Automatic Algorithm Transformation for Efficient Multi-Snapshot Analytics on Temporal Graphs
(时态图上对多快照分析进行高效算法自动转换)

蔡修远
数据管理和系统实现

華東師範大學
EAST CHINA NORMAL UNIVERSITY

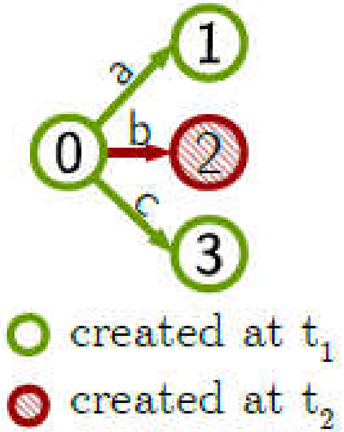
Why Multi-snapshot analytics

- **What is snapshot**

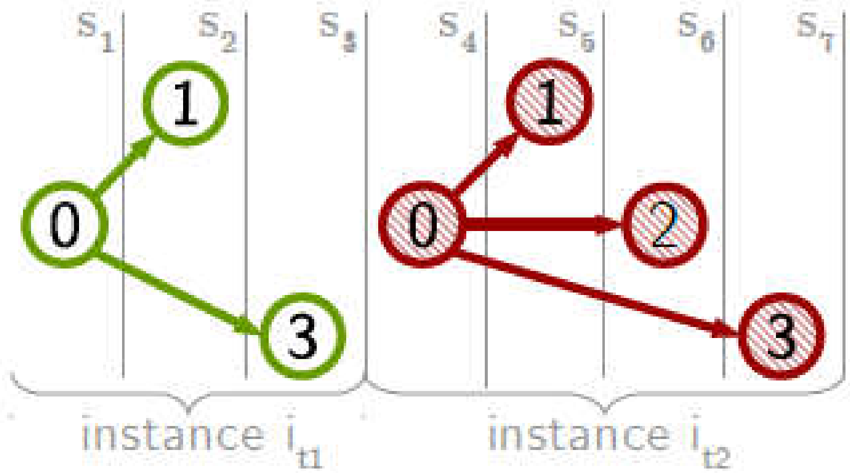
- **What can we do with multi-snapshot**

basic solution

➢ Independent Snapshot Execution



(a) Graph G^S

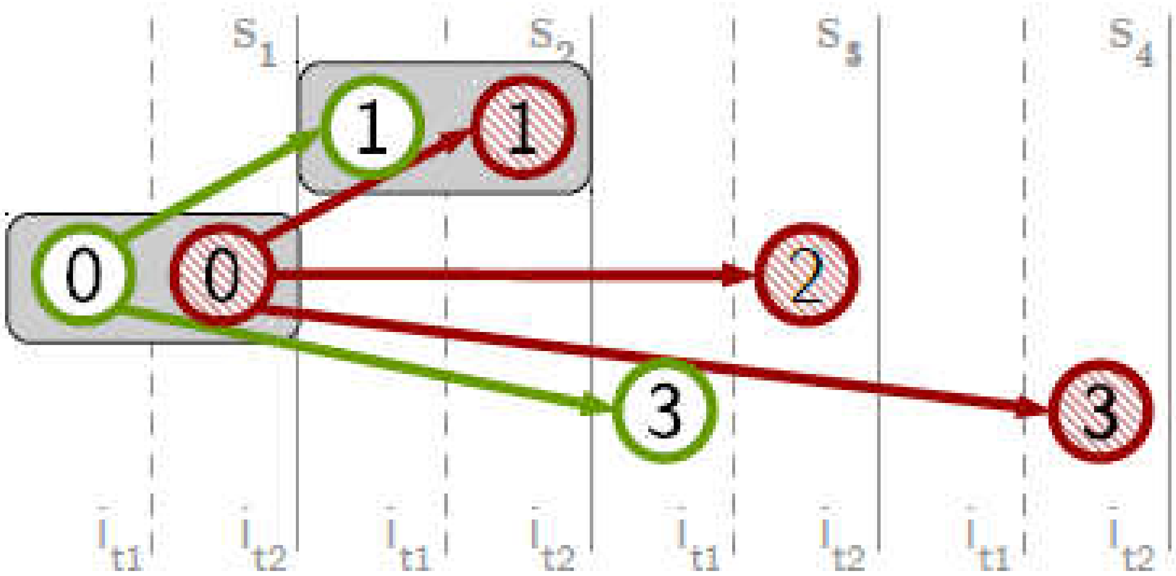


(b) Snapshot-at-a-time

- ➢ How to facilitate the similarity among snapshots

Advanced solution

- **Instance Interleaving**



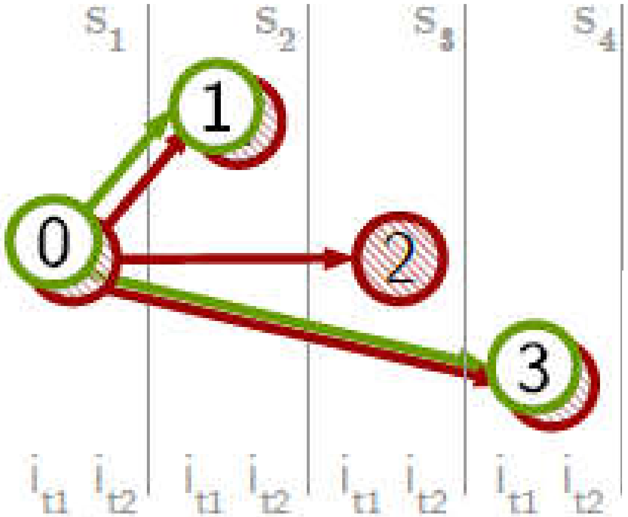
(c) Interleaved

$$i_{t_1} : G^S.neighbors(v_0) = [v_1, v_3]$$

$$i_{t_2} : G^S.neighbors(v_0) = [v_1, v_2, v_3]$$

Solution author propose

➢Graph-Synchronous Interleaved Execution



(d) SAMS

Algorithm 1 1-hop neighborhood traversal.

$visit(v_0)$
for $n \in G^S.neighbors(v_0)$ **do**
  $visit(n)$

Algorithm 2 SAMS variant of Algorithm 1 that allows its concurrent execution for two graph snapshots.

$visit(v_0,\ \{i_{t_1}, i_{t_2}\})$
for $(n, active_n) \in G^S.neighbors_*(v_0,\ \{i_{t_1}, i_{t_2}\})$ **do**
  $visit(n,\ active_n)$

Interleaving of Program Instances

华东师范大学
EAST CHINA NORMAL UNIVERSITY

➢ Rules for algorithm interleaving

Rule 1 Function definition

$$\begin{array}{l}\textbf{sim}\ (\_)\ \textbf{do}\\ \quad \textbf{fun}\ F(args\ldots)\\ \qquad stmt\end{array} \Bigg\rangle \begin{array}{l}\textbf{fun}\ F_*(S, args_*\ldots)\\ \quad \mathcal{A}(F_*) := S\\ \quad \textbf{sim}\ \binom{S}{\mathcal{A}(F_*)}\ \textbf{do}\\ \qquad stmt\end{array}$$

Rule 2 Statement sequence

$$\begin{array}{l}\textbf{sim}\ \binom{S}{D}\ \textbf{do}\\ \quad stmt_1\\ \quad stmt_2\end{array} \Bigg\rangle \begin{array}{l}\textbf{sim}\ \binom{S}{D}\ \textbf{do}\\ \quad stmt_1\\ \textbf{sim}\ \binom{S}{D}\ \textbf{do}\\ \quad stmt_2\end{array}$$

Rule 3 Expression evaluation and assignment

$$\begin{array}{l}\textbf{sim}\ \binom{S}{D}\ \textbf{do}\\ \quad v := E\end{array} \Bigg\rangle \begin{array}{l}v_* := eval_*(E, S \cap D)\end{array}$$

Rule 4 Conditional branch statement

$$\begin{array}{l}\textbf{sim}\ \binom{S}{D}\ \textbf{do}\\ \quad \textbf{if}\ E\ \textbf{then}\\ \qquad stmt_t\\ \quad \textbf{else}\\ \qquad stmt_f\end{array} \Bigg\rangle \begin{array}{l}e_* := eval_*(E, S \cap D)\\ T := \{i \mid e_i = true\}\\ \textbf{sim}\ \binom{T}{D}\ \textbf{do}\\ \quad stmt_t\\ \textbf{sim}\ \binom{S \setminus T}{D}\ \textbf{do}\\ \quad stmt_f\end{array}$$

Rule 5 While loop

$$\begin{array}{l}\textbf{sim}\ \binom{S}{D}\ \textbf{do}\\ \quad \textbf{while}\ E\ \textbf{do}\\ \qquad stmt\end{array} \Bigg\rangle \begin{array}{l}\quad \mathcal{A}(\phi) := S\\ \phi\text{: }\textbf{while}\ D \cap \mathcal{A}(\phi) \neq \varnothing\ \textbf{do}\\ \qquad e_* := eval_*(E, \mathcal{A}(\phi) \cap D)\\ \qquad T := \{i \mid e_i = true\}\\ \qquad \mathcal{A}(\phi) := \mathcal{A}(\phi) \cap T\\ \qquad \textbf{sim}\ \binom{T}{D \cap \mathcal{A}(\phi)}\ \textbf{do}\\ \qquad\quad stmt\end{array}$$

Rule 6 Loop ϕ break statement

$$\begin{array}{l}\textbf{sim}\ \binom{S}{D}\ \textbf{do}\\ \quad \textbf{break}\end{array} \Bigg\rangle \begin{array}{l}\mathcal{A}(\phi) := \mathcal{A}(\phi) \setminus (S \cap D)\end{array}$$

Rule 7 For loop

$$\begin{array}{l}\textbf{sim}\ \binom{S}{D}\ \textbf{do}\\ \quad \textbf{for}\ v \in E\ \textbf{do}\\ \qquad stmt\end{array} \Bigg\rangle \begin{array}{l}\quad \mathcal{A}(\phi) := S\\ \phi\text{: }\textbf{for}\ v_* \in collect_*(E, S \cap D)\ \textbf{do}\\ \qquad \textbf{sim}\ \binom{S \cap \mathcal{A}(v_*)}{D \cap \mathcal{A}(\phi)}\ \textbf{do}\\ \qquad\quad stmt\end{array}$$

Rule 8 Function call

$$\begin{array}{l}\textbf{sim}\ \binom{S}{D}\ \textbf{do}\\ \quad v := F(x\ldots)\end{array} \Bigg\rangle \begin{array}{l}v_* := F_*(S \cap D, x_*\ldots)\end{array}$$

Rule 9 Function ψ return statement

$$\begin{array}{l}\textbf{sim}\ \binom{S}{D}\ \textbf{do}\\ \quad \textbf{return}\ E\end{array} \Bigg\rangle \begin{array}{l}\textbf{sim}\ \binom{S}{D}\ \textbf{do}\\ \quad \mathcal{R}(\psi) := E\\ \mathcal{A}(\psi) := \mathcal{A}(\psi) \setminus (S \cap D)\\ \textbf{if}\ \mathcal{A}(\psi) = \emptyset\ \textbf{then return}\ \mathcal{R}(\psi)\end{array}$$

An example

華東師範大學 EAST CHINA NORMAL UNIVERSITY

Rule 7 For loop

$$\begin{array}{l}\textbf{sim } \binom{S}{D} \textbf{ do}\\ \quad \textbf{for } v \in E \textbf{ do}\\ \quad\quad stmt\end{array} \quad \rangle \quad \begin{array}{l}\mathcal{A}(\phi) := S\\ \phi\text{: } \textbf{for } v_* \in collect_*(E, S \cap D) \textbf{ do}\\ \quad \textbf{sim } \binom{S \cap \mathcal{A}(v_*)}{D \cap \mathcal{A}(\phi)} \textbf{ do}\\ \quad\quad stmt\end{array}$$

- Sim statement *interleaving marker*
- *S* scope constraints *D* dynamic constraints
- *E* expression *v* variable
- $\mathcal{A}(\phi)$ active set *collect*∗() function
- A(v∗) active set

Locality optimized Data Layout

华东师范大学
EAST CHINA NORMAL UNIVERSITY

- **Global property**
  - **Vertex's property in same memory location**
- **Local variables**
  - **Collocate their values in all program instances**

Experiment Setup

6 graph algorithms

PageRank   Triangle counting   BFS DFS   closeness centrality
Tarjan's strongly connected components

store graphs in the compressed sparse row (CSR) format

Ubuntu Linux 15.10

two Intel Xeon E5-2660 v2 CPUs having 20 logical threads at 2.2GHz and 256GB of main memory.

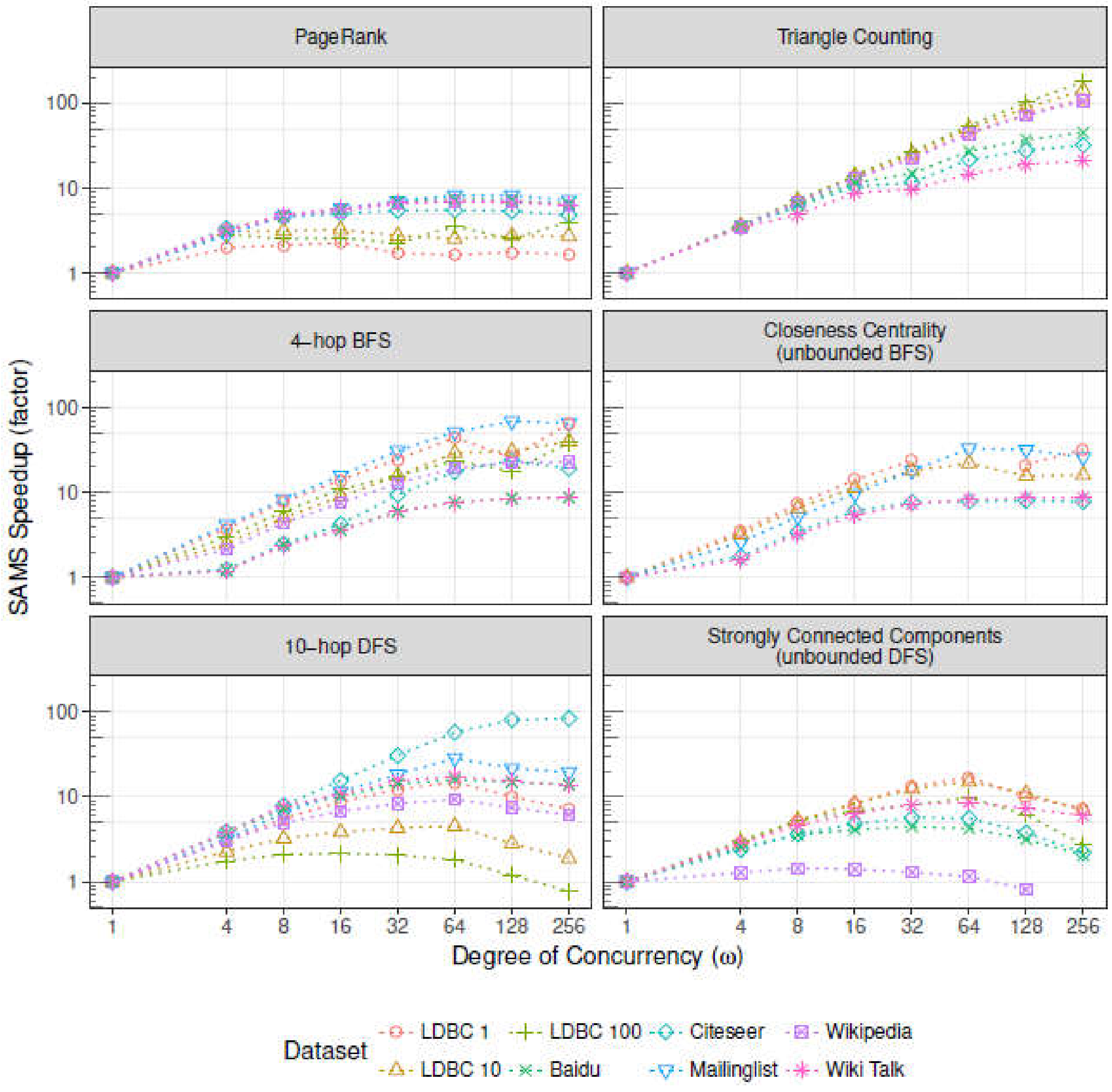
Datasets

Table 2: Properties of the evaluated data sets.

Graph	Vertices (k)	Edges (k)	∅ Degree	Diameter	∅ Diameter	Directed	Temporal
LDBC 1	34.4	1,010.6	29.4	5	2.8		X
LDBC 10	226.9	10,141.4	44.7	6	3.0		X
LDBC 100	1,611.9	101,747.9	63.1	6	3.2		X
Baidu	2,753.2	17,643.7	6.4	24	6.6	X	
Citeseer	384.4	1,751.5	4.6	70	18.6	X	
Mailinglist	27.9	1,014.1	36.3	112	7.2	X	X
Wiki Talk	2,502.0	5,021.4	2.0	14	5.0	X	
Wikipedia	1,870.7	39,953.1	21.4	376	4.6	X	X

evaluate all algorithms for 512 distinct graph snapshots, the first comprises 80% of all edges and their incident vertices

Speedup



- Speedup of SAMS over snapshot-at-a-time execution for varying degrees of concurrency.

competitor

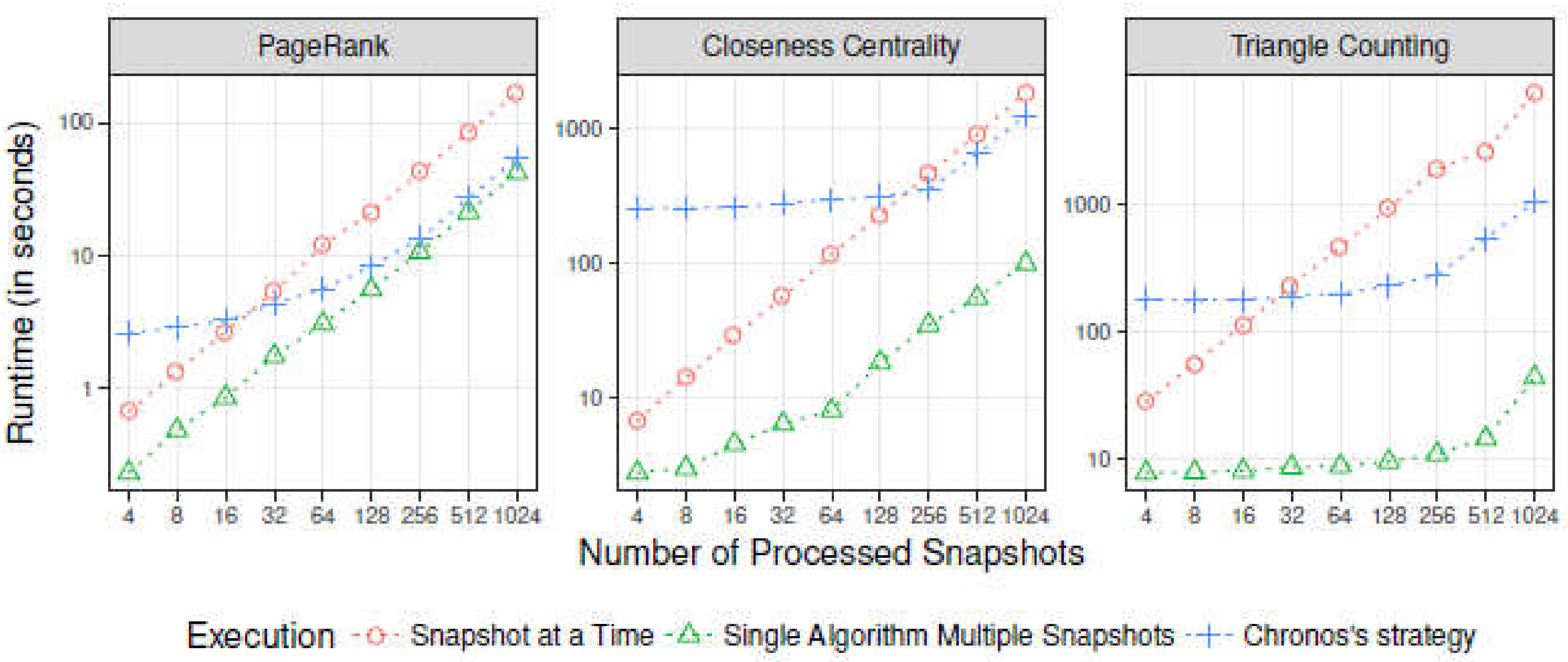


Figure 4: Absolute runtime to process a varying number of snapshots, using the LDBC 100 graph.

summary

華東師範大學 EAST CHINA NORMAL UNIVERSITY

- Inspiration:
  - Automatically transform
  - sharing common computation

- Deficiency:
  - instable
  - only for concurrency ≤256

- Further improvement:
  - Out-of-core and distributed processing