Gérer et analyser les grands graphes d'entités nommées

Jocelyn Bernard

Université Claude Bernard - Lyon 1 | ReportLinker / FindOut
Équipe GOAL | Département R&D

April 13, 2016

Plan

Les graphes

- un outil de modélisation
- des structures de données
- modéliser objets et problèmes
- nombreuses applications

Le cas des grands graphes d'entités nommées

Cas classique de BigData :

- De plus en plus grands
- Données variées
- Données arrivant de plus en plus vite

Le cas des grands graphes d'entités nommées

Cas classique de BigData :

- De plus en plus grands
- Données variées
- Données arrivant de plus en plus vite

En conséquence :

- Ne tiennent plus forcément en mémoire
- Classe de problèmes difficiles (classe NP)

Des problématiques

1. Construction de grands graphes d'entités nommées
2. Stockage, Indexation et gestion des graphes
3. Analyse des graphes

Construction de grands graphes d'entités nommées

Constat : Graphe de plus en plus grand et varié avec des données plus ou moins importantes selon la temporalité.

- Graphe plus lourds

Construction de grands graphes d'entités nommées

Constat : Graphe de plus en plus grand et varié avec des données plus ou moins importantes selon la temporalité.

- Graphe plus lourds
- Graphe de moins en moins compréhensible

Construction de grands graphes d'entités nommées

Constat : Graphe de plus en plus grand et varié avec des données plus ou moins importantes selon la temporalité.

- Graphe plus lourds
- Graphe de moins en moins compréhensible
- L'évaluation d'une information doit prendre en compte la temporalité

Construction de grands graphes d'entités nommées

Constat : Graphe de plus en plus grand et varié avec des données plus ou moins importantes selon la temporalité.

- Graphe plus lourds
- Graphe de moins en moins compréhensible
- L'évaluation d'une information doit prendre en compte la temporalité

Objectif : définir un format de graphe qui permet de prendre en compte ces problématiques.

Stockage, Indexation et gestion des graphes

Constat : Nécessite des outils permettant la manipulation de graphe

- Outils distribués

Stockage, Indexation et gestion des graphes

Constat : Nécessite des outils permettant la manipulation de graphe

- Outils distribués
- Spécialisés dans les grandes données et les graphes

Stockage, Indexation et gestion des graphes

Constat : Nécessite des outils permettant la manipulation de graphe

- Outils distribués
- Spécialisés dans les grandes données et les graphes
- Permettant l'exploration des graphes

Stockage, Indexation et gestion des graphes

Constat : Nécessite des outils permettant la manipulation de graphe

- Outils distribués
- Spécialisés dans les grandes données et les graphes
- Permettant l'exploration des graphes

Objectif : regarder les outils de manipulations de graphes et les adaptés à nos problématiques

Analyse des graphes

Constat : Problèmes de classe NP

- Problèmes difficiles

Analyse des graphes

Constat : Problèmes de classe NP

- Problèmes difficiles
- Algorithmes sur le graphe de plus en plus longs

Analyse des graphes

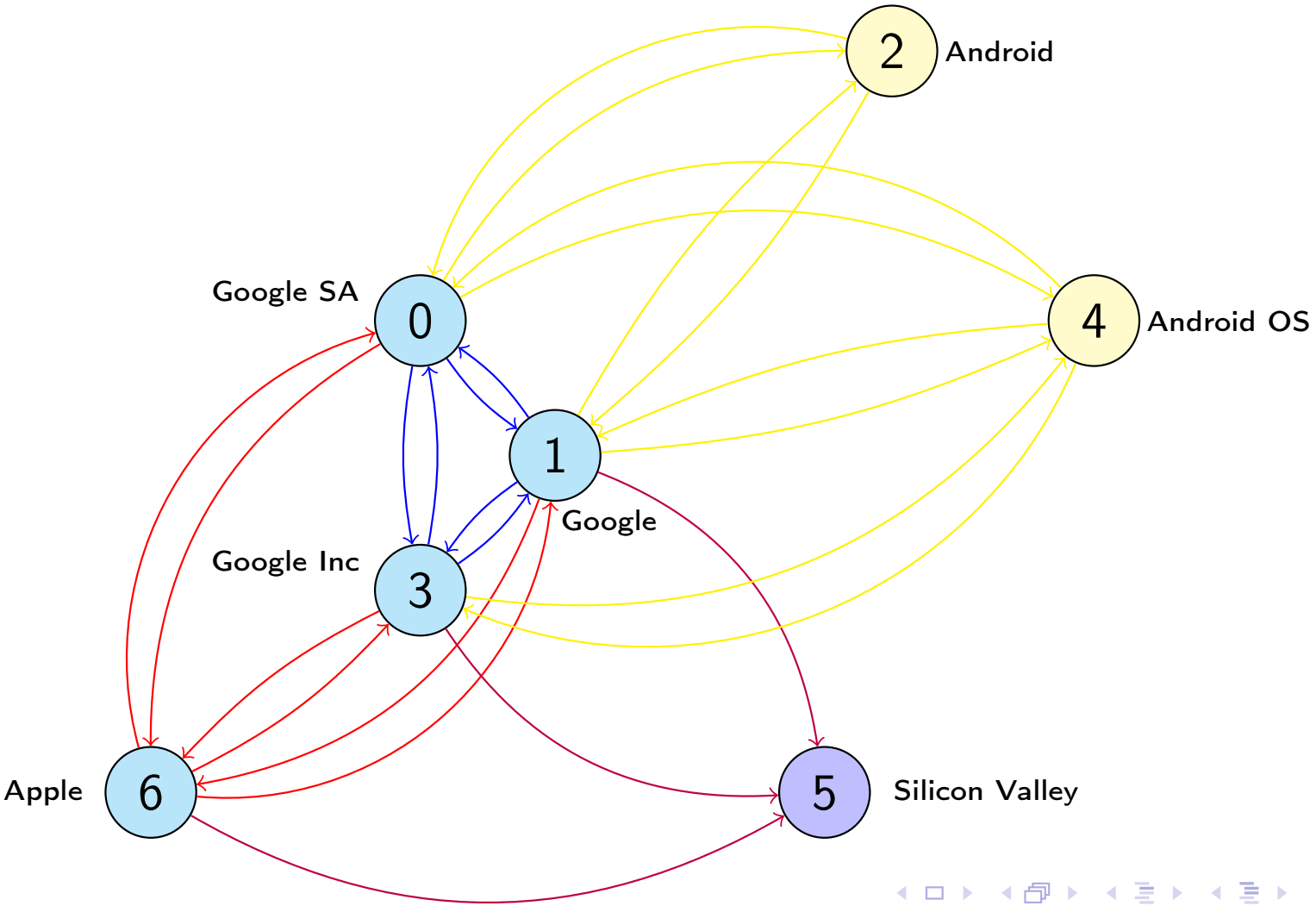
Constat : Problèmes de classe NP

- Problèmes difficiles
- Algorithmes sur le graphe de plus en plus longs

Objectif : proposer des méthodes adaptés au graphe et donc regarder les caractéristiques du graphe

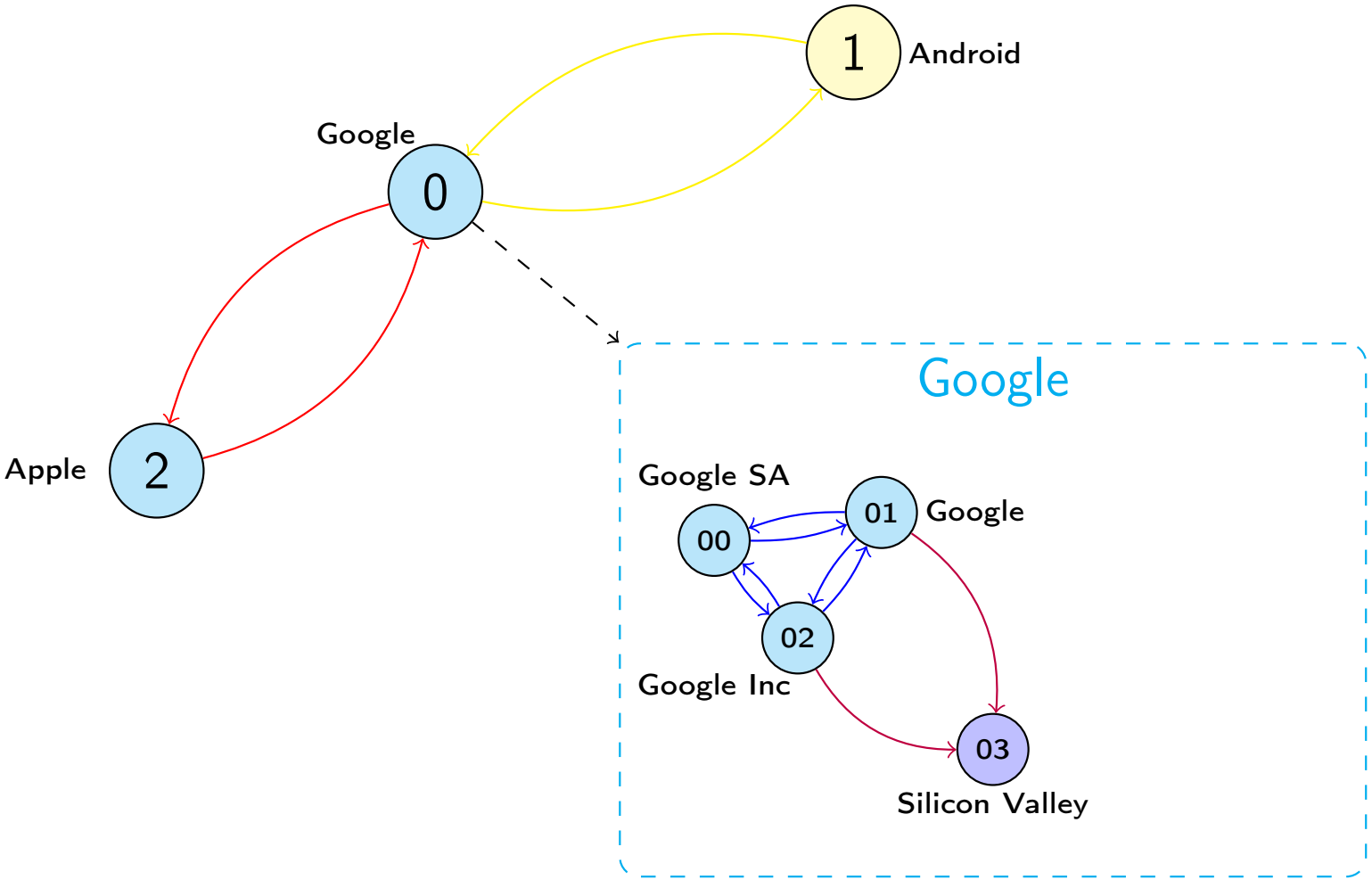
Graphe squelette

Graphe initial :



Graphe squelette

Graphe squelette :



Graphe temporel

L'attribut temporel est composé de 2 dates :

- Celle de la création de l'information
- Celle de la dernière validation de l'information

Du modèle de partition

La partition *Edge-Cut* :

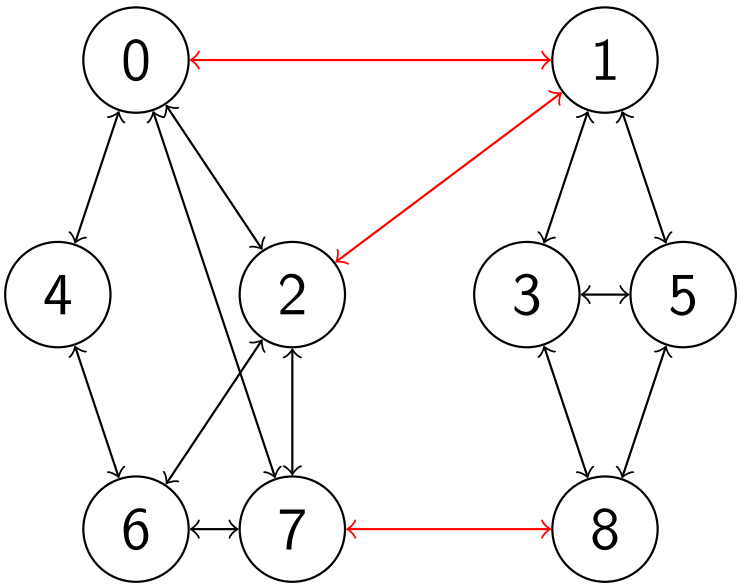


Figure: Découpe d'un graphe en deux sous-graphes par *edge-cut*

Du modèle de partition

La partition *Edge-Cut* :

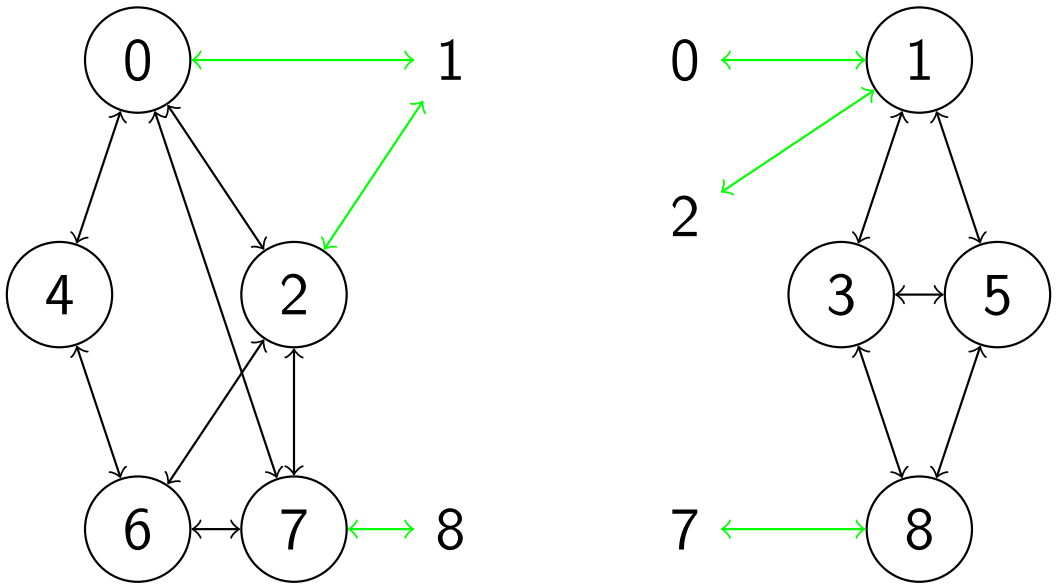


Figure: Découpe d'un graphe en deux sous-graphes par *edge-cut*

Du modèle de partition

La partition *Vertex-Cut* :

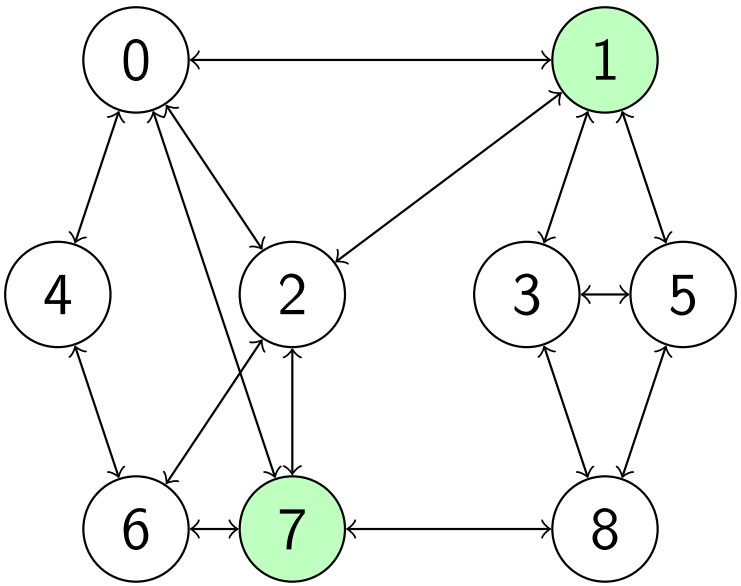


Figure: Découpe d'un graphe en deux sous-graphes par *vertex-cut*

Du modèle de partition

La partition *Vertex-Cut* :

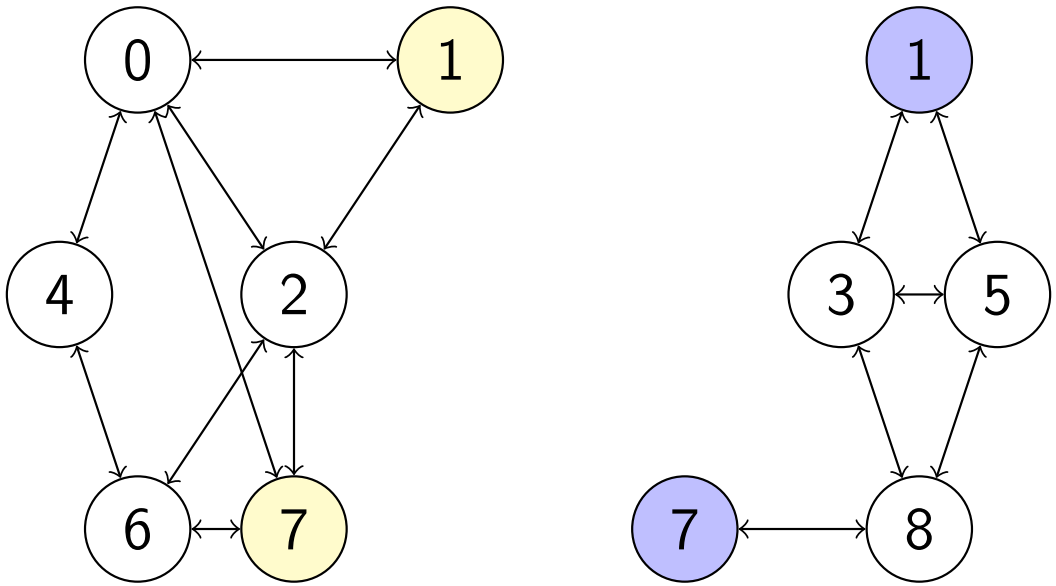


Figure: Découpe d'un graphe en deux sous-graphes par *vertex-cut*

Utilisation

Étude des papier présentant des algorithmes propres aux grands graphes dont ceux d'entités nommées sur différents problèmes :

- Problème de plus court chemin [Qi et al., 2013]
- Problème de découverte de patterns
  [Yang et al., 2014, Svendsen et al., 2015]
- Et cetera

Rappel

- Proposer un modèle de graphe adapté
- Proposer des outils adaptés pour la manipulation...
- ... et l'exploration / exploitation du graphe

Résultats initiaux

Étude théorique des logiciels de gestion de grands graphes :

Plate-forme	Année	Langage	Papier
Pregel	2010	C++	[Malewicz et al., 2010]
Giraph	2013	Java	[apa, b]
Giraph++	2013	Java	[Tian et al., 2013]
GiraphuC	2015	Java	[Han and Daudjee, 2015]
Mizan	2013	C++	[Khayyat et al., 2013]
GPS	2013	Java	[Salihoglu and Widom, 2013]
GraphLab	2009	C++	[Low et al., 2012]
PowerGraph	2012	C++	[Gonzalez et al., 2012]
PowerLyra	2015	C++	[Chen et al., 2015]
Spark (*GraphX*)	2013	Scala (Java)	[apa, c, Xin et al., 2013]
Flink (*Gelly*)	2015	Java (Scala)	[apa, a, apa, d]

Table: Présentation des plate-formes de traitement de graphes et de données

Résultats initiaux

Étude théorique des logiciels de gestion de grands graphes :

Plate-forme	Calcul	Partitionnement	Optimisation
Pregel	BSP	Edge-cut	-
Giraph	BSP	Edge-cut	-
Giraph++	BSP	Edge-cut	paradigme *Think Like a Graph*
GiraphuC	BAP	Edge-cut	-
Mizan	BSP	Edge-cut	migration dynamique
GPS	BSP	Edge-cut	LALP, migration dynamique
GraphLab	GAS	Edge-cut	-
PowerGraph	GAS	Vertex-cut	-
PowerLyra	GAS	Hybrid-cut	-
Spark (*GraphX*)	GAS	Vertex-cut	-
Flink (*Gelly*)	GAS	Vertex-cut	-

Table: Présentation des plate-formes de traitement de graphes et de données

Résultats initiaux

Caractérisation du graphe :

$\|V\|$	$\|E\|$	deg_{moy}	deg_{min}	deg_{max}
3 418 639	14 054 812	4,11	0	867 / 542
$indeg_{min}$	$indeg_{max}$	$outdeg_{min}$	$outdeg_{max}$	densité
0	472 / 310	0	395 / 327	$1{,}20 \times 10^{-6}$
triangle	Coef Cluest Local	Coef Cluest Global	diametre	$\|CC\|$
33 548 320	0.12747	0.xxx	21	1 375 918
$\|CFC\|$	$\|V\| \in$ maxCC	$\|E\| \in$ maxCC	$\|V\| \in$ maxCFC	$\|E\| \in$ maxCFC
1 543 394	1 504 249	12 757 724	1 295 601	11 501 267
$\|MCE\|$	$\|MCE\| > 2$	$\|MCE\| > 3$	$\|MCP\|$	$\|\|MCP\| \in MCE\|$
3 521 691	1 767 728	1 077 404	38	18

Table: Caractéristiques du graphe

Caractéristiques du ou de petit(s) monde(s)

Ouverture

A terme :

- Améliorer la qualité de réponse des requettes sur le graphe
- Proposer un enrichissement du graphe

Les références I

Apache flink.
https://flink.apache.org/.

Apache giraph.
http://giraph.apache.org/.

Apache spark.
http://spark.apache.org/.

Gelly, a graph module for apache flink.
https://ci.apache.org/projects/flink/flink-docs-master/apis/batch/libs/gelly.html.

Chen, R., Shi, J., Chen, Y., and Chen, H. (2015).
Powerlyra: Differentiated graph computation and partitioning on skewed graphs.
In *Proceedings of the Tenth European Conference on Computer Systems*, page 1. ACM.

Gonzalez, J. E., Low, Y., Gu, H., Bickson, D., and Guestrin, C. (2012).
Powergraph: Distributed graph-parallel computation on natural graphs.
In *Presented as part of the 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 12)*, pages 17–30.

Han, M. and Daudjee, K. (2015).
Giraph unchained: barrierless asynchronous parallel execution in pregel-like graph processing systems.
Proceedings of the VLDB Endowment, 8(9):950–961.

Les références II

Khayyat, Z., Awara, K., Alonazi, A., Jamjoom, H., Williams, D., and Kalnis, P. (2013).
Mizan: a system for dynamic load balancing in large-scale graph processing.
In *Proceedings of the 8th ACM European Conference on Computer Systems*, pages 169–182. ACM.

Low, Y., Bickson, D., Gonzalez, J., Guestrin, C., Kyrola, A., and Hellerstein, J. M. (2012).
Distributed graphlab: a framework for machine learning and data mining in the cloud.
Proceedings of the VLDB Endowment, 5(8):716–727.

Malewicz, G., Austern, M. H., Bik, A. J., Dehnert, J. C., Horn, I., Leiser, N., and Czajkowski, G. (2010).
Pregel: a system for large-scale graph processing.
In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 135–146. ACM.

Qi, Z., Xiao, Y., Shao, B., and Wang, H. (2013).
Toward a distance oracle for billion-node graphs.
Proceedings of the VLDB Endowment, 7(1):61–72.

Salihoglu, S. and Widom, J. (2013).
Gps: A graph processing system.
In *Proceedings of the 25th International Conference on Scientific and Statistical Database Management*, page 22. ACM.

Svendsen, M., Mukherjee, A. P., and Tirthapura, S. (2015).
Mining maximal cliques from a large graph using mapreduce: Tackling highly uneven subproblem sizes.
Journal of Parallel and Distributed Computing, 79:104–114.

Les références III


Tian, Y., Balmin, A., Corsten, S. A., Tatikonda, S., and McPherson, J. (2013).
From think like a vertex to think like a graph.
Proceedings of the VLDB Endowment, 7(3):193–204.


Xin, R. S., Gonzalez, J. E., Franklin, M. J., and Stoica, I. (2013).
Graphx: A resilient distributed graph system on spark.
In *First International Workshop on Graph Data Management Experiences and Systems*, page 2. ACM.


Yang, S., Wu, Y., Sun, H., and Yan, X. (2014).
Schemaless and structureless graph querying.
Proceedings of the VLDB Endowment, 7(7):565–576.

Merci