23rd USENIX Conference on File and Storage Technologies

Oasis: An Out-of-core Approximate Graph System via All-Distances Sketches

[†]***Tsun-Yu Yang***, [‡]Yi Li, [†]Yizou Chen, [‡]Bingzhe Li, and [†]Ming-Chang Yang

[†]The Chinese University of Hong Kong

[‡]The University of Texas at Dallas

Outline

- **Introduction**
- Background
- Oasis System
- Evaluation

Graph and Graph Processing

- Graphs are a powerful data structure that can express a wide range of real-world relationships. They do this by storing entities as vertices and connections between entities as edges.

- There are many important graph applications reply on neighborhood information.
  - Social network analysis
  - Recommendation system
  - Navigation planning

- Solutions:
  - In-memory graph processing ⇒ *efficient but expensive for large-scale graphs*
  - Out-of-core graph processing ⇒ *cheap but slow I/O bandwidth*

Approximate Graph Processing

- In many real-world applications, exact answers are not always necessary.

- All-distances sketch (ADS) has recently emerged as a promising scheme to capture neighborhood information.
  - ADS is a probabilistic data structure defined for each vertex. It is a “sketch” to summarize how a vertex u is connected to other vertices in a graph.
  - More precisely, an ADS of a vertex u contains the distances of u connected to other “landmark” vertices.

- According to an existing study [1], ADS is the only sketching scheme that combines the following three characteristics:
  - Multi-Functionality ⇒ ADS can be deployed for various applications
  - Controllable and Guaranteed Accuracy ⇒ Control the error bounds of approximation
  - Scalability ⇒ space and time complexity grow near-linearly with the graph scale

[1] Takuya Akiba and Yosuke Yano. Compact and scalable graph neighborhood sketching, In Proceedings of the 22nd ACM SIGKDD

Key Challenges of ADSs

Despite the fact that ADS is well-developed in theory, there is still a wide gap in its practical use in real-world cases, mostly because of its excessively high memory consumption.

Key Challenges:

1. Recent efforts in ADS mainly focus on theoretical aspects and propose algorithms with all-in-memory environments.
2. Since managing ADSs is more complex, most techniques from out-of-core graph systems is ineffective for ADS scenarios. So, running ADS construction on traditional out-of-core graph systems leads to poor performance.

Due to these challenges, we propose **Oasis**, an out-of-core approximate graph system that brings the ADS technique into practical use by leveraging storage effectively.

Outline

- Introduction
- **Background**
- Oasis System
- Evaluation

All-Distances Sketches – Theory

- Given a graph $G = (V, E)$, ADSs are defined with a integer parameter k and a random rank assignment function r to all vertices.
  - The parameter k decides the trade-off between sketch size and estimation accuracy.
  - r is a rank function, where $r(v) \rightarrow [0, 1]$ for any $v \in V$.

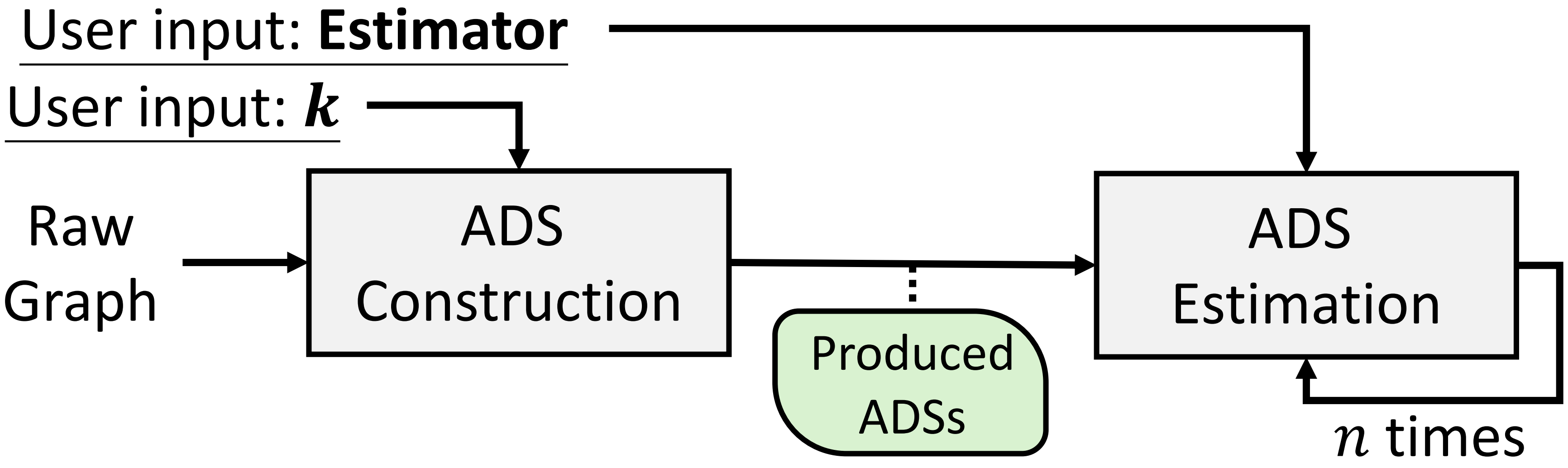
$$N(u, v) = \{x \in V \mid d_{u,x} < d_{u,v}\}$$

$$\pi(u, v) = k_r^{th}\{N(u, v)\}$$

$$ADS(u) = \left\{\left(v, d_{u,v}\right) \middle| \, v \in V, r(v) < \pi(u, v)\right\}$$

Each vertex has its own ADS array of size $O(k\log V)$
➔ The total size of ADSs is $O(Vk\log V)$

All-Distances Sketches – Overview


User input: **Estimator**

User input: $\boldsymbol{k}$

Raw Graph

ADS Construction

Produced ADSs

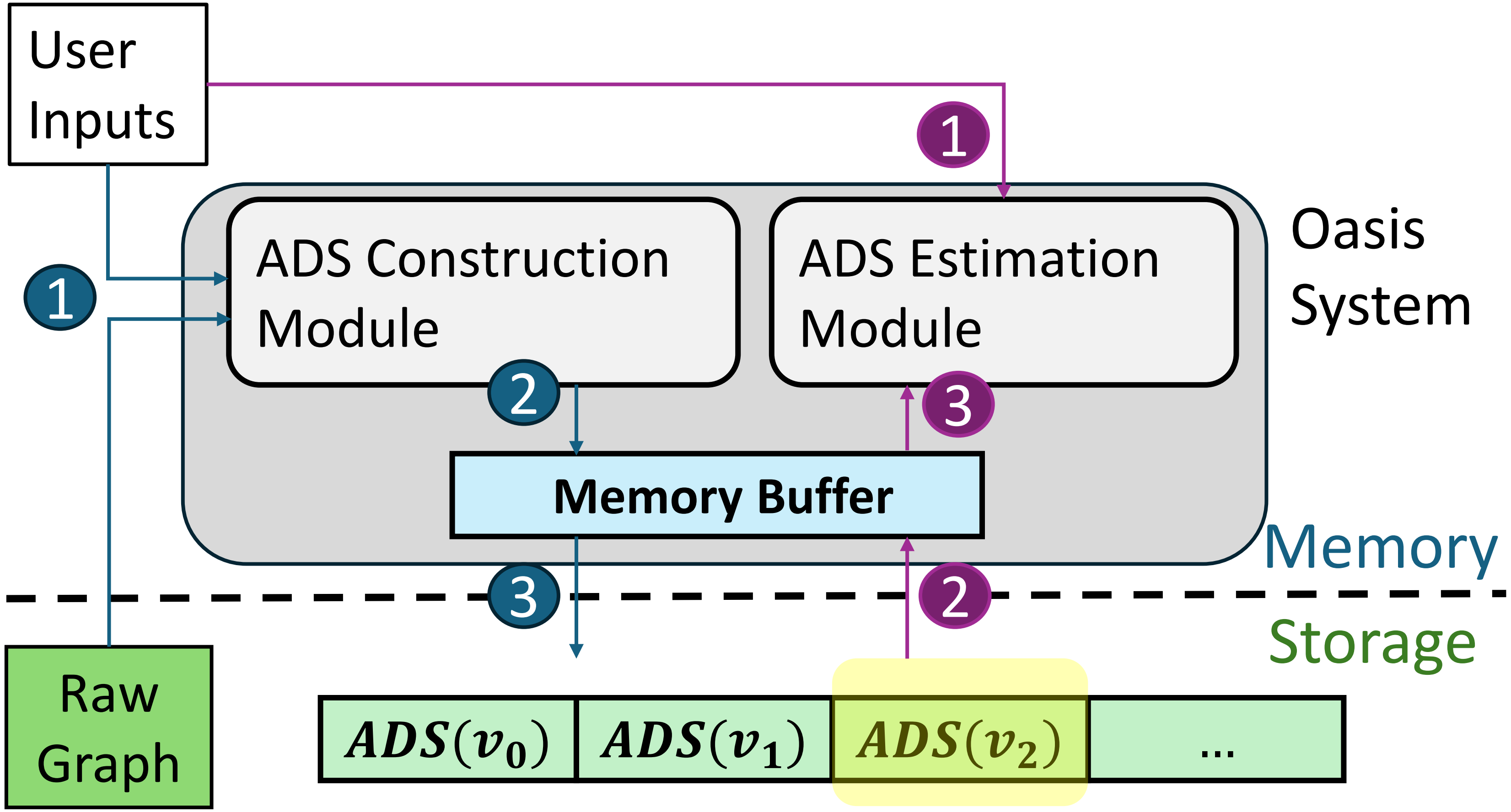
ADS Estimation

n times

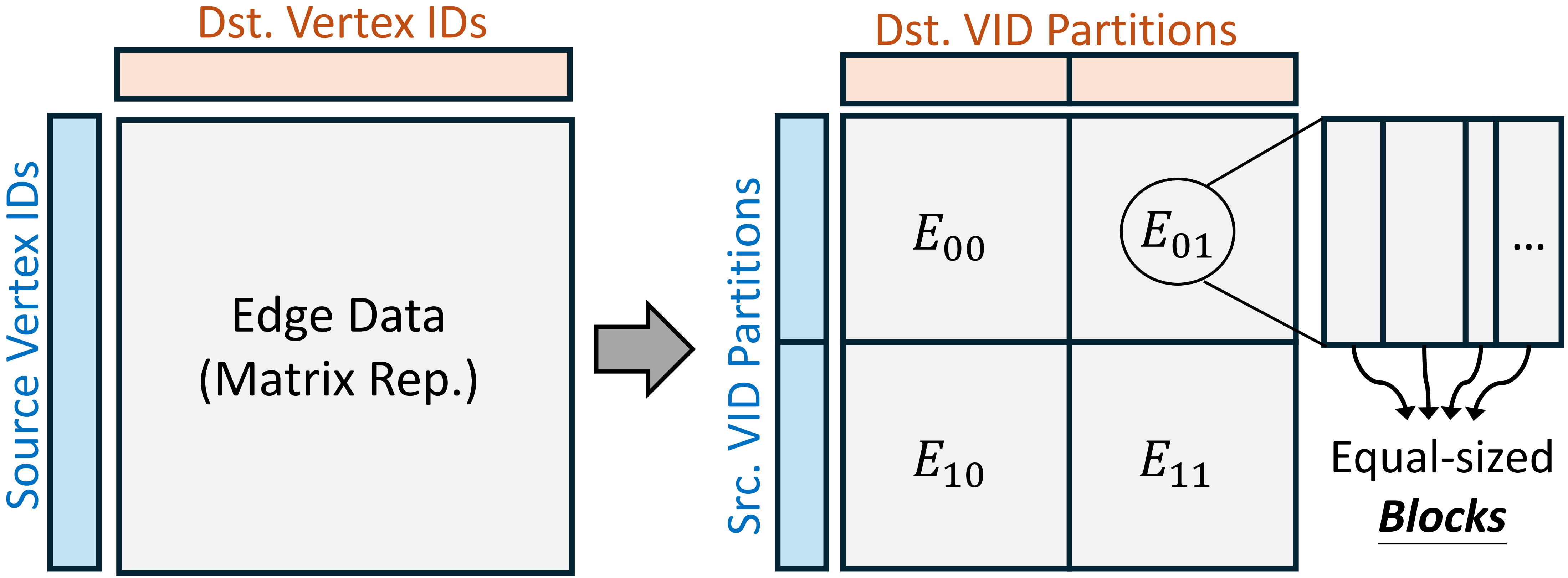
Outline

- Introduction
- Background
- **Oasis System**
- Evaluation

Overview of Oasis



Partition-based Data Layout

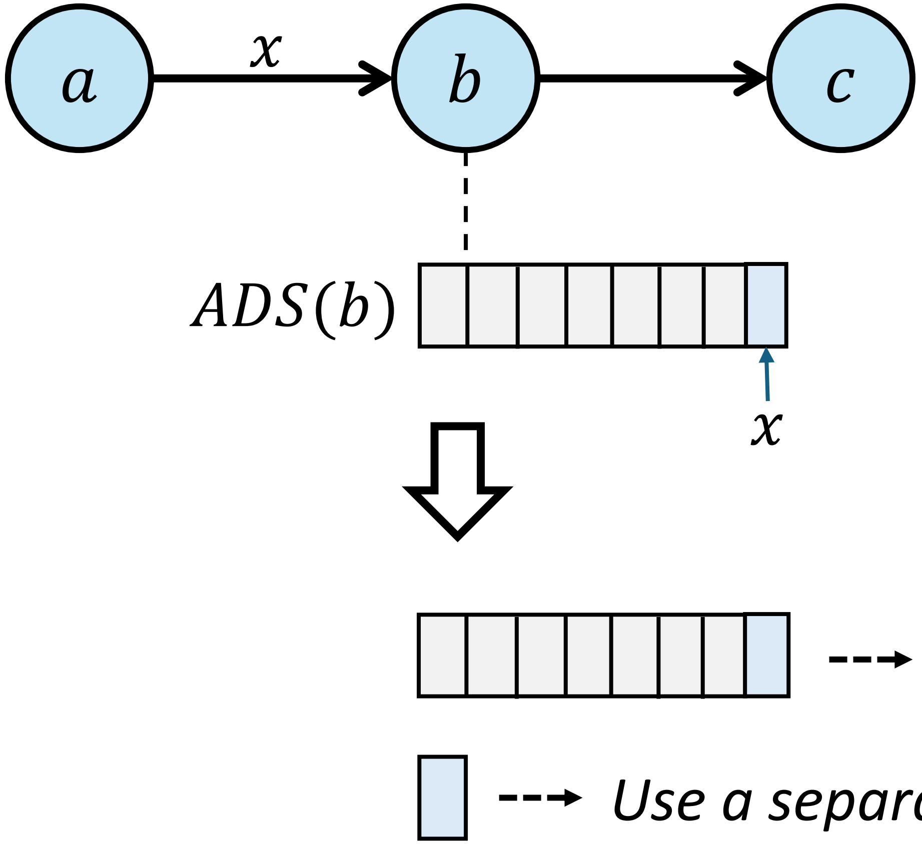


Divide Raw Edge Layout into Different Edge ***Grids*** based on Vertex ID Partitions

Each thread will handle one block

Active Data Separation

- Since ADS is the largest data structure during construction, how to minimize the I/O amount of loading ADSs is crucial.
- Active data separation is a technique aiming to minimize the loading for *active ADSs*.
  - *Active ADSs* refer to the set of ADSs that require processing in the current iteration.

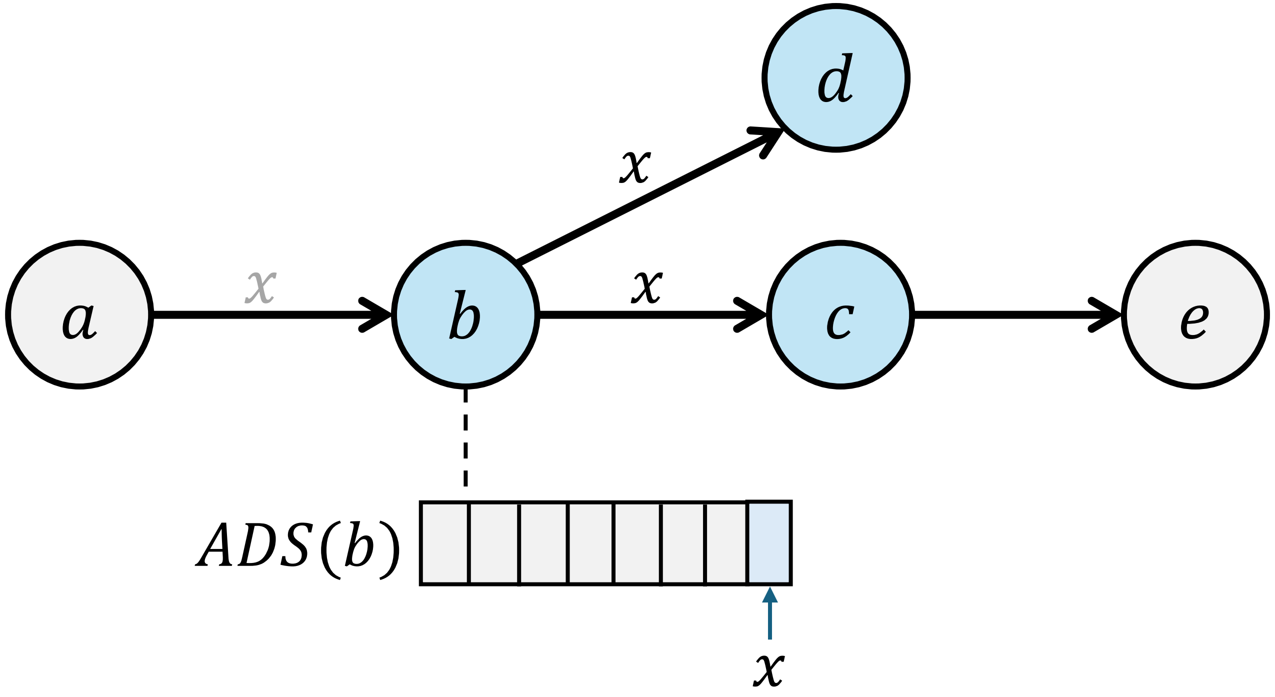


Original ADS file

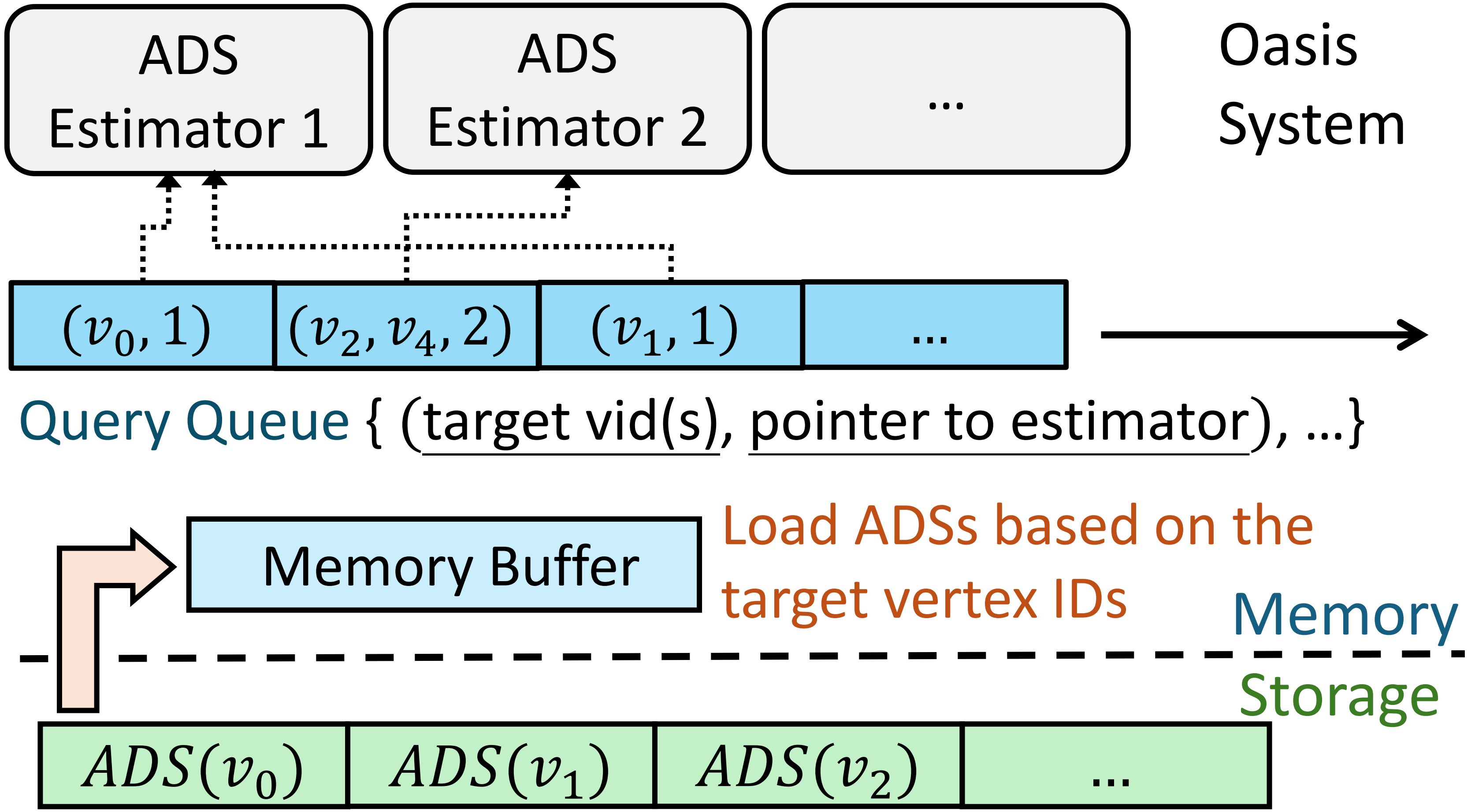
Use a separate file to hold the active ADSs

Selective ADS Accessing

- Selective ADS Accessing is designed to reduce unnecessary ADS reads by loading only the ADSs that actually receive updates.



Framework of Oasis ADS Estimation



1. Locality-aware Query Assignment
2. Grid-based Estimation

Outline

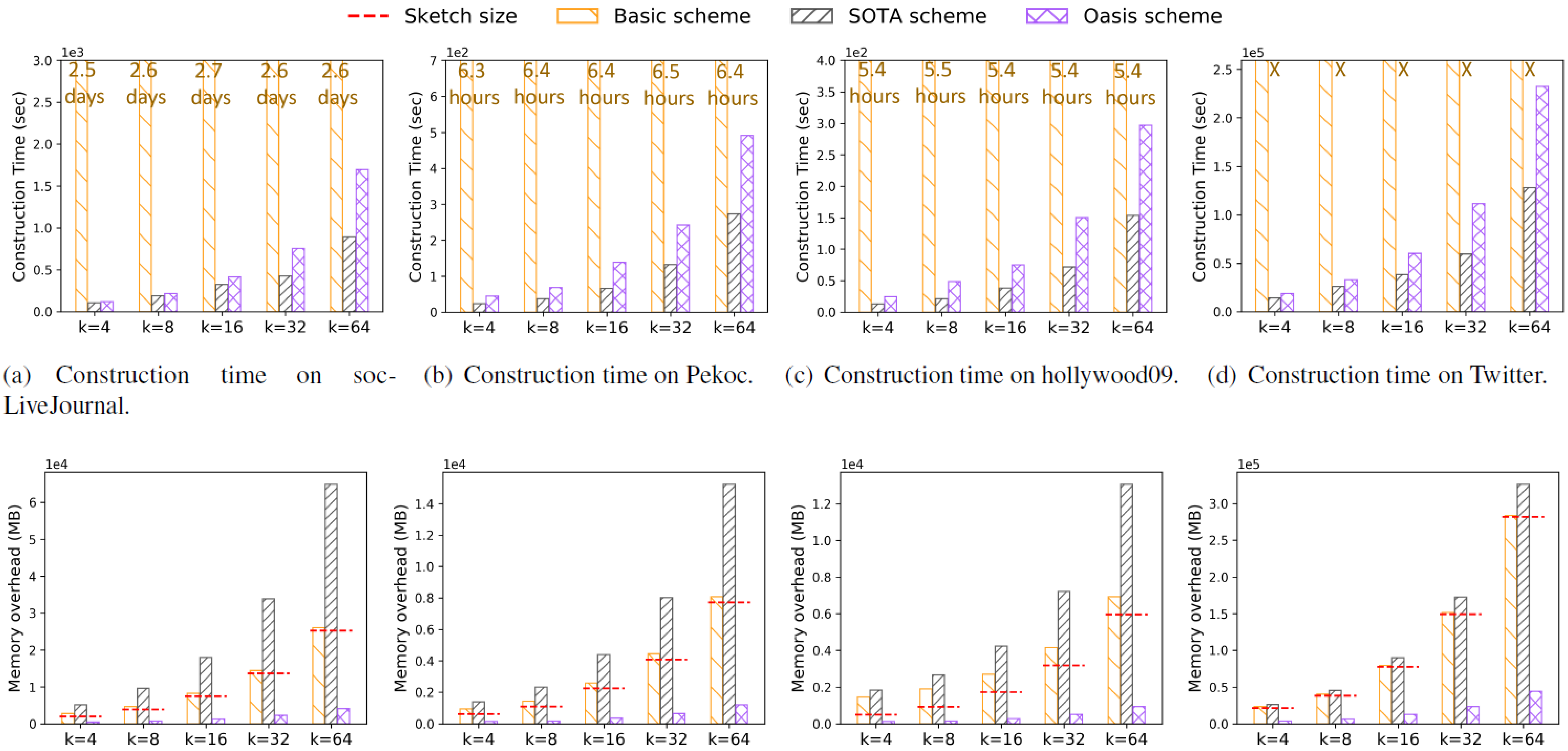
- Introduction
- Background
- Oasis System
- **Evaluation**

Evaluation Setup

- We compare Oasis against two in-memory schemes: *Basic* and *SOTA*
  - Basic is a straightforward implementation of ADS formula.
    - Perform graph traversal from every vertex.
    - The number of edge traversal is $O(VE)$.
  - SOTA is proposed to achieve significantly lower time complexity.
    - Run on transpose graph. Perform bounded graph traversal.
    - The number of edge traversal is $O(E\mathrm{klog}V)$.

- We use 16 partitions by default.

Comparisons of ADS Construction



(a) Construction time on soc-LiveJournal.

(b) Construction time on Pekoc.

(c) Construction time on hollywood09.

(d) Construction time on Twitter.

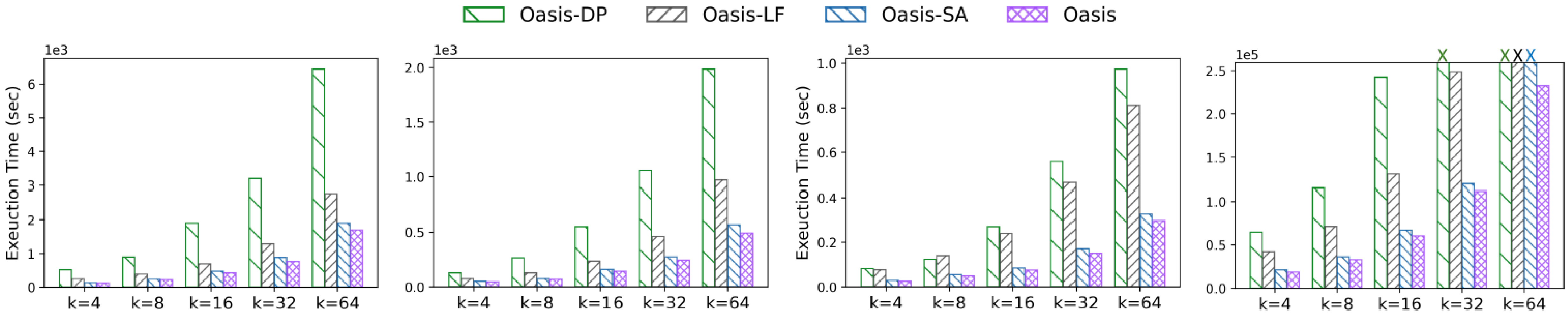
(e) Construction memory on soc-LiveJournal.

(f) Construction memory on Pekoc.

(g) Construction memory on hollywood09.

(h) Construction memory on Twitter.

Design Choices of ADS Construction



(a) Executoin time on soc-LiveJournal.

(b) Executoin time on Pekoc.

(c) Executoin time on hollywood09.

(d) Executoin time on Twitter.

Oasis-DP ⇒ Oasis without active data separation

Oasis-LF ⇒ Oasis without edge block

Oasis-SA ⇒ Oasis without selective ADS accessing

Conclusion

- This work introduces Oasis, which is an out-of-core approximate graph system based on ADSs to manage ADSs with low memory and high efficiency.

- First, this work studies how to construct ADSs with a small memory amount, and proposes various system-level optimizations to decently improve its construction time.

- Next, an ADS estimation framework is presented, allowing users to implement their estimators easily and provides efficient runtime estimation.

Thank you for your attention

Q&A