大規模グラフを効率的に処理するための基盤技術

Palgol:頂点主体並列グラフ処理のための領域特化言語

Palgol: a high-level domain-specific language for vertex-centric graph processing

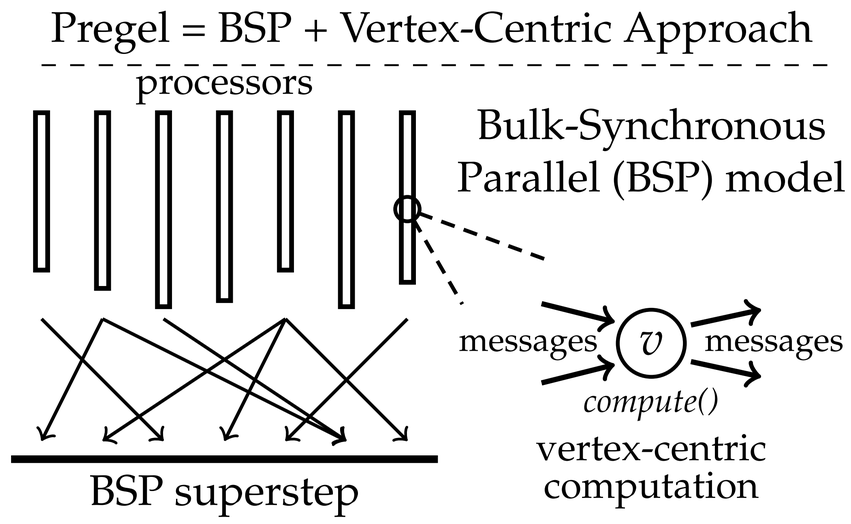
章 雍哲 (Yongzhe ZHANG)　柯 向上 (Hsiang-Shang KO)　胡 振江 (Zhenjiang HU)

頂点主体並列グラフ処理とは?

Graph algorithms are represented as the iterative execution of the same piece of code on every vertex.

代表的なソフトウェア

Pregel: A synchronous vertex-centric graph processing framework.



メリット

- high scalability
- universality

デメリット

- message-passing is hard
- vertices have to maintain complicated states

解決方法

領域特化言語：メッセージ無しで頂点主体並列プログラムを構成することをできるように

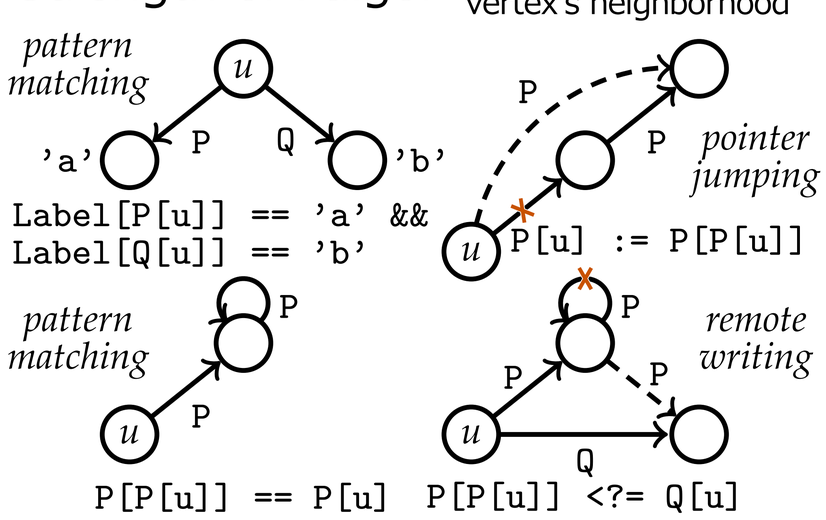
Palgol: a domain-specific language with remote access capabilities

- Chain access (new feature)
- Remote writes (enhanced feature)
- Neighborhood aggregation

```
let t = [ P[e.ref] | e <- Nbr[u] ]
```

Extending Green-Marl and Fregel's declarative programming models

Strength of Palgol



Example: pointer-jumping

```
// P[u]: store u's parent id
do
  // enter vertex-centric mode
  for u in V
    // P[P[u]] -> chain access
    if (P[P[u]] != P[u])
      P[u] := P[P[u]]
  end
until fix[P] // fixed-point
```

Palgol = iteration construct + vertex-centric model + remote access

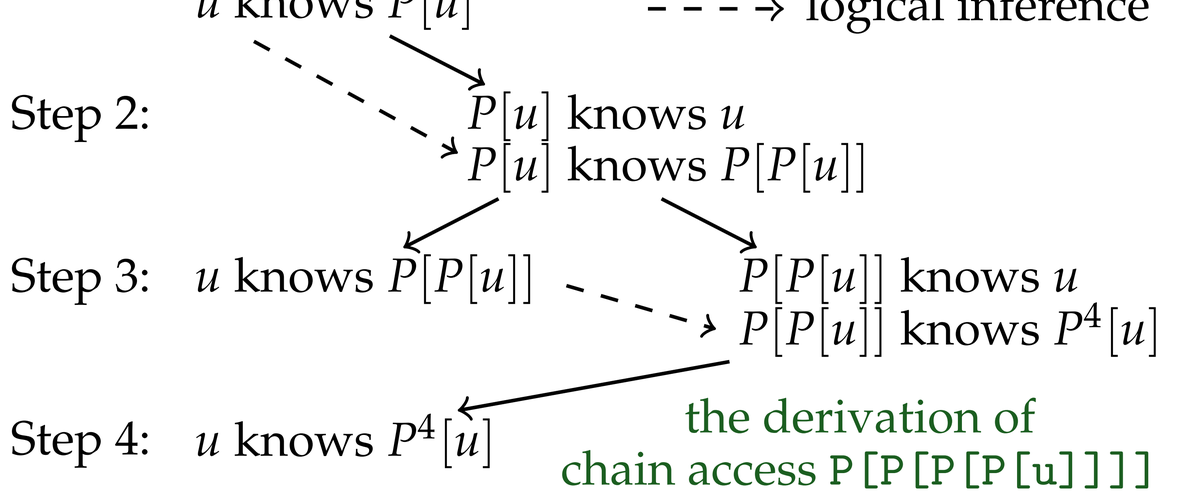
研究内容

Palgolで構成したプログラムをPregelへ変換する

Key technique: compiling general chain access

Logic system for handling `P[P[P[P[u]]]]`:

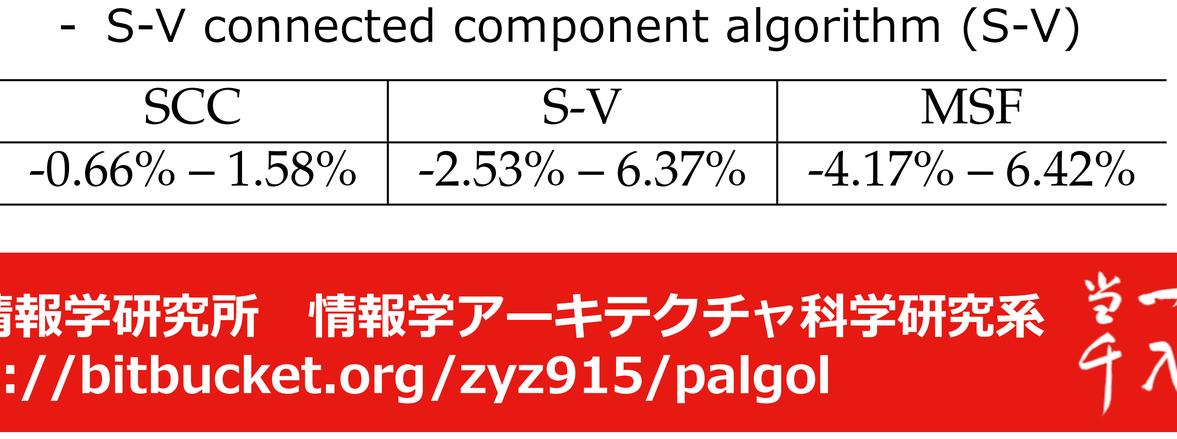
1. $\forall u.\, \mathrm{K}_u\, u$　every vertex u knows its own identifier u
2. $\forall u.\, \mathrm{K}_u\, P[u]$　every vertex u knows its local field $P[u]$
3. $(\forall u.\, \mathrm{K}_{w(u)}\, e(u)) \wedge (\forall u.\, \mathrm{K}_{w(u)}\, v(u)) \Rightarrow \forall u.\, \mathrm{K}_{v(u)}\, e(u)$
   we can make every $v(u)$ know $e(u)$ by letting every $w(u)$ send $e(u)$ to $v(u)$



A backtracking algorithm with memoization for finding a solution with minimum number of steps

Application: the pointer-jumping technique
(a useful graph transformation used in S-V, MSF and so on)

Given a forest where each vertex initially knows its parent, how to let every vertex point to its root?



all vertices constantly find their "grandparents"

Performance Evaluation

We compare Palgol with hand-coded Pregel programs on real-world graphs

- Minimum spanning forest (MSF)
- Strongly connected component (SCC)
- S-V connected component algorithm (S-V)

SCC	S-V	MSF
-0.66% – 1.58%	-2.53% – 6.37%	-4.17% – 6.42%

NII
連絡先：胡 振江 (Zhenjiang HU)　国立情報学研究所　情報学アーキテクチャ科学研究系
Email：hu@nii.ac.jp
https://bitbucket.org/zyz915/palgol