Distributed Frameworks for Alternating Least Squares

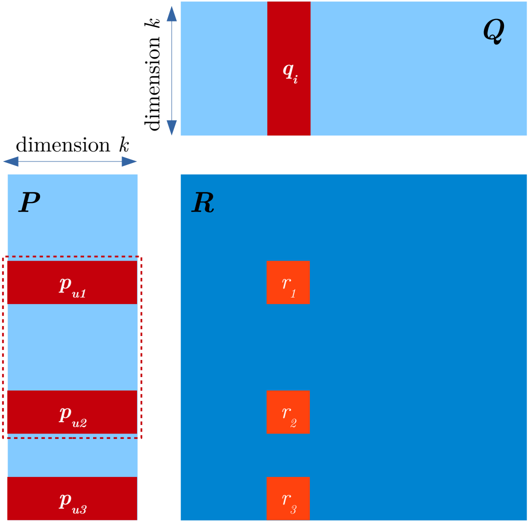
Márton Balassi **Róbert Pálovics** András A. Benczúr
{mbalassi, rpalovics, benczur}@ilab.sztaki.hu

Informatics Laboratory, Department of Computer and Automation Research Institute,
Hungarian Academy of Sciences

The publication was supported in part by the EC FET Open project "New tools and algorithms for directed network analysis" (NADINE No 288956)

8th ACM Conference on Recommender Systems
Foster City, Silicon Valley, USA, 6th-10th October 2014

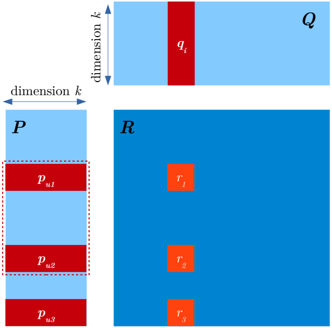
Alternating Least Squares



Alternating Least Squares

$$f_{RMSE}(P,Q) = \sum_{(u,i)\in\text{Training}} (R_{ui} - p_u \cdot q_i^T)^2 + \lambda \cdot (\|P\|_F^2 + \|Q\|_F^2)$$

- Update step for Q: $Q_i \leftarrow (P^TP)^{-1}P^TR_i$
- For each nonzero rating we communicate $(P^TP)^{-1}$ of dim k^2

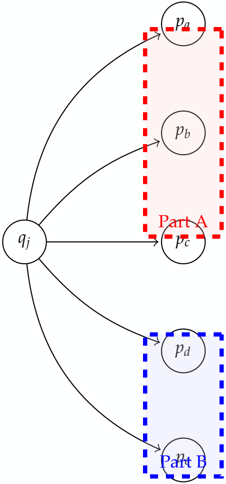


ALS MULTI-MACHINE NO SHARED MEMORY

- Goal: efficient ALS *and* models for other algorithms
- Problem: Large amount of communication alternating between rows and columns
  - ALS message size is quadratic in number of latent factors
- Drawback of "think as a node" philosophy
  - Repeat the same message for all graph nodes
  - Even if they reside on the same server

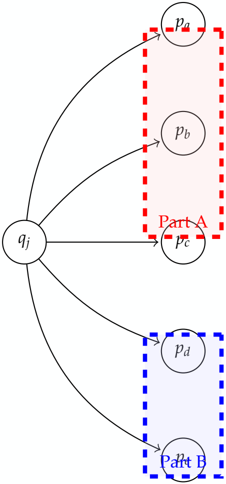
DISTRIBUTION OVERHEAD

- Partitioned graph or ratings matrix
- Naive approach: q_j communicates to each p_i individually
- In ALS, PageRank, $\ldots$, messages from q_j are identical
- **Network communication becomes the bottleneck.**



Proposed solution

- Efficient communication between partitions
- Translated to graph processing this is just a *multicast*.



Big Data Frameworks

- Big Data frameworks lack an operator for this job.
  - Hadoop (Mahout) Map, Reduce
  - Spark “Functional” operators on (memory) Resilient Distributed Datasets
  - Flink “Functional” operators and iteration
    Our experimental platform
- **Notion of the partition hidden from user when implementing ALS by vector-to-vector communication.**

Big Data Frameworks - Solution

- Mahout implementation: “CustomALS”.
- Algorithm provides an artificial partition ID
- Map-Reduce grouped by partition ID, expected one partition per reducer
- Partitioning to minimize the communication between partitions **not ensured** but left for the framework

Graph processing engines

- Bulk Synchronous Parallel (BSP)
  - Sends along ALL nonzero ratings
  - Even if the message is identical
  - This issue holds even for PageRank
- Example: Giraph
  - “Think like a vertex”, no partition notion
  - No multicast support in framework

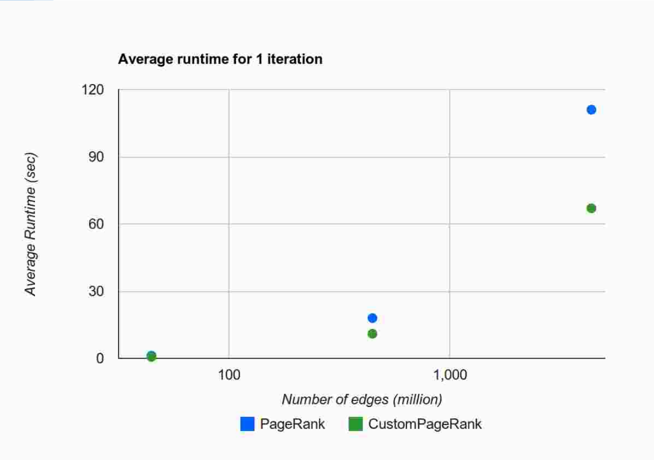
Distributed GraphLab

- Several optimization over plain BSP:
  - Framework support to distribute very high degree nodes: PowerGraph partitions scatters and gathers
  - Optimization: emit unchanged information by caching on gather side
  - Optimization: graph partitioning to reduce number of edges cut (hard to partition a real implicit ratings matrix)
- **But no handling for multiple identical messages**

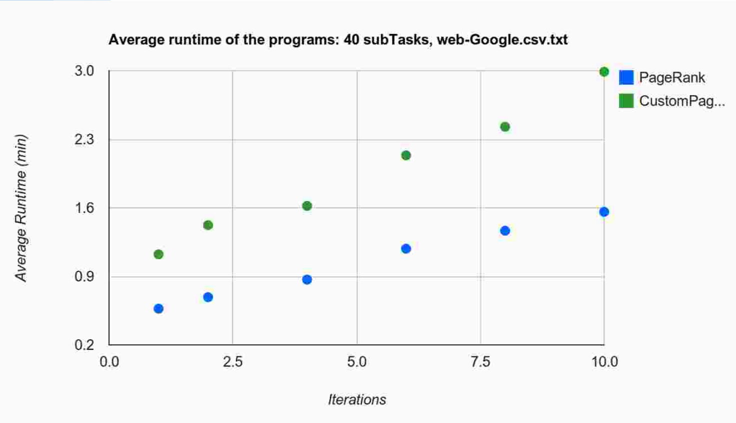
Experiments – distributed message passing in C++

- Proof of concept for a low communication task: PageRank
- We rely on direct control over partitions
- Each vertex sends the message to relevant partitions once
- Test on large Web crawl (`.pt`): 300M nodes, 1B edges
- Significant improvement

CUSTOM PAGERANK IN C++

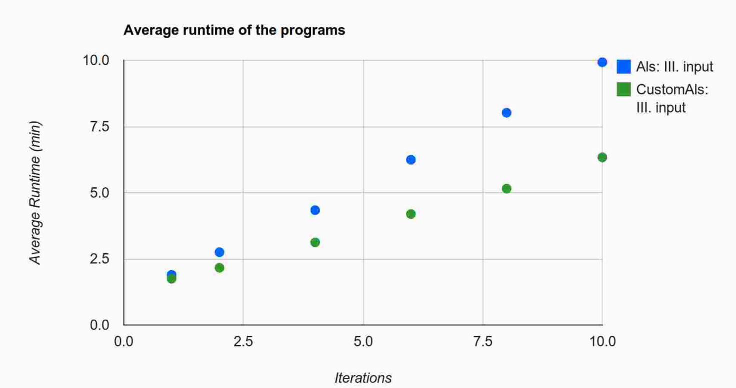


CustomPageRank in Apache Flink



- We define hypernodes – Mahout CustomALS style
- Insufficient for low communication tasks
- Web-Google graph from Stanford Large Network Dataset Collection, $9 \cdot 10^5$ nodes,

CustomALS in Apache Flink



- Generated test data 15 million ratings (courtesy: Gravity)
- Framework support already sufficient for ALS

Conclusions

- ALS multi-machine no shared memory
  - Heavy communication alternating between rows and columns
  - ALS message size is quadratic in number of latent factors
  - Affects MapReduce with no permanent storage (Mahout "CustomALS")
  - Graph parallel frameworks with nonzero ratings mapped to edges
- Ongoing experiments with Message Passing, Giraph, Apache Flink, and its Pregel implementation Spargel.
  - Communication primitives to bind identical messages - use multicast
  - Promising even for seemingly low communication intense algorithms such as PageRank.