

Pregel: Big(ger) Graph Analytics on A Dataflow Engine [R17-1]

- ▶ Y. Bu, V. Borkar, J. Jia, M.J. Carey, T. Condie (UC Irvine ほか)
- ▶ 大規模グラフの解析: MapReduceは不向き
- ▶ **Pregel (Google)**: グラフ解析のためのフレームワーク
 - ▶ Giraph: オープンソース版
 - ▶ **“Think like a vertex”** のアプローチ
 - ▶ ユーザは, グラフの頂点(ノード)の立場でユーザ定義関数を記述
 - ▶ システム側が並列に実行
 - ▶ プロセス指向の実装: 各プロセスに処理を割り当て
 - ▶ メモリ上にないデータへの対応が不十分
 - ▶ DB分野における並列処理技術を生かしていない

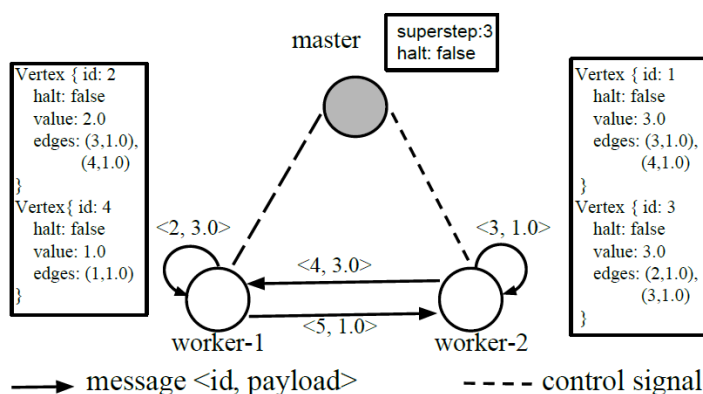
Some figures are incorporated from the original paper.

Pregelの概要(1)

目的

- ▶ **集合指向**で**反復**に対応する**データフロー**アプローチ
- ▶ 実装: 論理問合せプランを構築し, リレーショナル演算子 (join および group-by) のデータフローの繰り返しとして実行
- ▶ インメモリおよびアウトオブコア (out-of-core) の両方のタスクを効率的に実行可能
- ▶ アプローチ: Pregelの状態を入れ子リレーションで表現

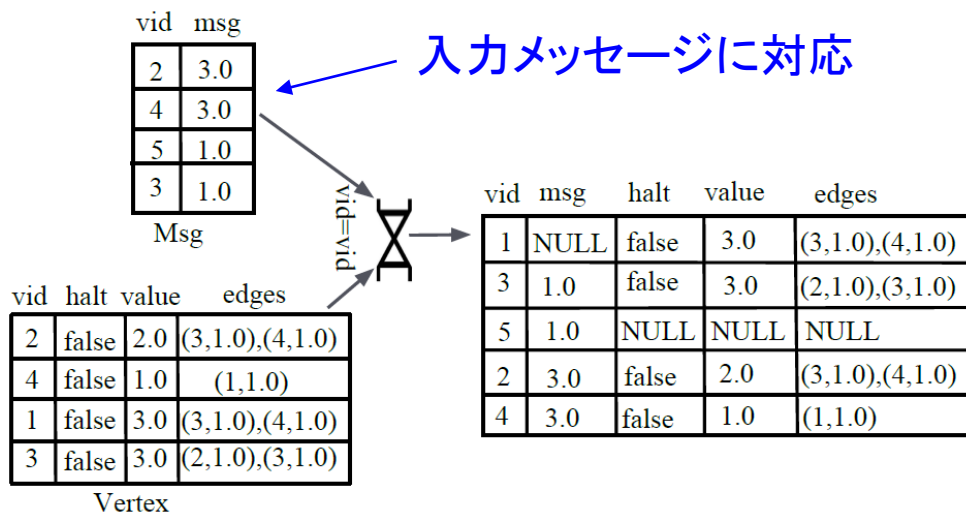
Pregelの プロセス 構造



リレーション	スキーマ
Vertex	(vid, halt, value, edges)
Msg	(vid, payload)
GS	(halt, aggregate, superstep)

Pregelixの概要(2)

▶ 論理プラン(例): メッセージパッシングを結合で表現



外部結合 (outer join) で表現

- ・左外部結合: 新たな頂点へのメッセージ送信
- ・右外部結合: メッセージ送信がない場合に相当

▶ ユーザ定義関数 (UDF)

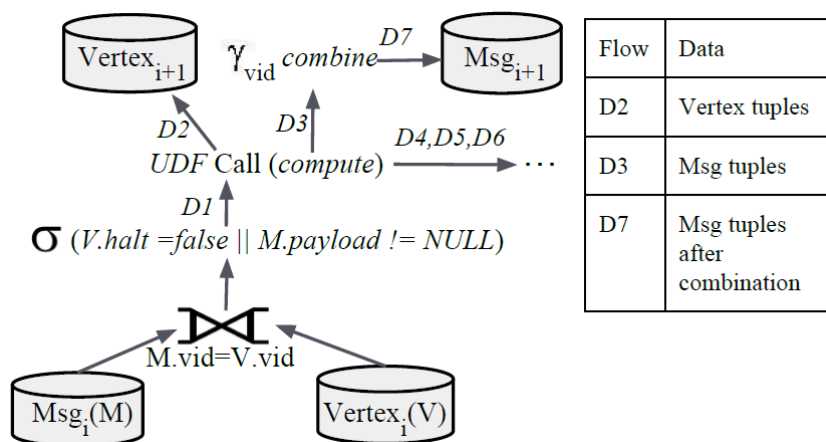
▶ プログラムの振舞いを決定

UDF	記述
compute	各スーパステップにおいて、各アクティブノードで実行される
combine	メッセージを集約する関数
aggregate	大域的な状態を集約する関数
resolve	グラフポロジの変更に対応する処理

Pregelixの概要(3)

▶ 論理プランの一部を切り出した例

- ▶ 結合の後, アクティブノードに対しcomputeを呼び出す
- ▶ 次の繰り返しの頂点集合とメッセージ集合 (combineを使用) を構築

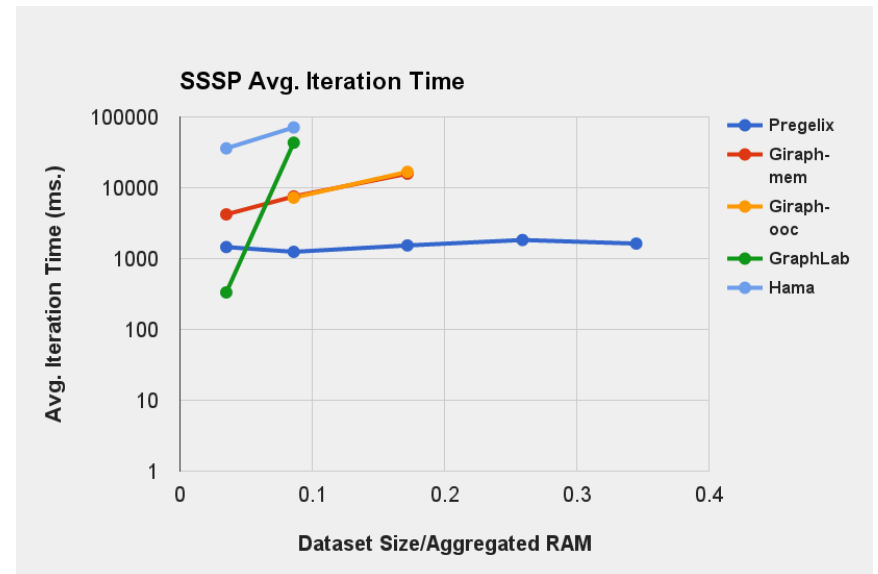
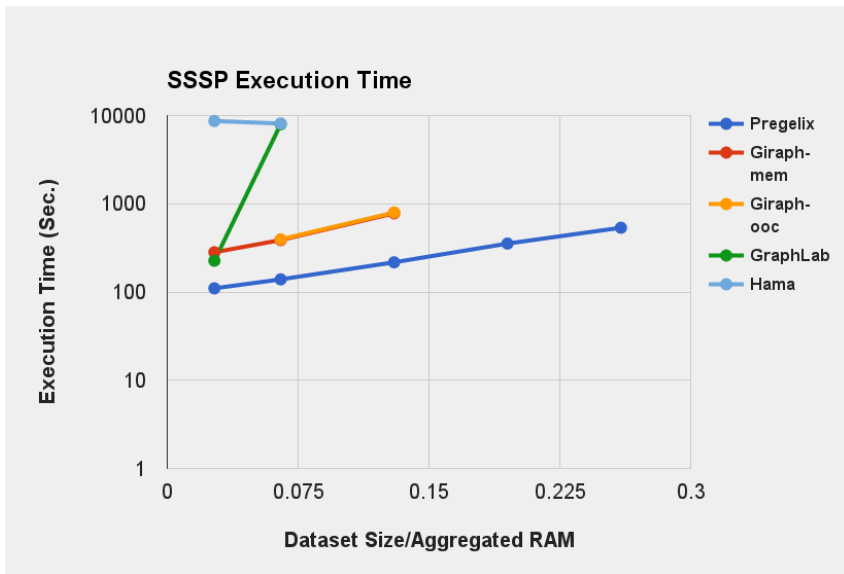


▶ 問合せ実行: [Hyracks](#)上に実装

- ▶ 並列の結合処理, 複数の問合せプランを考慮した最適化

性能評価

- ▶ 出発点が一つの最短路問題 (Single Source Shortest Path, SSSP) での評価例：実行時間および反復回数



- ▶ アウトオブコアなタスクに対応
- ▶ メッセージが疎なタスク (SSSPなど) ではGiraphより高速. メッセージが密なタスク (PageRankなど) では同等程度