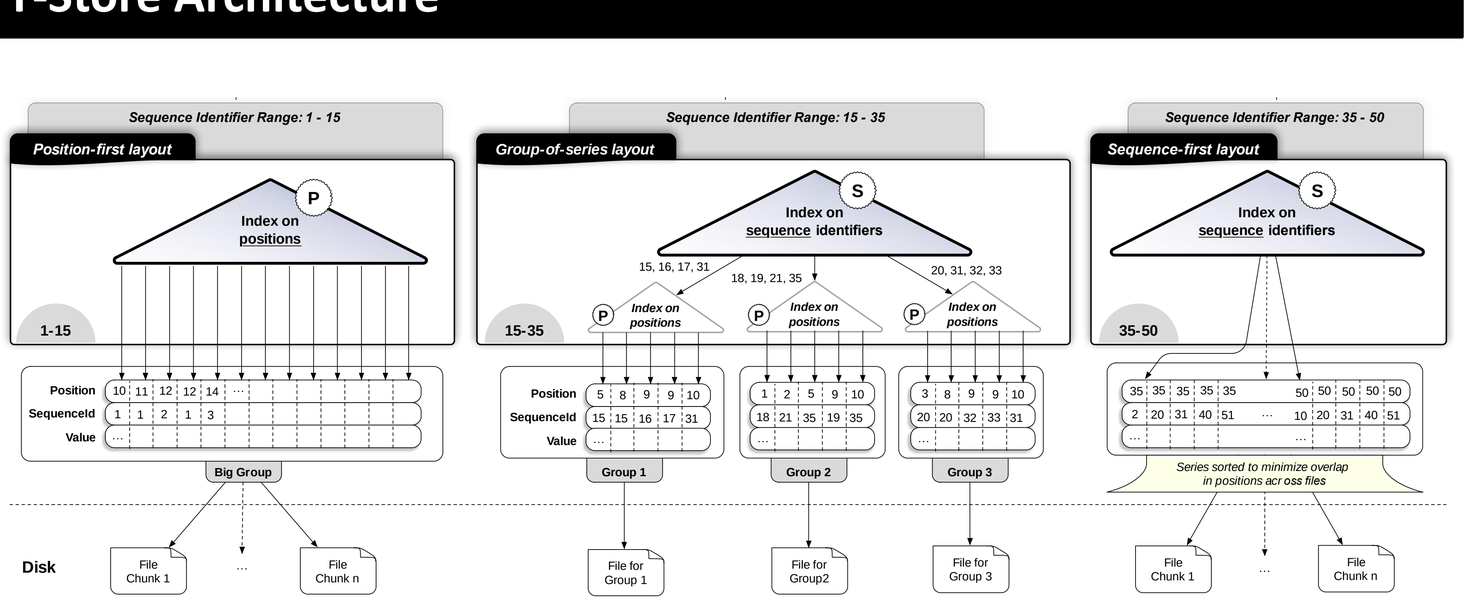
Cost Modelling

Optimization Problem

Sequence-first data layout

$$c_{sfirst}(q_i, D) = sel_{seq}(q_i) * (log_f |D|) * c_{lat} * \alpha + \sum_{\forall s_i \in s q_i} log_2(|s_i|) * c_{lat} + \left\lceil sel_{pts}(q_i) * \frac{r_{size}}{p_{size}} \right\rceil * c_{thr}$$

T-Store Architecture



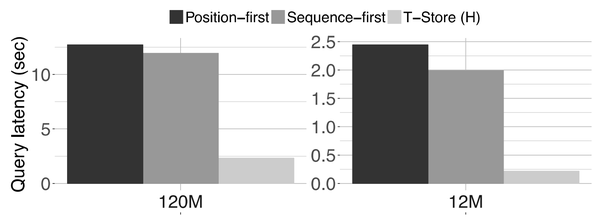
Results

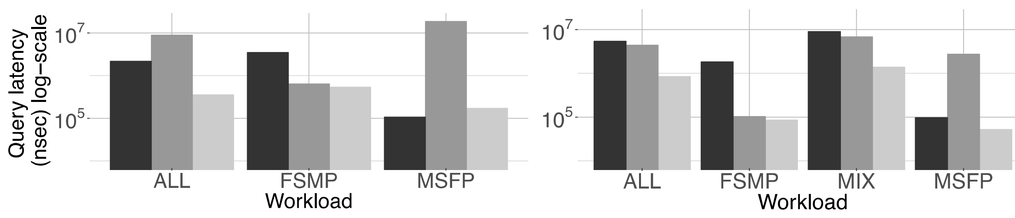
T-Store **outperforms static layouts** by **intelligently partitioning** data series and storing them using the **appropriate data layout**

FSMP: Part of series for which the workload that touches few series and many positions

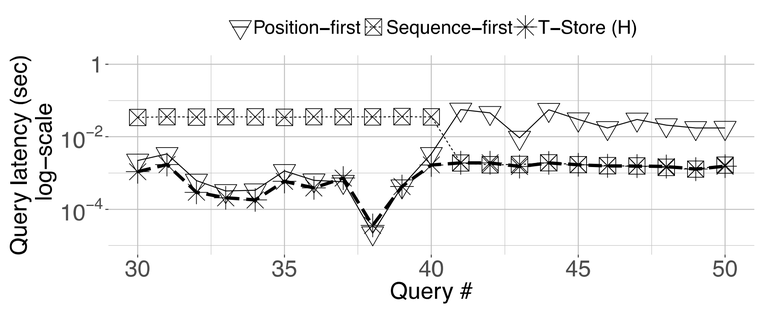
MSFP: Part of the series for which workload that touches many series but few positions

MIX: Part of the series for which both access patterns are exhibited





T-Store identifies the **optimal data layout for each query**



Funded by the European Union's Horizon 2020 research and innovation program under grant agreement NESTOR (Marie Curie #748945).