Towards A Next Generation Data-Parallel Platform For Graph Processing

Karthik Nilakant (Supervisor: **Dr Eiko Yoneki**)
Email: {firstname.lastname@cl.cam.ac.uk}

Background

Graph-Structured Data

Many large datasets of current interest are best represented using graphs: the global IP network, the worldwide web, social networks and bioinformatic data.

These datasets may be very large, and often grow rapidly. For example, Facebook's social graph has grown to 80 billion edges, in under a decade.

Graph processing on relational database engines suffer from expensive join operations, therefore purpose-built graph databases have emerged.

Graph Processing

Graph algorithms have been studied extensively in computer science, and are already widely used.

Recent interest has focused on parallelising algorithms for tasks such as identifying connected components, calculating popularity metrics and heuristic search.

The **bulk synchronous parallel** model has been adopted for use with existing data-parallel frameworks such as *MapReduce*:

1. Parallel processes operate locally per vertex.
2. Processes then exchange data over edges.
3. Processes synchronise, and re-iterate.

Current issues

Partitioning

Data from neighbouring vertices should be grouped together on one machine, whilst ensuring even distribution of data. Random partitioning will not work well.

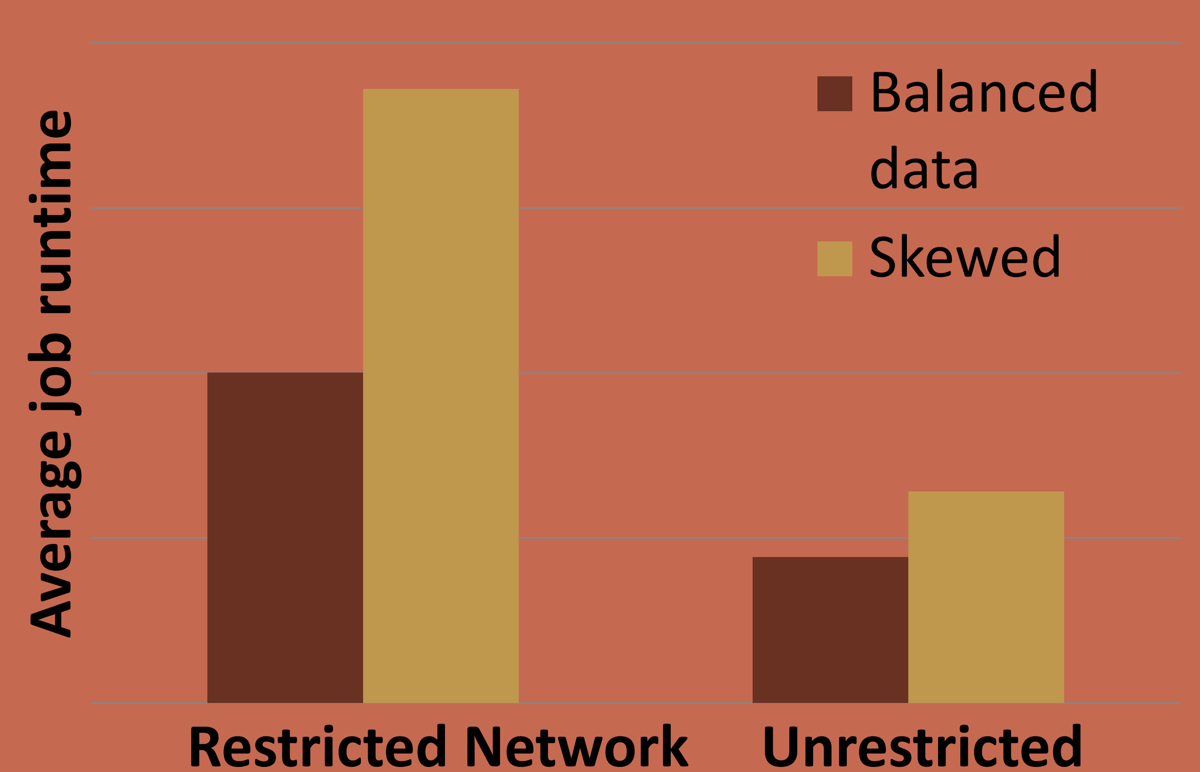
Data Locality

On widely distributed clusters, restricted bandwidth will exacerbate the problems caused by excessive "shuffling" of data.

Parallelism

Not all graph algorithms can be parallelised well, leading to scalability issues. The static task topologies enforced by current platforms are not suited to this.

Insight



This figure illustrates the performance impact on a small *Hadoop* cluster, after data imbalance and network restrictions were introduced.

This could arise from poor partitioning of graph vertices, or poor scheduling locality.

Current *Hadoop*-based BSP approaches perform well on highly parallel tasks such as PageRank, but are outperformed by in-memory databases for low-parallelism tasks.

Dynamic task graph generation may assist in reducing the overhead on iterative graph computations

Goals

Our aim is to develop a new graph-processing platform that achieves the following:

- Active organisation of data based on semantic locality, adapting to incremental changes.
- Data-local scheduling of tasks, and minimisation of network communication.
- Applicable to both sequential and highly parallel computations.

Systems Research Group

srg@cambridge:~$

Microsoft Research

UNIVERSITY OF CAMBRIDGE

This work is supported by Microsoft Research through its PhD Scholarship Programme.