HDRF: Stream-Based Partitioning for Power-Law Graphs.

Fabio Petroni, L. Querzoni, K. Daudjee, S. Kamali, G. Iacoboni
Graphs

- large amount of real data can be represented as a graph.

- the problem of optimally partitioning a graph while maintaining load balance is important in several contexts.
Distributed Graph Computing Frameworks

- support efficient computation on really large graphs.
  - graph analytics; data mining and machine learning tasks.

- input data partitioning can have a significant impact on the performance of the graph computation.
  - affects network usage, memory occupation, synchronization.
Balanced Graph Partitioning

- partition $G$ into smaller components of (ideally) equal size

- **edge-cut**: vertex disjoint partitions
- **vertex-cut**: edge disjoint partitions

- a vertex can be cut in multiple ways and span several partitions while a cut edge connects only two partitions.
Balanced Graph Partitioning

- partition $G$ into smaller components of (ideally) equal size

- a vertex can be cut in multiple ways and span several partitions while a cut edge connects only two partitions.
Vertex-Cut

- Modern distributed graph computing frameworks use vertex-cut.

Computation steps are associated with edges

\( v \)-cut perform better on power law graphs

\( \text{(Gonzalez et al., 2012)} \)

Create less storage and network overhead
Power-Law Graphs

- characteristic of real graphs: power-law degree distribution.
  - most vertices have few connections while a few have many.

- the probability that a vertex has degree \( d \) is \( P(d) \propto d^{-\alpha} \).
- \( \alpha \) controls the “skewness” of the degree distribution.
Balanced Vertex-Cut Graph Partitioning

- \( v \in V \) vertex; \( e \in E \) edge; \( p \in P \) partition.
- \( A(v) \) set of partitions where vertex \( v \) is replicated.
- \( \sigma \geq 1 \) tolerance to load imbalance.
- the size \( |p| \) of partition \( p \) is its edge cardinality.

\[
\min \frac{1}{|V|} \sum_{v \in V} |A(v)| \quad \text{s.t.} \quad \max_{p \in P} |p| < \sigma \frac{|E|}{|P|}
\]

- the objective function is the replication factor (RF).
  - average number of replicas per vertex.

**minimize replicas**
reduce (1) bandwidth, (2) memory usage and (3) synchronization

**balance the load**
efficient usage of available computing resources
Streaming Setting

- input data is a list of edges, consumed in *streaming fashion*, requiring only a **single pass**.

- handle graphs that don’t fit in the main memory.
- impose minimum overhead in time.
- scalable, easy parallel implementations.
- assignment decision taken cannot be later changed.
Algorithms Taxonomy

- stream-based vertex-cut graph partitioning
  - history agnostic
    - hashing
      - Hashing (Gonzalez et al., 2012)
      - DBH (Xie et al., 2014)
    - constrained partitioning
      - Grid (Jain et al., 2013)
      - PDS (Jain et al., 2013)
  - history aware
    - Greedy (Gonzalez et al., 2012)
    - HDRF (Petroni et al., 2015)
I favor the replication of high-degree vertices. The number of high-degree vertices in power-law graphs is very low. Overall reduction of the replication factor.
in the context of robustness to network failure.
- if few high-degree vertices are removed from a power-law graph then it is turned into a set of isolated clusters.
- focus on the locality of low-degree vertices.
The HDRF Algorithm

incoming edge

vertex without replicas
vertex with replicas

case 1
vertices not assigned to partitions
The HDRF Algorithm

incoming edge

vertex without replicas

vertex with replicas

case 1
place e in the least loaded partition
The HDRF Algorithm

incoming edge

<table>
<thead>
<tr>
<th>vertex without replicas</th>
</tr>
</thead>
<tbody>
<tr>
<td>vertex with replicas</td>
</tr>
</tbody>
</table>

**case 1**

place e in the least loaded partition

**case 2**

only one vertex has been assigned
The HDRF Algorithm

**case 1**
place e in the least loaded partition

**case 2**
place e in the partition
The HDRF Algorithm

**case 1**
place e in the least loaded partition

**case 2**
place e in the partition

**case 3**
vertices assigned, common partition
The HDRF Algorithm

*incoming edge*

- **vertex without replicas**
- **vertex with replicas**

**case 1**
place e in the least loaded partition

**case 2**
place e in the partition

**case 3**
place e in the intersection
Create Replicas

\begin{figure}
\centering
\begin{tikzpicture}
\node (A) at (0,0) [draw, circle, fill=blue!20] {e};
\node (B) at (1,0) [draw, circle, fill=orange!20] {\textit{case 4}};
\draw (A) -- (B);
\end{tikzpicture}
\caption{empty intersection}
\end{figure}
Create Replicas

\[\text{case 4} \quad \text{empty intersection}\]

\[\text{standard Greedy solution}\]

\[\text{case 4} \quad \text{least loaded partition in the union}\]
Create Replicas

\[ e \]

**case 4**

empty intersection

**standard Greedy solution**

\[ e \]

**case 4**

least loaded partition in the union

\[ \delta(v_1) > \delta(v_2) \]

**HDRF**

\[ e \]

**case 4**

replicate vertex with highest degree
Equivalent Formulation: Maximize The Score

- computing a score $C(e, p)$ for all partitions $p \in P$.
- assigns $e$ to the partition that maximizes $C(e, p)$.

$$C(e, p) = C_{REP}(e, p) + C_{BAL}(p)$$

- the balance term breaks ties in replication term.
- this may not be enough to ensure load balance.
Equivalent Formulation: Maximize The Score

- computing a score $C(e, p)$ for all partitions $p \in P$.
- assigns $e$ to the partition that maximizes $C(e, p)$.

\[
C(e, p) = C_{REP}(e, p) + \lambda \cdot C_{BAL}(p)
\]

the balance term breaks ties in replication term.

this may not be enough to ensure load balance.

- $\lambda$ controls the importance of the balance term.
- balanced partitions even when classical greedy fails.
Ordered Stream Of Edges

- without $\lambda$
Ordered Stream Of Edges

- without $\lambda$
Ordered Stream Of Edges

- without $\lambda$
Ordered Stream Of Edges

- without $\lambda$
Ordered Stream Of Edges

- without \( \lambda \)

all the graph in a single partition
Ordered Stream Of Edges

- with $\lambda$
Experiments - Settings

- standalone partitioner.
  - VGP, a software package for one-pass vertex-cut balanced graph partitioning.
  - measure the performance: replication and balancing.

- GraphLab.
  - HDRF has been integrated in GraphLab PowerGraph 2.2.
  - measure the impact on the execution time of graph computation in a distributed graph computing frameworks.

- stream of edges in random order.
Experiments - Datasets

- real-world graphs.
- synthetic graphs.

| Dataset            | $|V|$  | $|E|$  |
|--------------------|------|------|
| MovieLens 10M      | 80.6K| 10M  |
| Netflix            | 497.9K| 100.4M|
| Tencent Weibo      | 1.4M | 140M |
| twitter-2010       | 41.7M| 1.47B|

$\alpha = 1.8$
$\alpha = 2.2$
$\alpha = 2.6$
$\alpha = 3.0$
$\alpha = 3.4$
$\alpha = 4.0$
Results - Synthetic Graphs Replication Factor

- 128 partitions

![Graph showing replication factor vs. alpha for different graph types and partition sizes](image-url)
Results - Synthetic Graphs Replication Factor

- 128 partitions

![Graph showing replication factor vs skewness and alpha]

- HDRF
- PDS $|P|=133$
- Grid $|P|=121$
- Greedy
- DBH

**Notes:**
- Less edges leads to a less dense graph, making it easier to partition.
- The graph shows how replication factor changes with skewness and alpha parameter.
Results - Synthetic Graphs Replication Factor

- 128 partitions

![Graph showing replication factor and partitioning methods]

- More edges, more dense → difficult to partition
- Less edges, less dense → easier to partition

- HDRF
- PDS $|P|=133$
- Grid $|P|=121$
- Greedy
- DBH

<table>
<thead>
<tr>
<th>skewing</th>
<th>density</th>
<th>partition difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td>skewed</td>
<td>more</td>
<td>difficult</td>
</tr>
<tr>
<td>homogenous</td>
<td>less</td>
<td>easier</td>
</tr>
</tbody>
</table>

Results - Synthetic Graphs Replication Factor

- 128 partitions

![Graph diagram showing replication factor vs. alpha for different graph types (skewed, homogeneous) and partitioning methods (HDRF, PDS, grid, greedy, DBH). The graph illustrates that skewed graphs with more edges and more dense are difficult to partition, while homogeneous graphs with less edges and less dense are easier to partition. The diagram also highlights the exploitation of power-law distributions.]
Results - Synthetic Graphs Replication Factor

- 128 partitions

![Graph showing replication factor vs. alpha for different distributions and replication strategies.](image-url)
Results - Real-Word Graphs Replication Factor

- 133 partitions

![Bar chart showing replication factor for different datasets and methods: PDS, DBH, greedy, HDRF. The datasets are Tencent Weibo, Netflix, MovieLens 10M, twitter-2010. The replication factors for each dataset and method are listed in the chart.]
Results - Real-Word Graphs Replication Factor

- 133 partitions

<table>
<thead>
<tr>
<th></th>
<th>Tencent Weibo</th>
<th>Netflix</th>
<th>MovieLens 10M</th>
<th>twitter-2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replication Factor</td>
<td>7.9</td>
<td>11.3</td>
<td>11.5</td>
<td>8.0</td>
</tr>
</tbody>
</table>

**HDRF** achieves a replication factor about **40%** smaller than **DBH**, more than **50%** smaller than **Greedy**, almost **3×** smaller than **PDS**, more than **4×** smaller than **Grid** and almost **14×** smaller than **Hashing**.
Results - Load Relative Standard Deviation

MovieLens 10M

- PDS
- grid
- greedy
- DBH
- HDRF
- hashing

Graph showing the load relative standard deviation (%) against partitions for different methods.
Results - Graph Algorithm Runtime Speedup

- SGD algorithm for collaborative filtering on Tencent Weibo.

The speedup is proportional to both:
  - the advantage in replication factor.
  - the actual network usage of the algorithm.
Conclusion

- HDRF is a one-pass vertex-cut graph partitioning algorithm.
- based on a greedy vertex-cut approach that leverages information on vertex degrees.
- we provide a theoretical analysis of HDRF with an average-case upper bound for the vertex replication factor.
- experimental study shows that:
  - HDRF provides the smallest replication factor with close to optimal load balance.
  - HDRF significantly reduces the time needed to perform computation on graphs.
- the stand-alone software package for one-pass v-cut balanced partitioning at https://github.com/fabiopetroni/VGP.
Thank you!

Questions?

Fabio Petroni
Sapienza University of Rome, Italy

Current position:
PhD Student in Engineering in Computer Science

Research Interests:
data mining, machine learning, big data

petroni@dis.uniroma1.it