Clock-G: Système de gestion des graphes temporels pour le domaine d'objets connectés

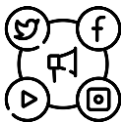
Maria Massri, Researcher, Orange
Philippe Raipin, Chef de projet de recherche Web of Things platform, Orange
Zoltan Miklos, Maître de conférence, Université de Rennes

Per3S

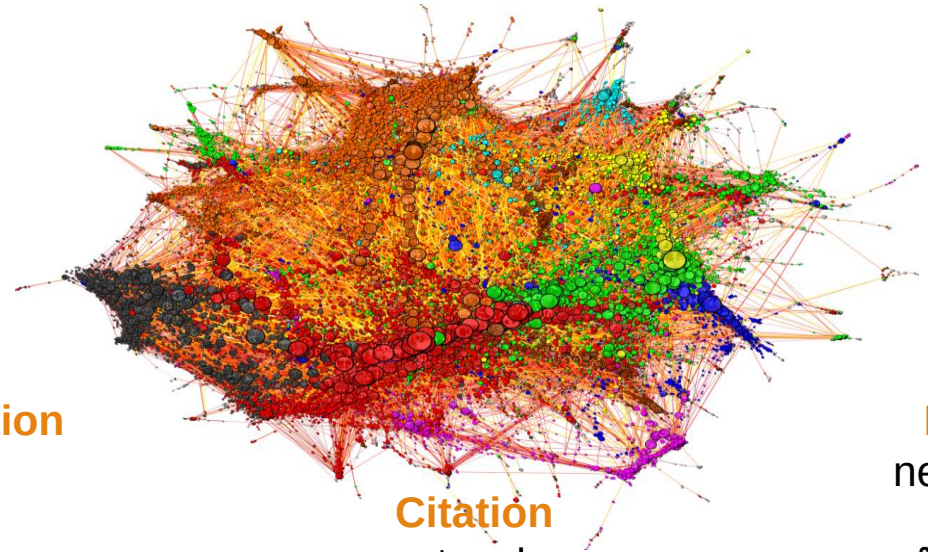
Mai 30, 2023

Introduction

Social networks



Transportation networks



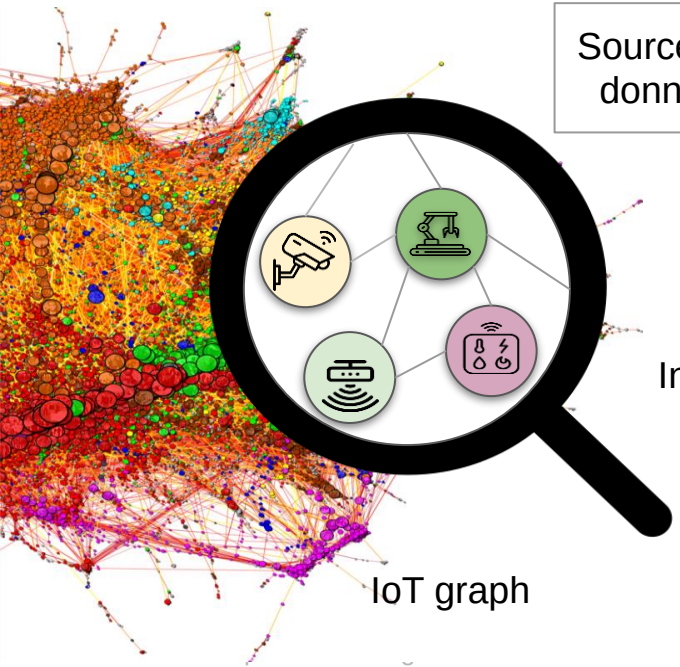
Citation networks

IoT networks

Brain networks

Contexte

- **Thing'in**[1] est une plateforme développée par Orange qui assure la gestion d'un graphe d'objets connectés (IoT) et fournit des fonctionnalités avancées de requêtage et de visualisation du graphe.



IoT graph

Sources de données

Ingestion de données

Thing'in platform

Graph database

Requêtage Visualisation

Clients

Entreprises

Administrations publiques

Utilisateurs individuels

Motivation

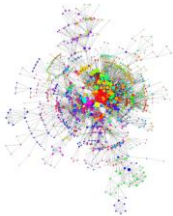
Graph Queries

Appareils IoT dans un bâtiment ?

Capteurs connectés à un appareil IoT ?

Produits manufacturés ?

Thing'in



Base de données graphe

Temporal Graph Queries

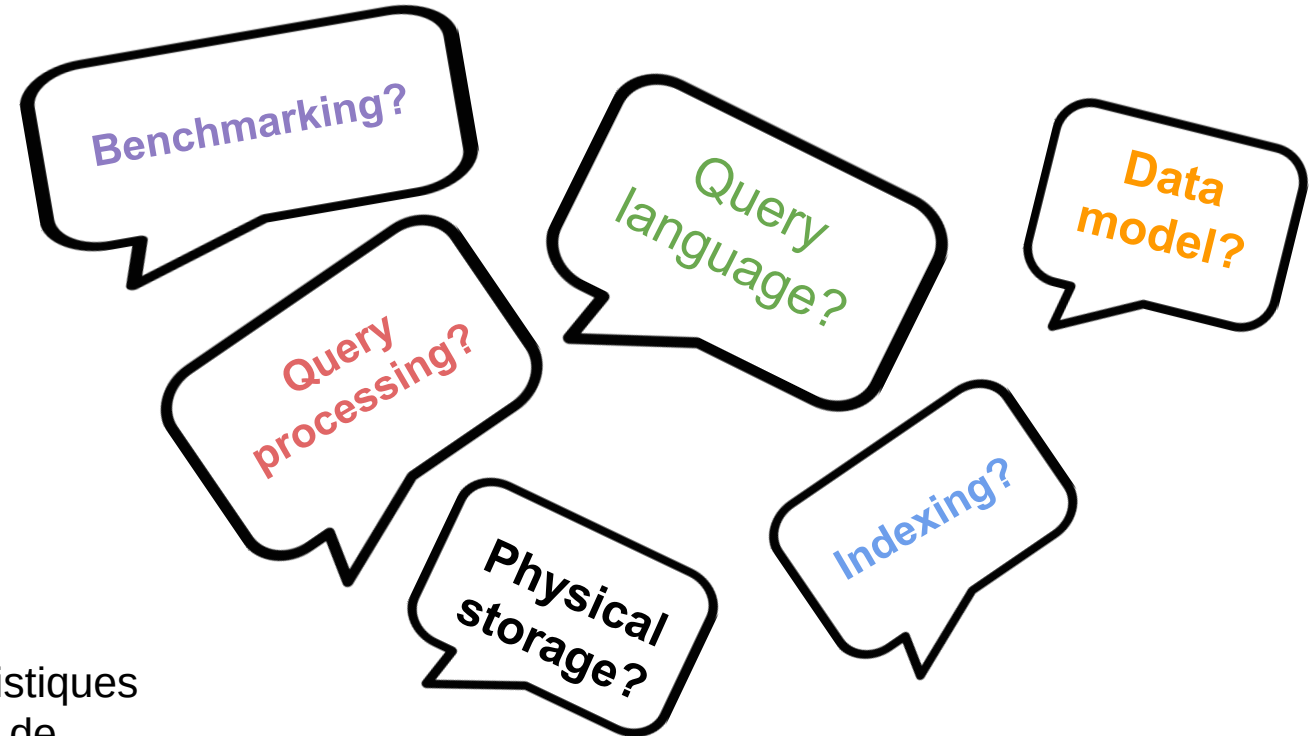
Les appareils IoT dans un bâtiment **avant une défaillance du système**?

Les machines causant des **retards** de livraison?

Cycle de vie d'un produit tout au long de la chaine de production ?

Objectif: Conception d'un prototype d'un système de gestion de graphes temporels

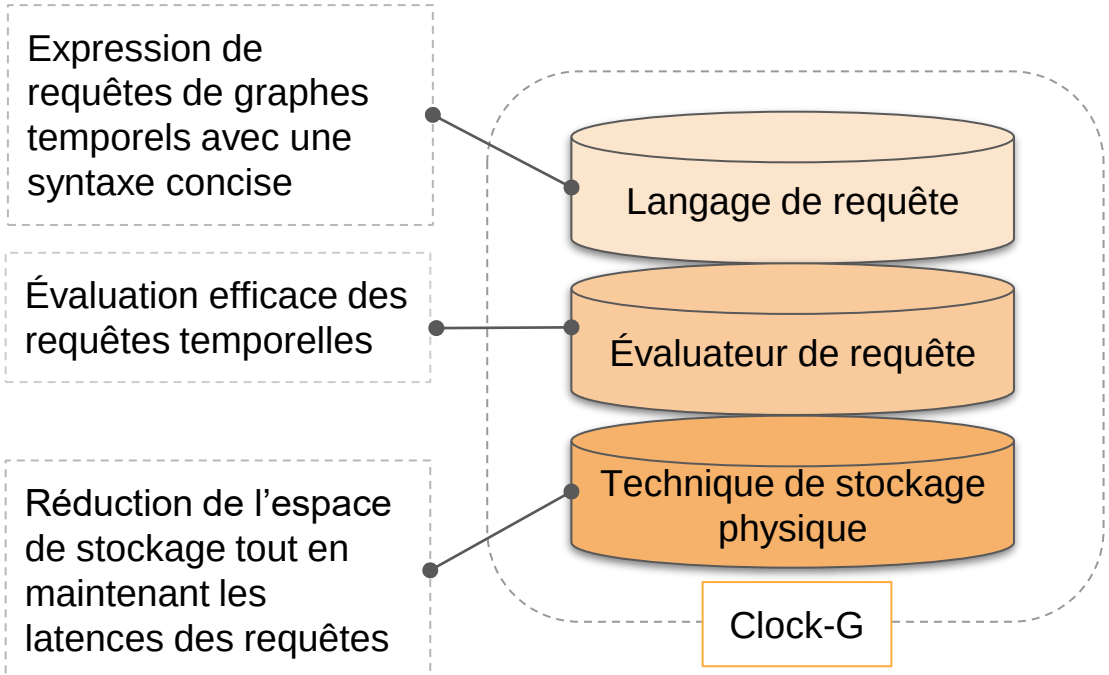
Défis



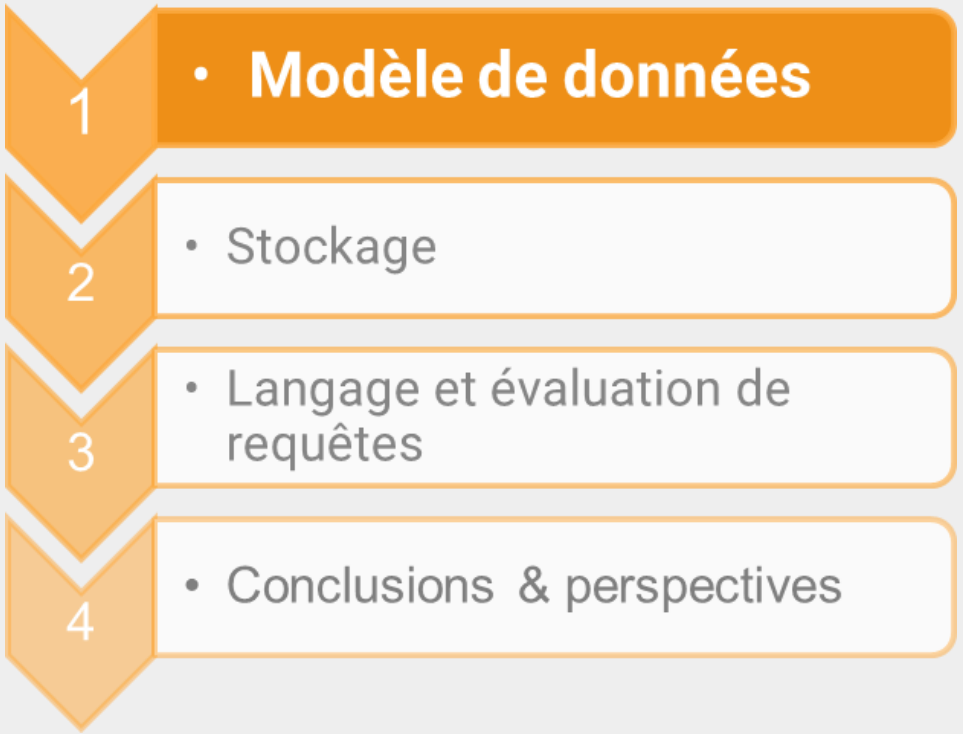
Nombreuses caractéristiques de modélisation et de conception!

(Pitoura, 2017)

Présentation générale de Clock-G



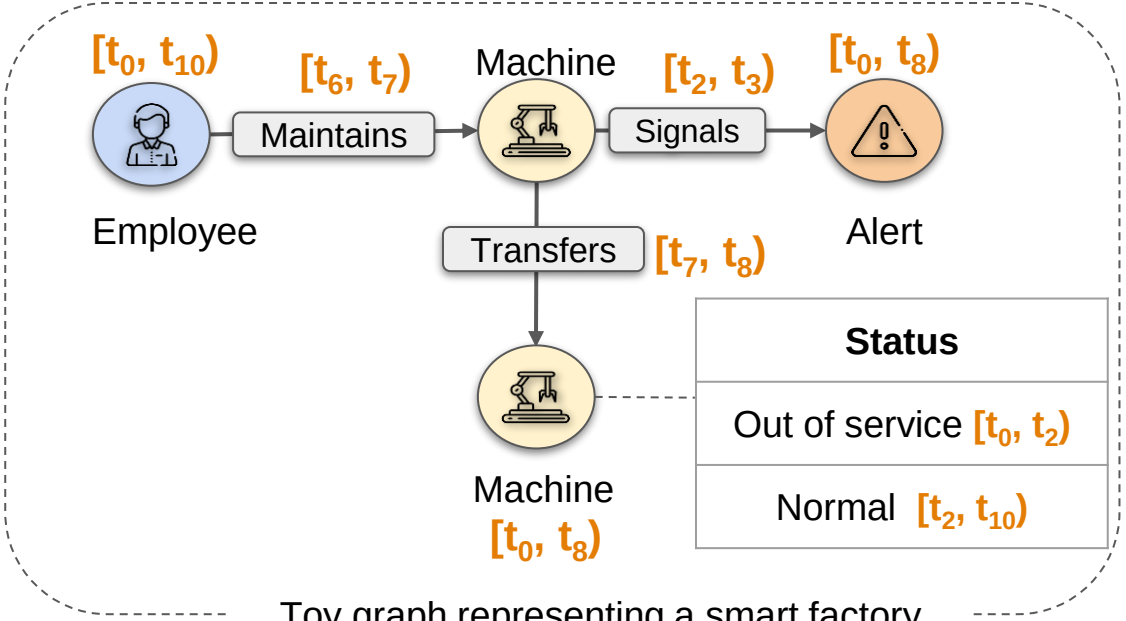
Plan


1
• Modèle de données
2
• Stockage
3
• Langage et évaluation de requêtes
4
• Conclusions & perspectives

Modèle de graphes de propriétés temporels

Temporal property graph (TPG) model

- Une collection de **nœuds** et d'**arêtes**, chacun ayant un ensemble d'étiquettes et un ensemble d' **intervalles de validité**.
- Chaque entité peut avoir un ensemble de **propriétés** composés d'un ensemble de paires **valeur/intervalle** de validité.

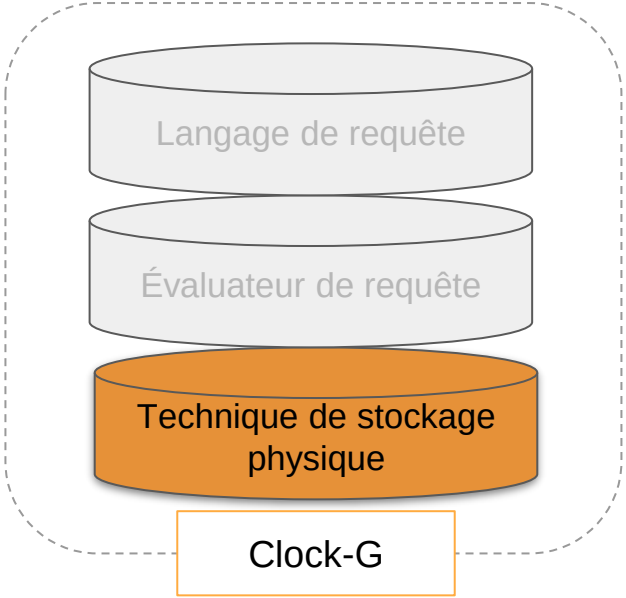


Toy graph representing a smart factory

Outline


1
• Modèle de données
2
• Stockage
3
• Langage et évaluation de requêtes
4
• Conclusions & perspectives

Comment réduire l'espace de stockage tout en maintenant le temps de réponse des requêtes?



État de l'art sur les méthodes de stockage

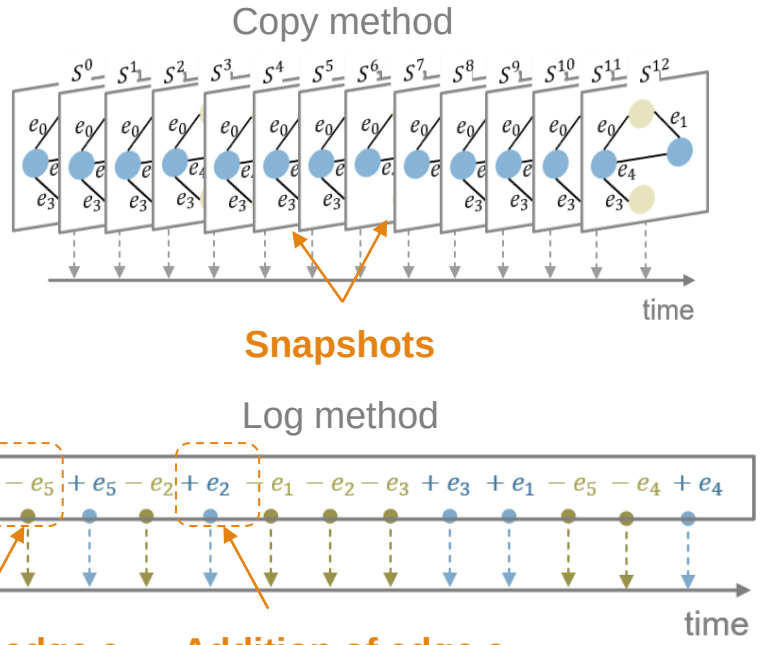
Il existe deux techniques de base pour stocker des graphes temporels : Copy et Log (Salzberg and Tsotras, 1999)

Méthode Copy Stockage de captures instantanées complètes du graphe (Snapshots).

+ Évaluation de requêtes rapide.
- Consommation d'espace élevée.
- Perte de données.

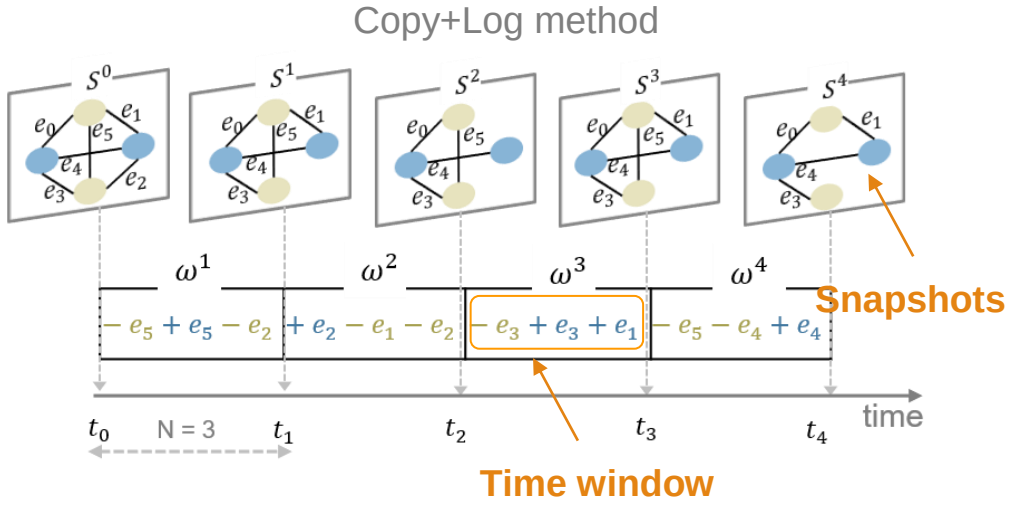
Méthode Log Stockage de mises à jour (logs).

+ Espace de stockage optimal.
- Évaluation de requêtes lente.



Méthode Copy+Log Stockage de *logs* dans des fenêtres temporelles et des *snapshots*.
ImmortalGraph **(Miao et al., 2015)**, Auxo **(Han et al., 2019)**, TGI **(Khurana and Deshpande, 2013)**

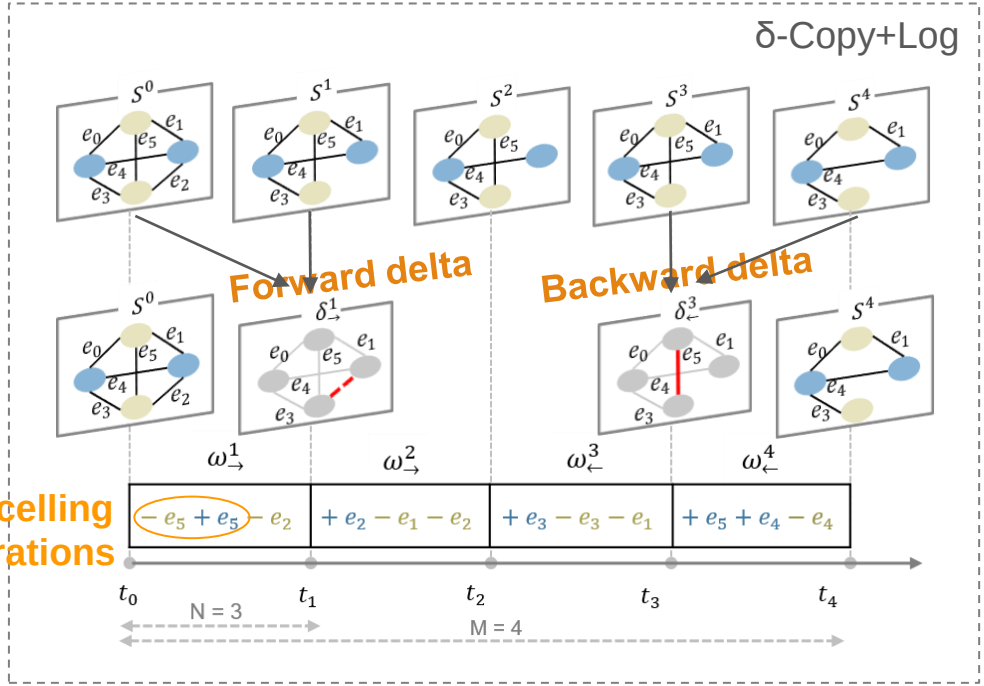
+ Réduction de l'espace de recherche.
- Le stockage de snapshots complets est gourmand en espace, notamment pour les graphes à grande échelle.



δ-Copy+Log: Storage internals

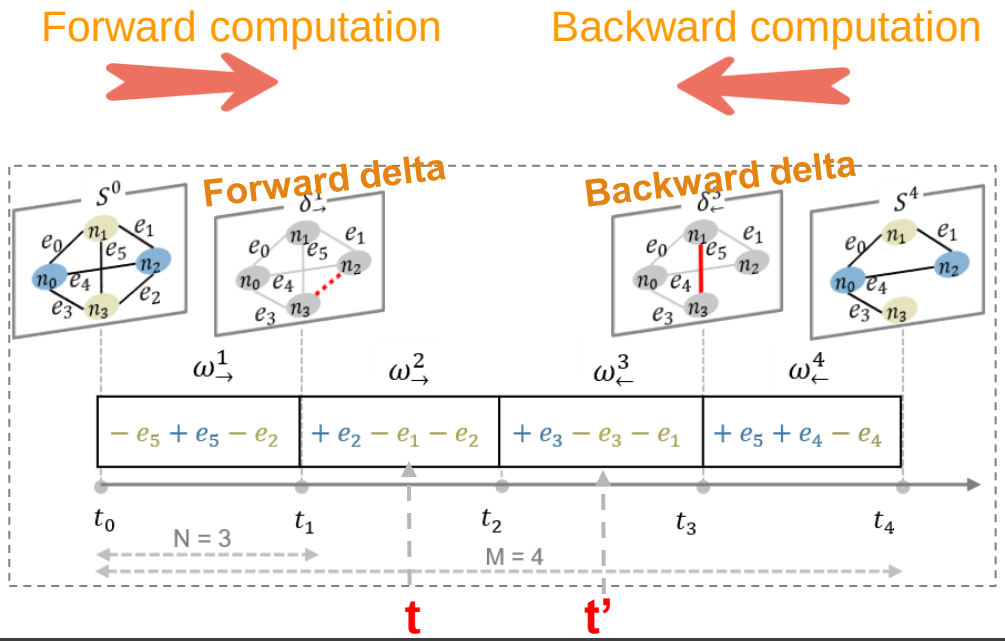
δ-Copy+Log (Clock-G ICDE 2022) **stocke des deltas au lieu des snapshots.**

- **Forward delta** = S^i - S^{i-1}
- **Backward delta** = S^{i-1} - S^i
- ***M***: Nombre de fenêtres temporelles entre les snapshots.
- Varying the parameter M can tune the performance of our system.



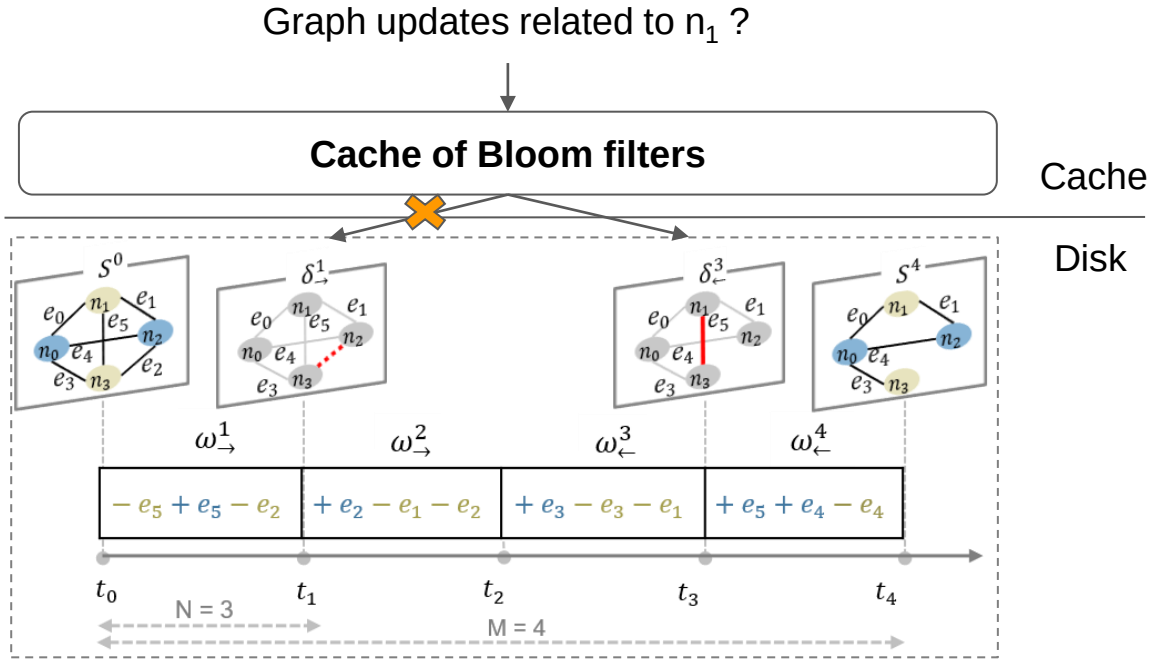
Optimisation du requêtage

- Le résultat d'une requête est évalué:
  - En format **forward** si t est plus proche de l'instantané de départ.
  - En format **backward** si t est plus proche de l'instantané finale.



Optimisation du requêtage

Chaque delta est associée à un **filtre de Bloom[1]** afin de réduire les accès inutiles au disque.



1. Le filtre de Bloom est une structure de données probabiliste couramment utilisée dans les systèmes de gestion de bases de données (DBMS) pour tester l'existence d'un élément dans un ensemble.

Environnement de test

- Tous les tests d'évaluation ont été réalisés avec la **configuration suivante** :
  - 32 Intel(R) Xeon(R) E5-2630L v3 1.80GHz CPUs
  - 264 GB RAM
  - 1TB SSD

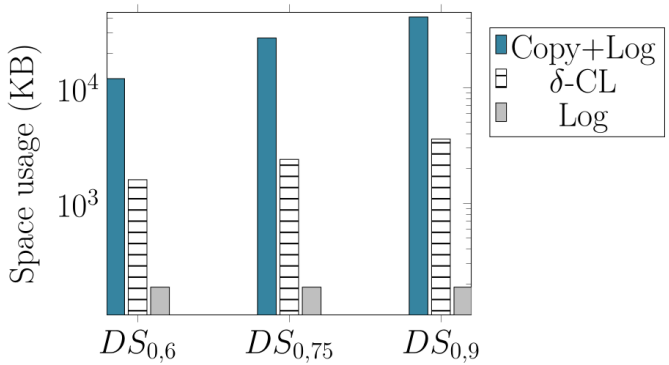
- Graphes synthétiques (DS_{pa}):
  - **1 million** nœuds
  - **50 million** mises à jour
  - Probabilité d'ajout d'arêtes (**p_a**)

$\{DS_{0.6}, DS_{0.75}, DS_{0.9}\}$

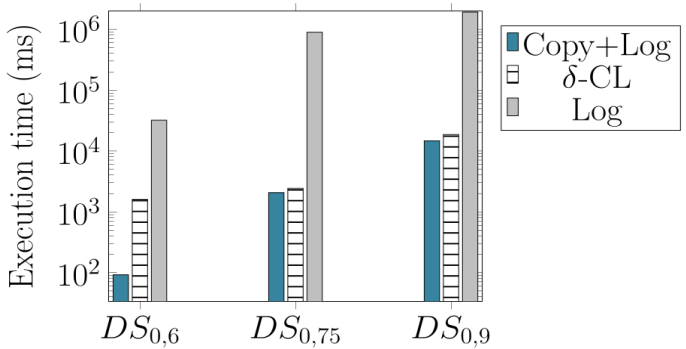
~75 % des mises à jour sont des ajouts

Résultats (1/3)

- Comparaison de la méthode δ-Copy+Log (δ-CL) avec les méthodes alternatives Copy+Log et Log
  - **Réduction d'espace** pouvant atteindre **114x** % à Copy+Log.
  - **Réduction du temps d'exécution** pouvant atteindre **443x** % à Log.
  - **Compromis** entre les performances de Copy+Log et Log.



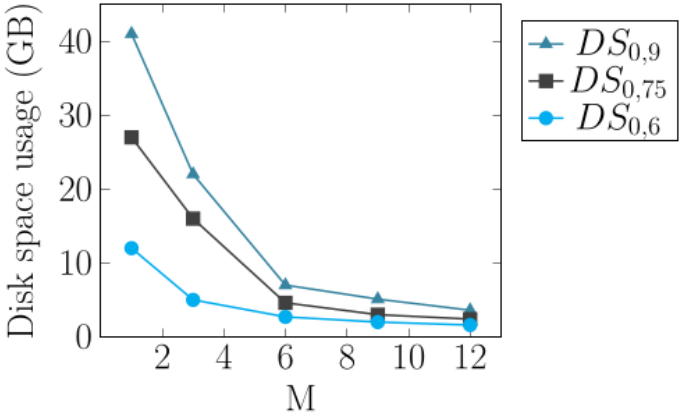
N=10K, M= 12



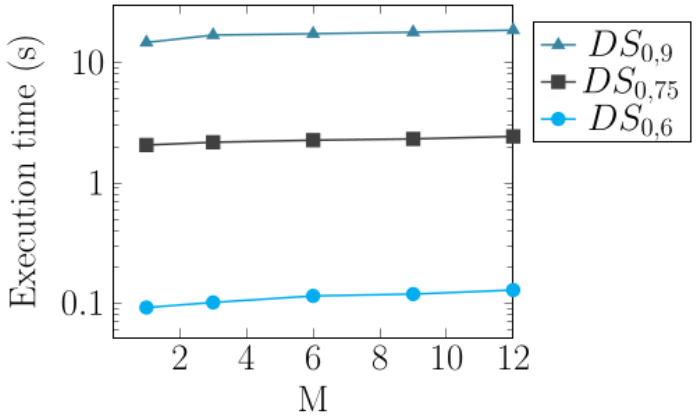
(5-Hops point queries)

Résultats (2/3)

- Effet de la variation de **M** (fenêtres temporelles entre deux instantanés).
  - L’augmentation de M réduit significativement l'utilisation de l'espace, mais entraîne une légère augmentation du temps d'exécution.



N=10k

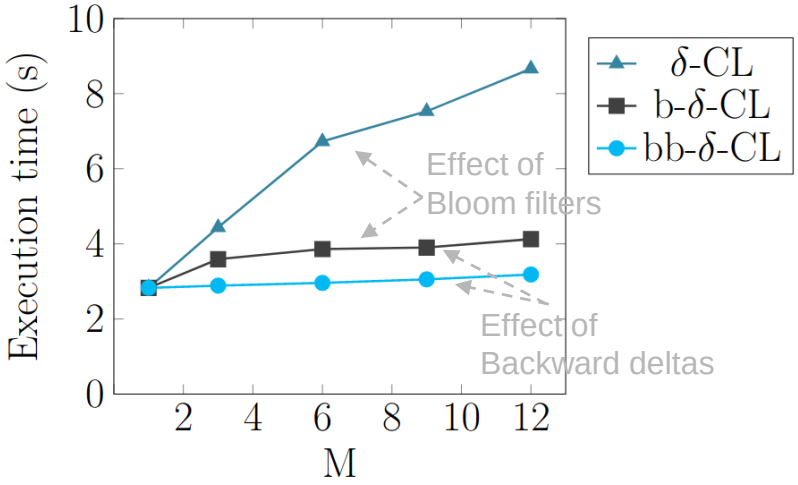


(5-Hops point queries)

Résultats (3/3)

- Évaluation des méthodes suivantes :
  - **δ-CL**: δ-Copy+Log (with only forward deltas)
  - **b-δ-CL**: Bloomed δ-Copy+Log
  - **bb-δ-CL**: Backward Bloomed δ-Copy+Log

- En utilisant des deltas arrière, le temps d'exécution est davantage réduit pour des valeurs de M plus élevées.



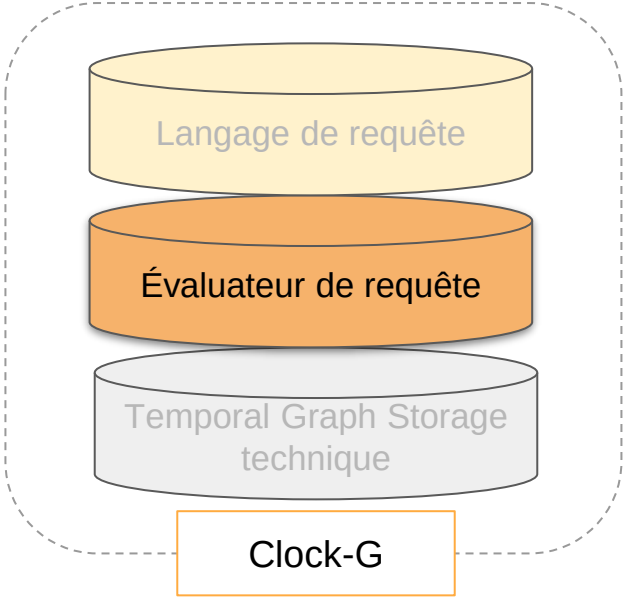
(# time windows between two snapshots)

(5-Hops point queries)

Plan


1
• Modèle de données
2
• Stockage
3
• Langage et évaluation de requêtes
4
• Conclusions & perspectives

Comment étendre un langage et un processeur de requêtes graphiques traditionnels pour faire face à la dimension temporelle?



T-Cypher

- **T-Cypher**[1]: langage de requête de graphes temporels étendant Cypher **(Francis et al., 2018)** avec des expressions temporelles.

- T-Cypher permet l'expression de requêtes temporelles de graph pattern, de navigation et d'agrégation avec une syntaxe concise.

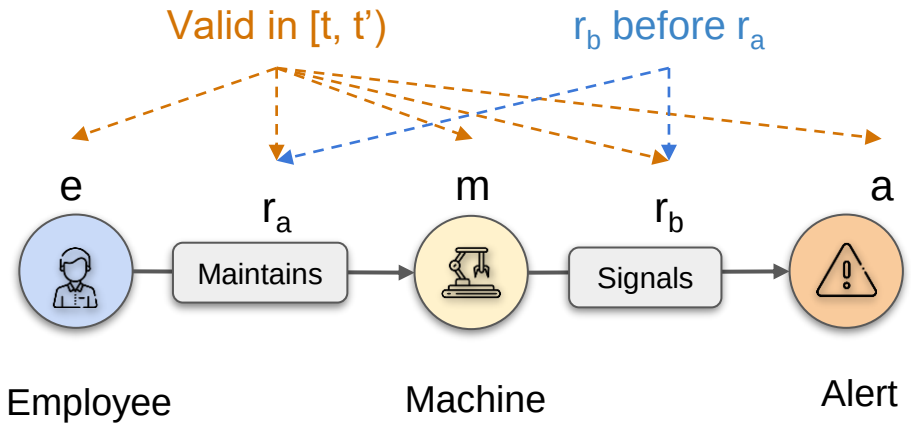
- Cypher
  - Populaire
  - Syntaxe User-friendly
  - Intégré dans les normes de **GQL** **(Deutsch et al., 2022)**

1. https://wiki.thinginthefuture.com/public/Historization

Syntaxe de T-Cypher

Requête de *temporal graph pattern matching*

Retourner les machines signalant une alerte avant une maintenance effectuée par un employé dans l'intervalle [t, t').



Cypher query

```
MATCH (e) -[ra]-> (m) -[rb]-> (a)
RETURN m
```

Pattern expression

Return expression

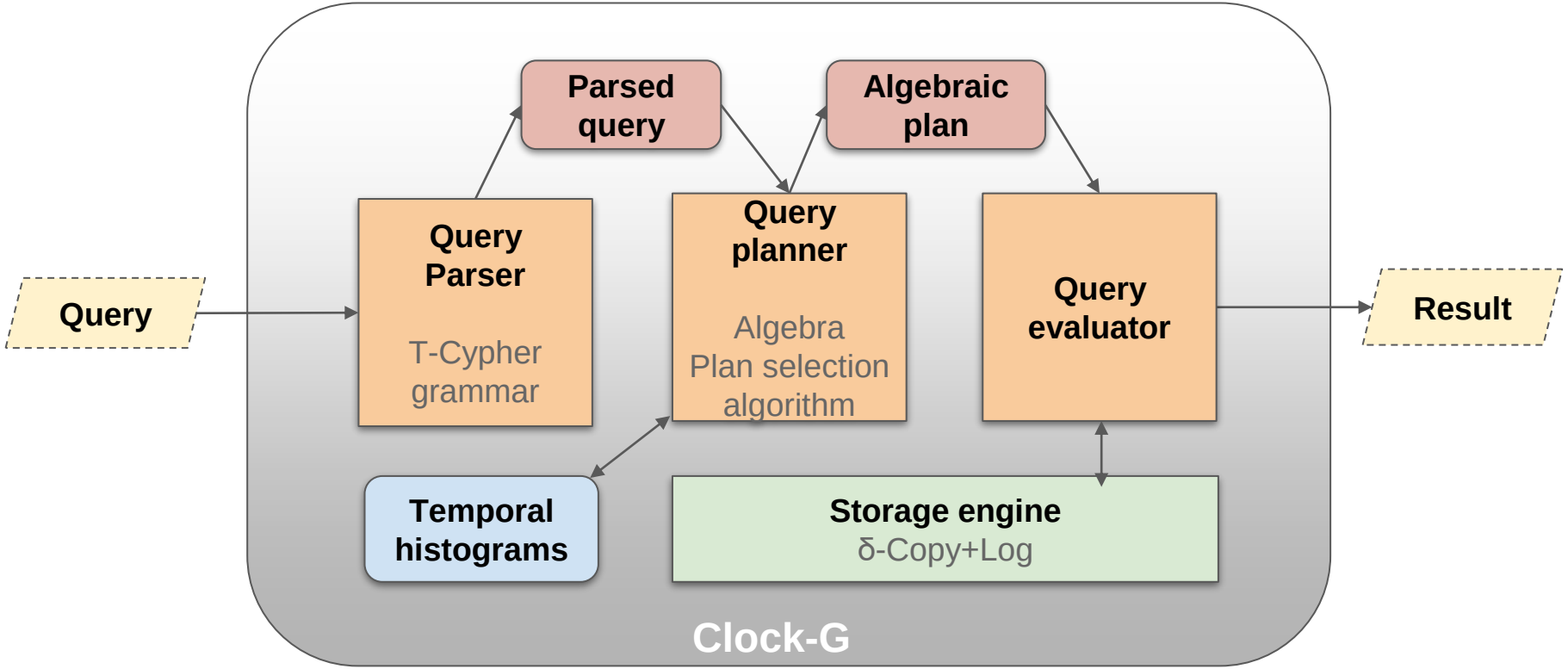
T-Cypher query

```
RANGE_SLICE [t, t')
MATCH (e) -[ra]-> (m) - [rb]-> (a)
WHERE rb@T BEFORE ra@T
RETURN m
```

Time slicing clause

Temporal constraint

Pipeline d'évaluation des requêtes T-Cypher dans Clock-G



Traditional query processing pipeline described in the book ***Querying graphs*** **(Bonifati et al., 2018)**

Plan

1. Modèle de données
2. Stockage
3. Langage et évaluation de requêtes
4. **Conclusions & perspectives**

Conclusions

- Conception d'un prototype d'un système de gestion de graphes temporels, **Clock-G**, comprenant :

  - **Un langage de requête** de graphes temporels (Intégration industrielle dans Thing'in[1]).

  - **Une technique de stockage** efficace en termes d'espace qui maintient les latences de requête.

  - **Un processeur de requêtes** qui évalue efficacement nos requêtes de graphes temporels proposées.

1. https://wiki.thinginthefuture.com/public/Historization

Perspectives

- Ajout d'une prise en charge spatiale pour suivre les positions géographiques des objets IoT.

- Exploiter l'historique pour entraîner des modèles d'apprentissage et anticiper les défaillances du système.

Merci de votre attention

Maria Massri

Publications

(RTGEN++ FGCS 2023) Maria Massri, Zoltan Miklos, Philippe Raipin, and Pierre Meye, Amaury Bouchra Pilet, and Thomas Hassan, “**RTGEN++: A relative temporal graph generator**,” ***FGCS***, 2022.

(Clock-G ICDE 2022) Maria Massri, Zoltan Miklos, Philippe Raipin, and Pierre Meye, “**Clock-G: A temporal graph management system with space-efficient storage technique**,” in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 2022, pp. 2263–2276. doi: 10.1109/ICDE53745.2022.00215.

(RTGEN++ FGCS) Maria Massri, Zoltan Miklos, Philippe Raipin, and Pierre Meye, Amaury Bouchra Pilet, and Thomas Hassan, “**Clock-G: Temporal graph management system**,” ***TLKDS***, (Under revision).

(RTGEN DATAPLAT@EDBT 2022) Maria Massri, Zoltan Miklos, Philippe Raipin, and Pierre Meye, “**RTGEN: A Relative Temporal Graph GENerator**,” in *DATAPLAT workshop at the EDBT/ICDT 2022 Joint Conference*, 2022.

(GDBAlive JAIT 2020) Maria Massri, Philippe Raipin, and Pierre Meye, “**GDBAlive: a Temporal Graph Database Built on Top of a Columnar Data Store**,” *Journal of Advances in Information Technology. JAIT*, Sep. 2020, Available: https://hal.inria.fr/hal-02937283

(TGDBMS BDA 2021) Maria Massri, “**Towards designing a temporal graph management system (Doctoral paper)**,” in *Actes de la conférence BDA 2021*, 2021, p. 91.

References

Aiello, William, Fan Chung, and Linyuan Lu. 2001. “A Random Graph Model for Power Law Graphs.” *Experimental Mathematics* 10 (1): 53–66.

Alami, Karim, Radu Ciucanu, and Engelbert Mephu Nguifo. 2017. “Synthetic Graph Generation from Finely-Tuned Temporal Constraints.” In *TD-LSG@ PKDD/ECML*, 44–47.

Allen, James F. 1983. “Maintaining Knowledge about Temporal Intervals.” *Communications of the ACM* 26 (11): 832–43.

Arenas, Marcelo, Pedro Bahamondes, Amir Aghasadeghi, and Julia Stoyanovich. 2022. “Temporal Regular Path Queries.” In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 2412–25. https://doi.org/10.1109/ICDE53745.2022.00226.

Bagan, Guillaume, Angela Bonifati, Radu Ciucanu, George HL Fletcher, Aurélien Lemay, and Nicky Advokaat. 2016. “GMark: Schema-Driven Generation of Graphs and Queries.” *IEEE Transactions on Knowledge and Data Engineering* 29 (4): 856–69.

Benyahia, Oualid, Christine Largeron, Baptiste Jeudy, and Osmar R Zaïane. 2016. “Dancer: Dynamic Attributed Network with Community Structure Generator.” In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 41–44. Springer.

Bonifati, Angela, George Fletcher, Hannes Voigt, and Nikolay Yakovets. 2018. “Querying Graphs.” *Synthesis Lectures on Data Management* 10 (3): 1–184.

Chung, Fan, and Linyuan Lu. 2002. “Connected Components in Random Graphs with given Expected Degree Sequences.” *Annals of Combinatorics* 6 (2): 125–45.

References

Debrouvier, Ariel, Eliseo Parodi, Matías Perazzo, Valeria Soliani, and Alejandro Vaisman. 2021. “A Model and Query Language for Temporal Graph Databases.” *The VLDB Journal* 30 (5): 825–58. https://doi.org/10.1007/s00778-021-00675-4.

Deutsch, Alin, Nadime Francis, Alastair Green, Keith Hare, Bei Li, Leonid Libkin, Tobias Lindaaker, et al. 2022. “Graph Pattern Matching in GQL and SQL/PGQ.” In *Proceedings of the 2022 International Conference on Management of Data*, 2246–58. Philadelphia PA USA: ACM. https://doi.org/10.1145/3514221.3526057.

Erling, Orri, Alex Averbuch, Josep Larriba-Pey, Hassan Chafi, Andrey Gubichev, Arnau Prat, Minh-Duc Pham, and Peter Boncz. 2015. “The LDBC Social Network Benchmark: Interactive Workload.” In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 619–30. SIGMOD ’15. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2723372.2742786.

Flamary, Rémi, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, et al. 2021. “POT: Python Optimal Transport.” *Journal of Machine Learning Research* 22 (78): 1–8.

Francis, Nadime, Alastair Green, Paolo Guagliardo, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Stefan Plantikow, Mats Rydberg, Petra Selmer, and Andrés Taylor. 2018. “Cypher: An Evolving Query Language for Property Graphs.” In *Proceedings of the 2018 International Conference on Management of Data*, 1433–45.

Ghrab, Amine, Oscar Romero, Sabri Skhiri, Alejandro A. Vaisman, and Esteban Zimányi. 2016. “GRAD: On Graph Database Modeling.” *CoRR* abs/1602.00503. http://arxiv.org/abs/1602.00503.

Han, Wentao, Kaiwei Li, Shimin Chen, and Wenguang Chen. 2019. “Auxo: A Temporal Graph Management System.” *Big Data Mining and Analytics* 2 (1): 58–71. https://doi.org/10.26599/BDMA.2018.9020030.

References

He, Huahai, and Ambuj K. Singh. 2010. “Query Language and Access Methods for Graph Databases.” In *Managing and Mining Graph Data*, edited by Charu C. Aggarwal and Haixun Wang, 125–60. Boston, MA: Springer US. https://doi.org/10.1007/978-1-4419-6045-0_4.

Holland, Paul W, Kathryn Blackmond Laskey, and Samuel Leinhardt. 1983. “Stochastic Blockmodels: First Steps.” *Social Networks* 5 (2): 109–37.

Hölsch, Jürgen, and Michael Grossniklaus. 2016. “An Algebra and Equivalences to Transform Graph Patterns in Neo4j.” In *Proceedings of the Workshops of the EDBT/ICDT 2016 Joint Conference (EDBT/ICDT 2016)*, edited by Themis Palpanas and Kostas Stefanidis. CEUR Workshop Proceedings. http://ceur-ws.org/Vol-1558/paper24.pdf.

Junghanns, Martin, Max Kießling, Alex Averbuch, André Petermann, and Erhard Rahm. 2017. “Cypher-Based Graph Pattern Matching in Gradoop.” In . GRADES’17. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3078447.3078450.

Khurana, Udayan, and Amol Deshpande. 2013. “Efficient Snapshot Retrieval over Historical Graph Data.” In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, 997–1008. https://doi.org/10.1109/ICDE.2013.6544892.

Marton, József, Gábor Szárnyas, and Dániel Varró. 2017. “Formalising OpenCypher Graph Queries in Relational Algebra.” In *Advances in Databases and Information Systems*, edited by Mārīte Kirikova, Kjetil Nørvåg, and George A. Papadopoulos, 182–96. Cham: Springer International Publishing.

Massri, Maria, Zoltan Miklos, Philippe Raipin, and Pierre Meye. 2021. “T-Cypher: A Temporal Graph Query Language.” https://project.inria.fr/tcypher/.

References

Miao, Youshan, Wentao Han, Kaiwei Li, Ming Wu, Fan Yang, Lidong Zhou, Vijayan Prabhakaran, Enhong Chen, and Wenguang Chen. 2015. "ImmortalGraph: A System for Storage and Analysis of Temporal Graphs." *ACM Transactions on Storage* 11 (3): 14:1-14:34. https://doi.org/10.1145/2700302.

Moffitt, Vera Zaychik, and Julia Stoyanovich. 2017. "Temporal Graph Algebra." In . DBPL '17. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3122831.3122838.

Pitoura, Evaggelia. 2017. "Historical Graphs: Models, Storage, Processing." In *European Business Intelligence and Big Data Summer School*, 84–111. Springer.

Rizzolo, Flavio, and Alejandro A Vaisman. 2008. "Temporal XML: Modeling, Indexing, and Query Processing." *The VLDB Journal—The International Journal on Very Large Data Bases* 17 (5): 1179–1212.

Salzberg, Betty, and Vassilis J. Tsotras. 1999. "Comparison of Access Methods for Time-Evolving Data." *ACM Comput. Surv.* 31 (2): 158–221. https://doi.org/10.1145/319806.319816.

Schmidt, Michael, Michael Meier, and Georg Lausen. 2010. "Foundations of SPARQL Query Optimization." In , 4–33. ICDT '10. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/1804669.1804675.

Semertzidis, Konstantinos, and Evaggelia Pitoura. 2016. "Durable Graph Pattern Queries on Historical Graphs." In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, 541–52. https://doi.org/10.1109/ICDE.2016.7498269.

Zhang, Tianming, Yunjun Gao, Linshan Qiu, Lu Chen, Qingyuan Linghu, and Shiliang Pu. 2020. "Distributed Time-Respecting Flow Graph Pattern Matching on Temporal Graphs." *World Wide Web* 23 (1): 609–30. https://doi.org/10.1007/s11280-019-00674-0.