UNIVERSITE DE YAOUNDE I
Faculté des Sciences
Département d'Informatique
BP 812 Yaoundé-Cameroun

UNIVERSITY OF YAOUNDE I
Faculty of Science
Department of Computer Science
P.O. Box 812 Yaounde-Cameroon

Langage Dédié pour la Fouille des Réseaux Sociaux sur des Plates-formes Multi-cœurs

Mémoire en vue de l'obtention du

Diplôme de Master en Informatique

présenté et soutenu par

MESSI NGUELE Thomas ***07U165***

Sous la co-direction de :

Pr. Jean-François MEHAUT
UJF/CEA

Pr. Maurice TCHUENTE
UY1/LIRIMA-UMMISCO

Yaoundé, 2013

Réseau social

- ensemble d'entités (individus, organisations,...) interconnectées par des liens (amitié, échanges de messages,...).
- modélisation à l'aide des **graphes**
  - **très grand nombre de noeuds**, facebook 1.15 milliard, YouTube 1 milliard, Twitter 200 millions
  - **faible densité de liens**, un noeud est lié en moyenne à 150 autres noeuds.
- Exemple : le réseau des chercheurs de physique quantique (Quant-ph) [4] a 1060 nœuds et 1044 arêtes ;
  - arêtes possibles : **561270**, soit une densité de **0.2%** ;
  - katz [6] ou rules [5] tourne sur **560226** soit **99.8%**.
  - **Plusieurs heures d'exécution en mode séquentiel**.

Comment réduire le temps d'exécution ?

- Usage des architectures multi/many-cœurs.
- Mais, difficultés pour écrire des programmes parallèles efficaces et tirant profit de la multitude des cœurs disponibles [1, 2].
  - Écriture du code de bas niveau spécifique à la plate-forme.

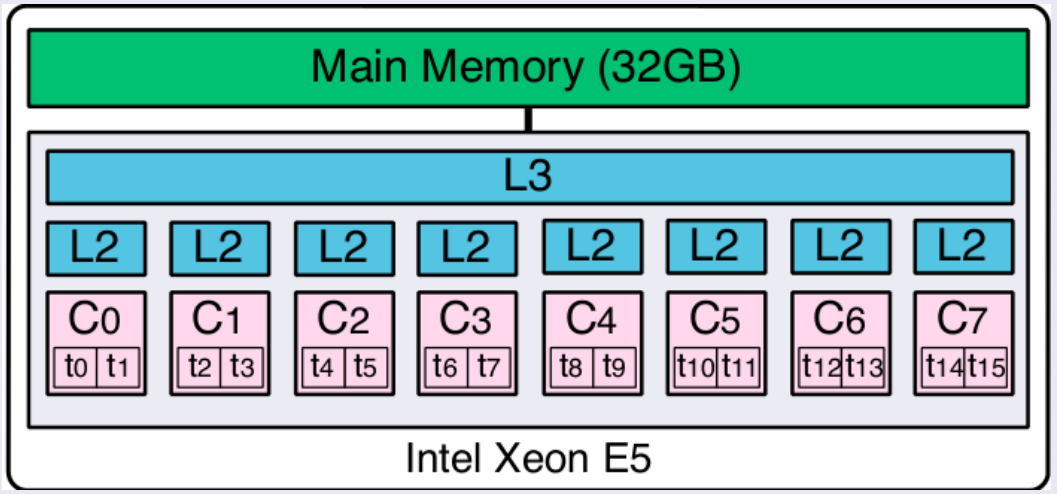
Problème

- Comment permettre aux chercheurs (algorithmiciens) de la fouille des réseaux d'écrire **facilement** et **efficacement** des **programmes parallèles** s'exécutant sur des multi/many-cœurs ?
- Deux étapes proposées :
  1. Étude des possibilités des architectures multi/many-coeurs.
  2. Conception et réalisation d'un langage dédié (*Domain Specific Langage*, **DSL**).

Plan

Xeon E5 d'Intel

Vue simplifiée du Xeon E5

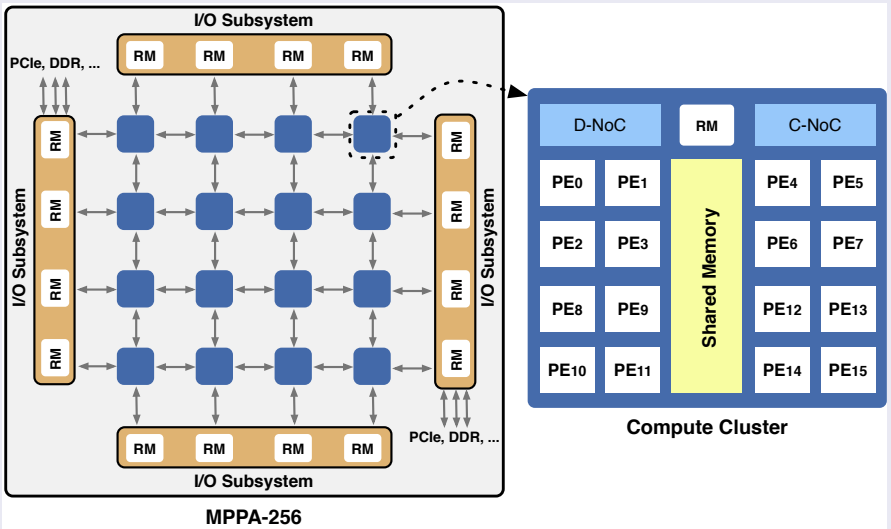


Caractéristiques

- processeur généraliste
- 8 coeurs, supportant l'hyperthreading ;
- 3 niveaux de cache
- 2.4GHZ
- 68.6 watt

MPPA-256 de Kalray

Vue simplifiée du MPPA-256

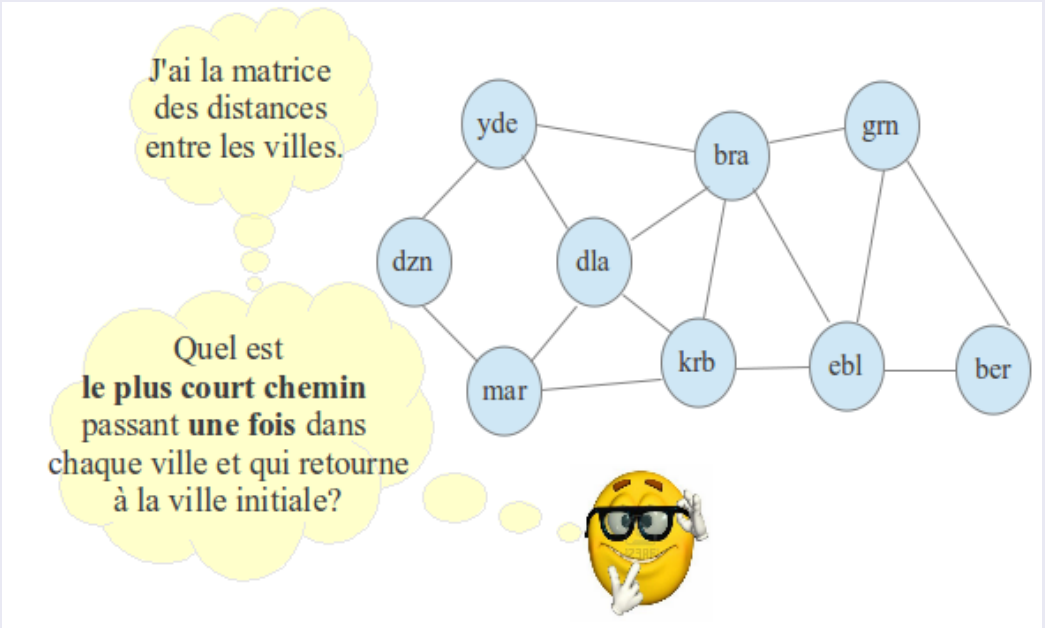


Caractéristiques

- 16 clusters de calcul
- 16 coeurs par cluster, soit en tout 256 coeurs
- 2 Mo de RAM par cluster
- 400 MHZ
- 10.7 watt

Le TSP

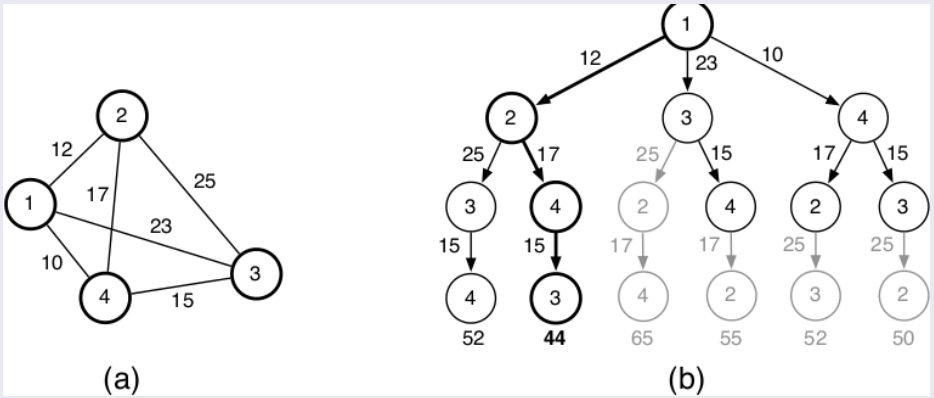
Problème du voyageur de commerce



À propos du TSP

- L'itinéraire le plus court
- Problème NP-complet
- Influencé par le dégré de parallélisme

Mét. de séparation et d'évaluation



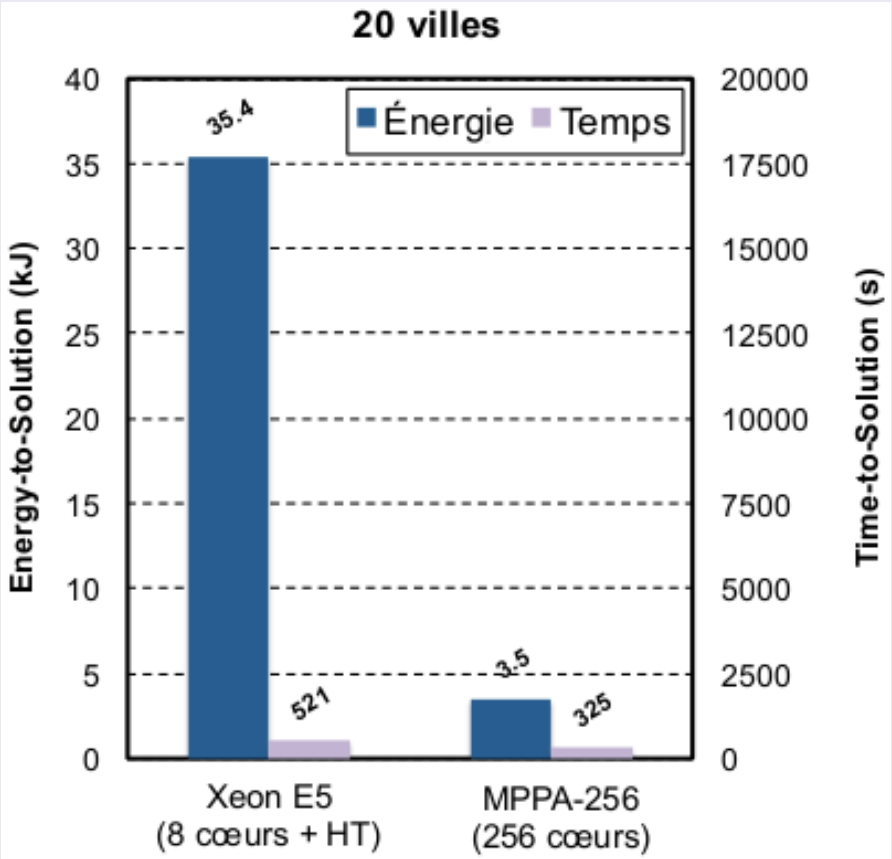
- Chemins à explorer plus courts que le meilleur chemin déjà rencontré,
- complexité dans le pire des cas $O(n!)$.

1. Algorithme séquentiel : mise à jour séquentielle du chemin minimal.
2. Algorithme multi-threadé : présence d'une file de tâches, mise à jour parallèle.
3. Algorithme distribué : mise à jour diffusée dans tous les postes de calcul.

Adaptation sur MPPA-256

- Solution multi-threadée pour les deux plates-formes, *min_path* comme variable partagée.
- Inappropriée pour le MPPA-256, caches non cohérents.
- Correction avec les inscructions spécifiques à la plate-forme.

MPPA-256 vs Xeon E5



Observations

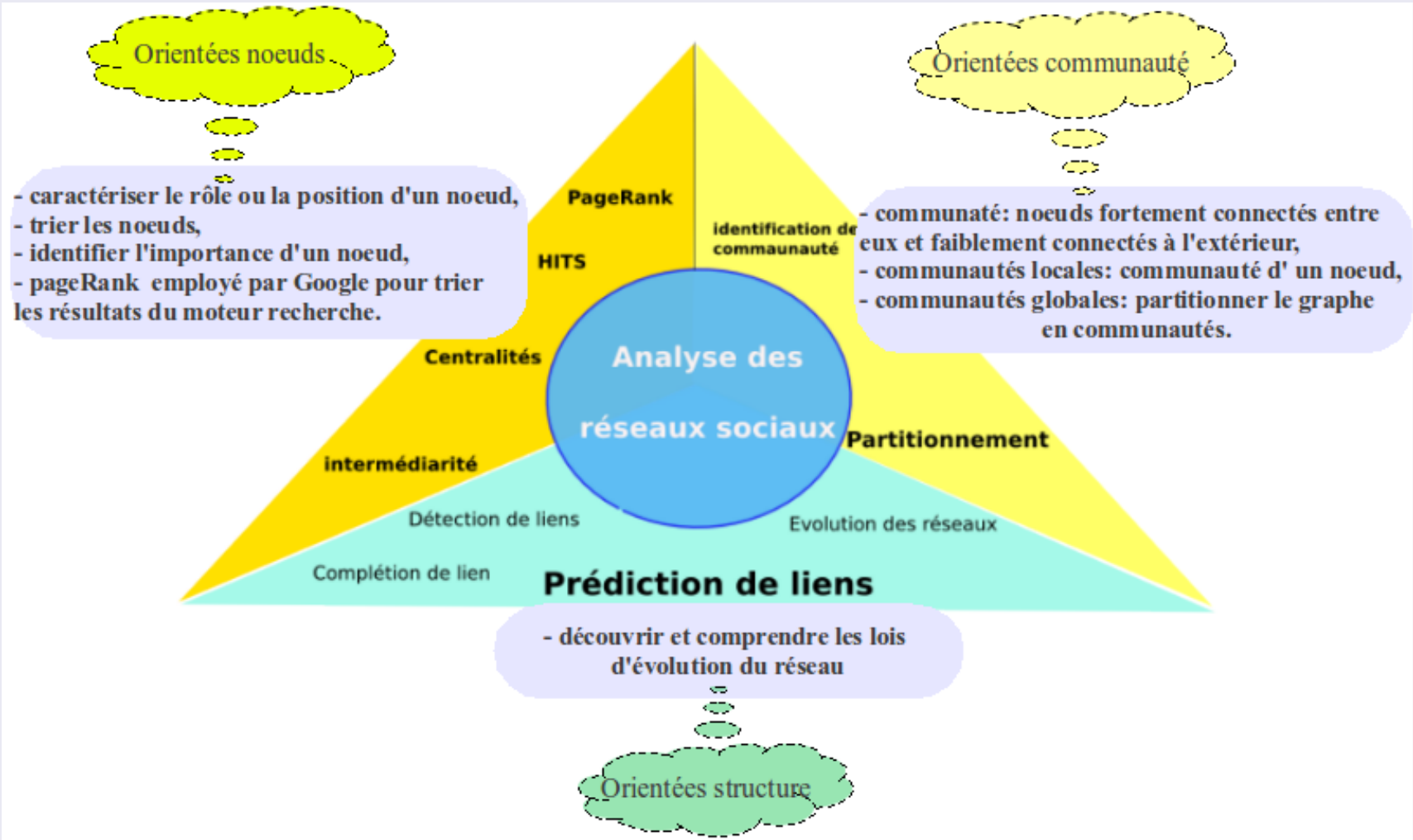
- le TSP s'exécute 1,6x plus vite sur MPPA-256
- MPPA-256 réduit la conso. de Xeon E5 de 90%

Expli. : mise en oeuvre du TSP

- peu d'échanges de messages entre les postes,
- diffusion implémentée avec les messages asynchrones à travers le NOC,
- donc absence de contention.

Many-cœurs comme MPPA-256 **donnent de très bons résultats**, mais nécessitent beaucoup d'efforts de la part du programmeur.

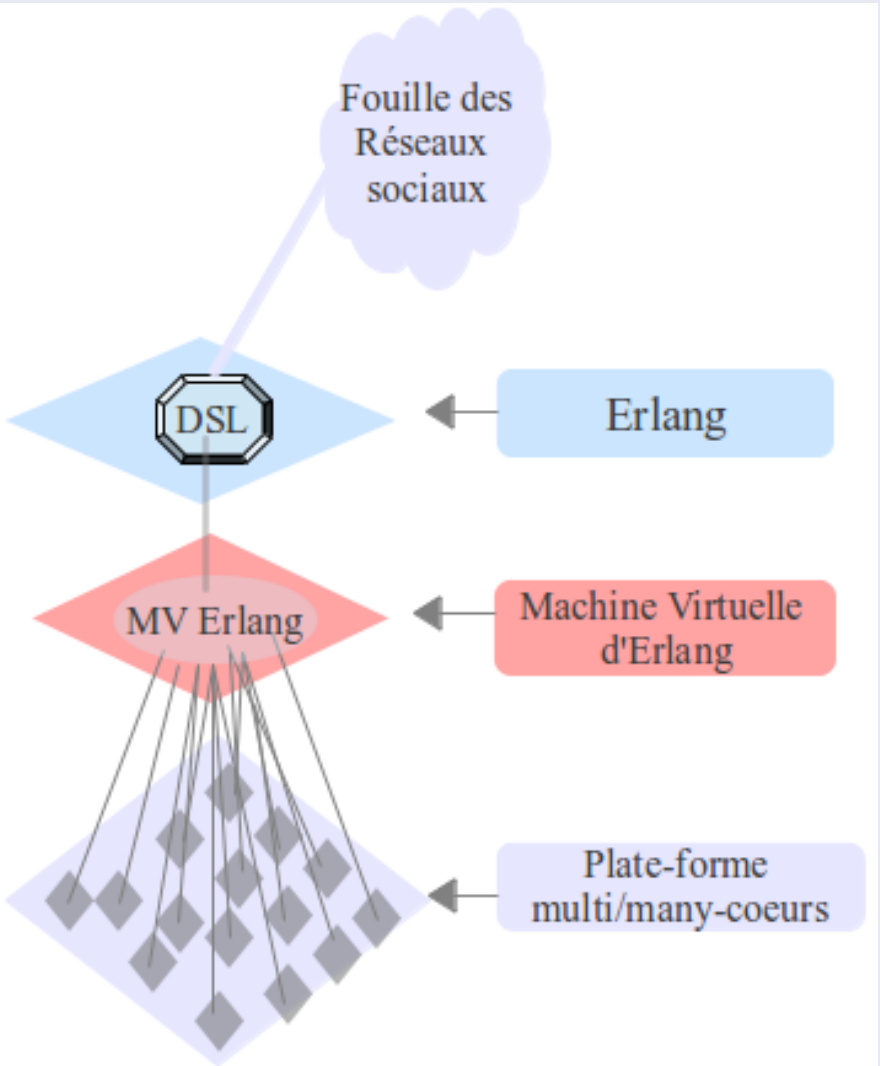
Tâches dans la fouille des réseaux sociaux



Généralités sur les DSL

- Langage de programmation conçu pour un domaine précis
- Exemples : HTML, SQL, YACC, ...
- Deux approches d'implémentation [3] :
  1. DSL externe ou "stand alone" ;
  2. DSL interne ou embarqué.
- Ici, l'approche embarquée avec Erlang comme langage hôte.

Présentation du DSL

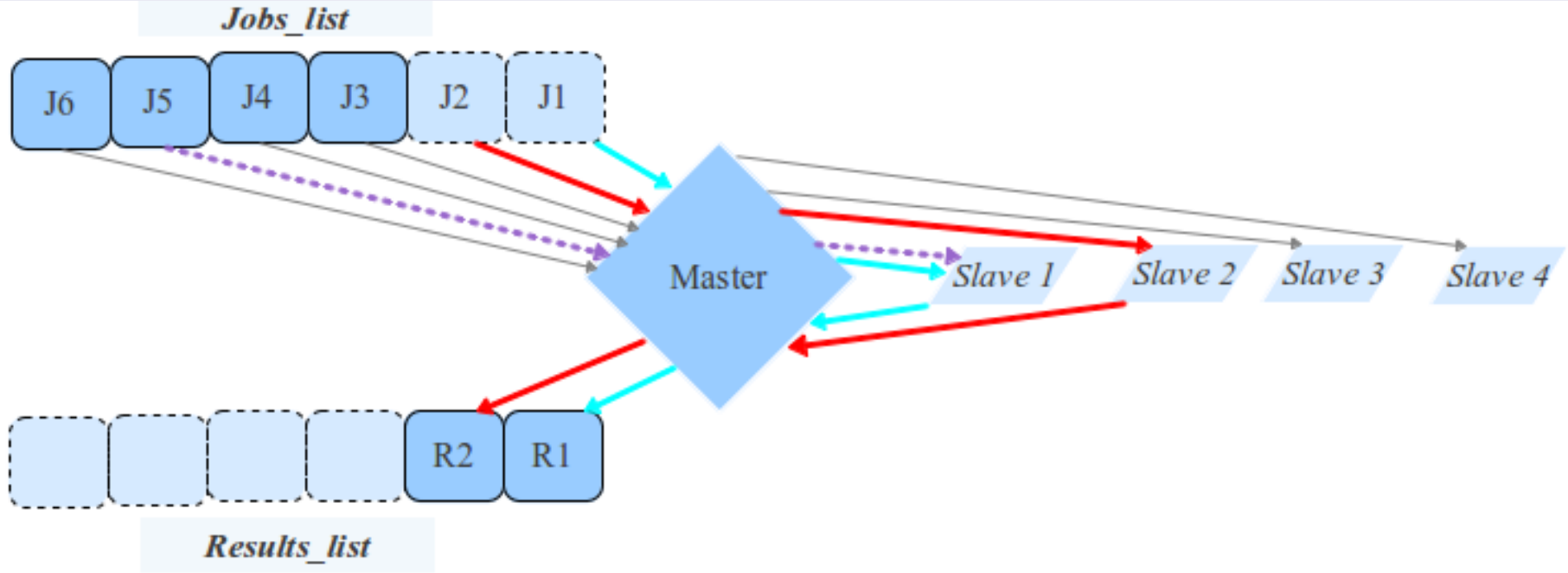


* DSL est embarqué dans Erlang
* faciliter la programmation parallèle sur des multi/many-cœurs,

- types de base d'Erlang restent utilisés,
- 3 principaux types : graphe, arête et noeud (à partir de digraph),

- Plusieurs primitives utilisées dans la fouille des réseaux
- création d'objets graphe, manipulation (nœuds, arêtes), ...
- DSL Green-Marl, les travaux de l'equipe (E. Viennet, B. Kaledjé, V. Kamga, M. Tchuente)

Parallélisme dans le DSL



- parallel_op permet d'écrire facilement des programmes parallèles,
- le maître a une liste de job et une liste des resultats,
- le maitre attribue des jobs aux esclaves et range les resultats des esclaves,
- approche implémentée dans [**?**].

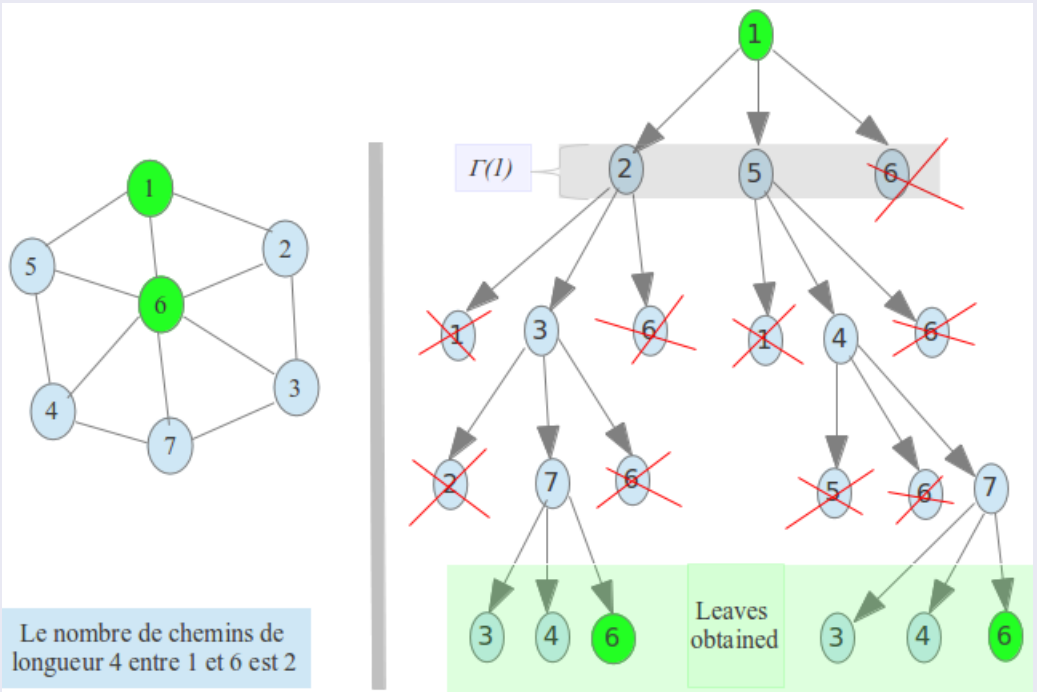
Le score de Katz [6, 7]

- Mesure de similarité basée sur les distances entre les nœuds.
- $katz\_score(x, y) = \sum_{l=1}^{\infty}(\beta^l.|paths_{x,y}^{<l>}|)$

Algorithme

1: **katz_score**(x, y, β, max_l, G)
2: $ktz\_score \leftarrow 0.0$
3: **for** $l = 1$ to max_l **do**
4: $\quad leaves\_list \leftarrow$ **paths_leaves**$(x, y, l, 0, G)$
5: $\quad nb\_path\_l \leftarrow occurence\_count(y, leaves\_list)$
6: $\quad ktz\_score \leftarrow ktz\_score + \beta^l * nb\_path\_l$
7: **end for**
8: **return** ktz_score

Calcul du nombre de chemins



Commentaires

- x=1, y=6, l=4.
- Arbre de racine x et de hauteur l.
- Nombre de y en feuilles : 2.

Implémentation avec le DSL

```
compute_katz_score(Filename, NbProcess, Beta, Max_l) ->
  {G, Node_list, Edge_list,Directed} = load_graph_gml(Filename),
  Min_node = lists:min(Node_list),
  Max_node = lists:max(Node_list),
  All_possible_edge = generate_all_possible_edges(Min_node, Max_node),
  Candidate = build_candidate_edge_list(All_possible_edge, Edge_list, Directed, []),
  parallel_op(dsl_graph,sequential_katz_score,[Beta, Max_l, G, Directed],NbProcess,Candidate).
```

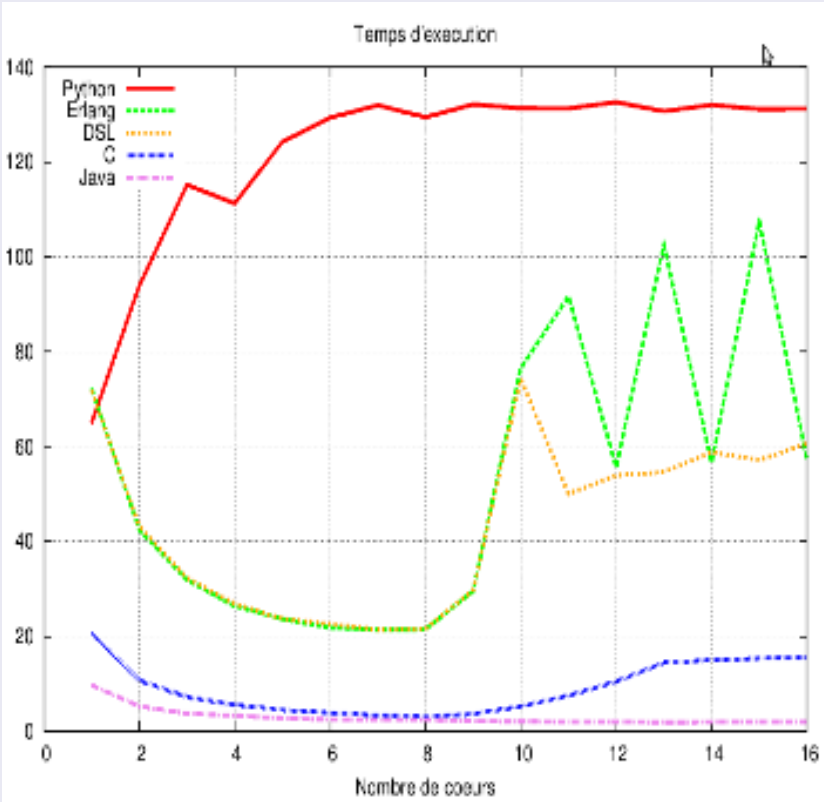
- Moins d'effort à fournir.

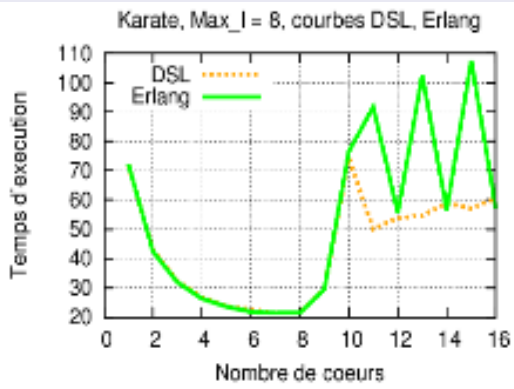
Nombre de lignes de code

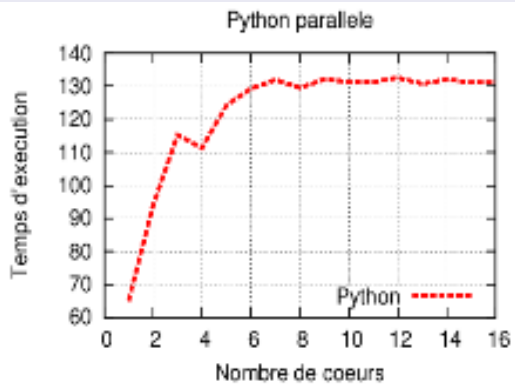
Langages	C	Java	Python	Erlang	DSL
Nombre de ligne de code	944	462	284	274	30

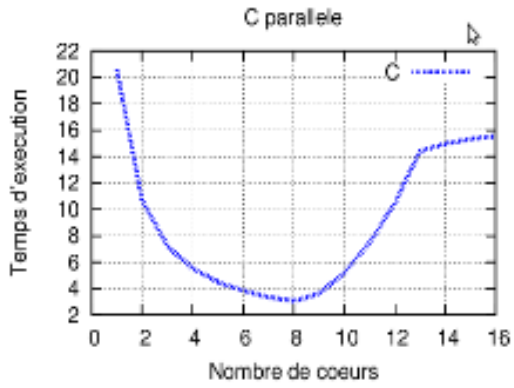
- Le DSL a **moins de lignes de code** que les autres langages.

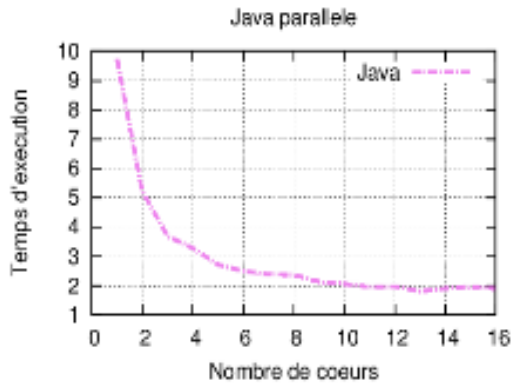
Évolution du temps d'exécution sur Numa32





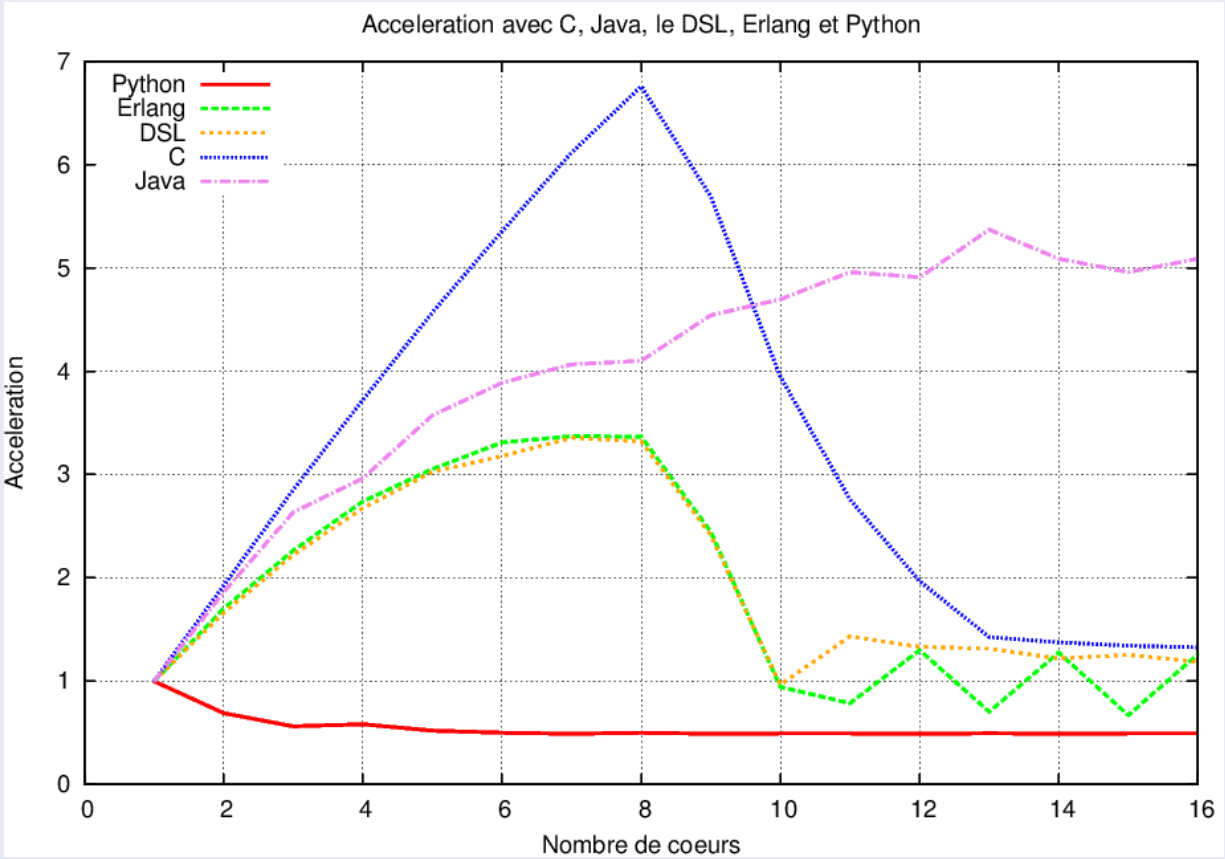






- Diminution du temps d'exécution en fonction du nombre de cœurs.
- À partir de 9 cœurs, le temps d'exécion augmente.

Gains de parallélisme



- Accélération croissante jusqu'à 8 cœurs.
- Même accélération entre DSL et Erlang jusqu'à 8 cœurs

Conclusion

- Objectif : **faciliter** la **programmation parallèle** des algorithmes de **fouille des réseaux sociaux** sur des plates-formes multi/many-coeurs.
- Contributions :
  1. Étude des possibilités des architectures multi/many-coeurs.
     - Très bonnes performances,
     - mais nécessitent beaucoup d'efforts.
  2. DSL pour la fouille des réseaux sociaux.
     - Facilité assurée,
     - performance limitée par le langage hôte.

Perspectives

- DSL version stand-alone
- Gestion fine de l'allocation des threads
- Inclure d'autres applications et d'autres architectures dans l'étude des défis de programmation.

Merci de votre aimable attention!


Marcio Castro, Emilio Francesquin, Thomas Messi Nguélé, and Jean-François Méhaut.
Analysis of computing and energy performance of multicore, numa, and manycore platforms for an irregular application.
In *Proceedings of the 3rd Workshop on Irregular Applications : Architectures and Algorithms*, page 5. ACM, 2013.


Marcio Castro, Emilio Francesquin, Thomas Messi Nguélé, and Jean-François Méhaut.
Multicœurs et manycœurs : Une analyse de la performance et l'Éfficacité Énergétique d'une application irrégulière.
Proceedings of the CRI'2013, Conférence de Recherche en Informatique, decembre 2013.
Accepted.


Arie Van Deursen, Paul Klint, and Joost Visser.
Domain-specific languages : An annotated bibliography.
ACM SIGPLAN NOTICES, 35 :26–36, 2000.


http ://arxiv.org.

Quant-ph.


Vanessa Kamga, Maurice Tchuente, and Emmanuel Viennet.
Prévision des liens dans les graphes bipartites avec attributs.
Mémoire de master 2 recherche, Université de Yaoundé 1, Département d'Informatique, Septembre 2013.


Leo Katz.
A new status index derived from sociometric analysis.
Psychometrica-vol.18, No.1, 18(1), March 1953.


David Liben-Nowell and Jon Kleinberg.
The link prediction problem for social networks.
January 8 2004.