DAIET: Data Aggregation In nETwork

Amedeo Sapio, Ibrahim Abdelaziz, Marco Canini, Panos Kalnis

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

Problem

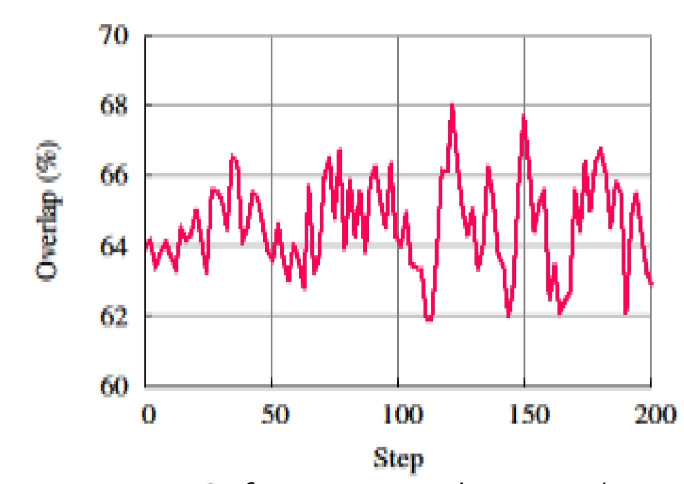
Network communications are often a scalability bottleneck for partition/aggregate applications

In applications like MapReduce, machine learning, graph processing, data and computations are distributed among many servers and partial results exchanged over the network may generate a large number of flows that overrun buffers and receivers

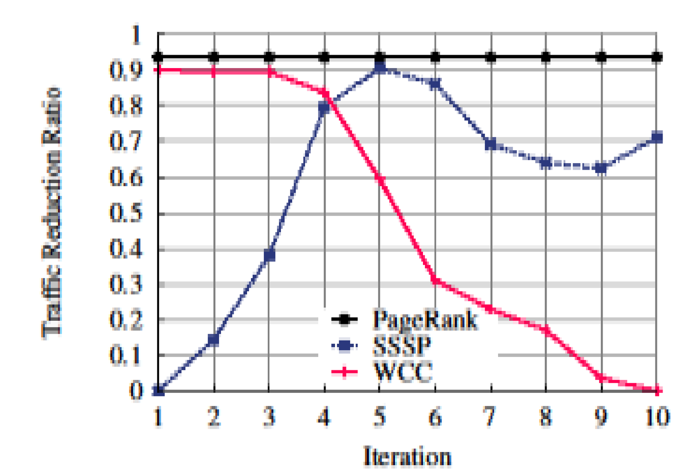
Motivating Scenarios

Experimental results show that the updates sent at the end of an iteration in Machine learning algorithms have a significant overlap (**25%-68%**), which represents the potential gain of in-network aggregation

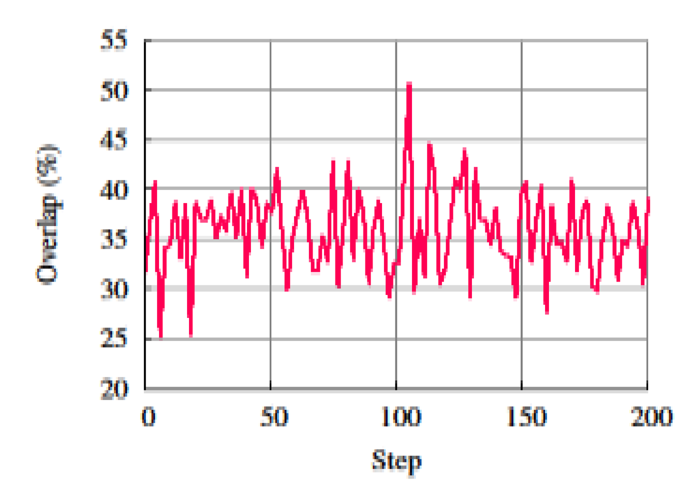
Similarly, aggregation of messages exchanged between vertices in Graph Analytics algorithms can provide a large traffic reduction (**48%-93%**)



Soft-Max Neural Network



Graph Analytics Algorithms



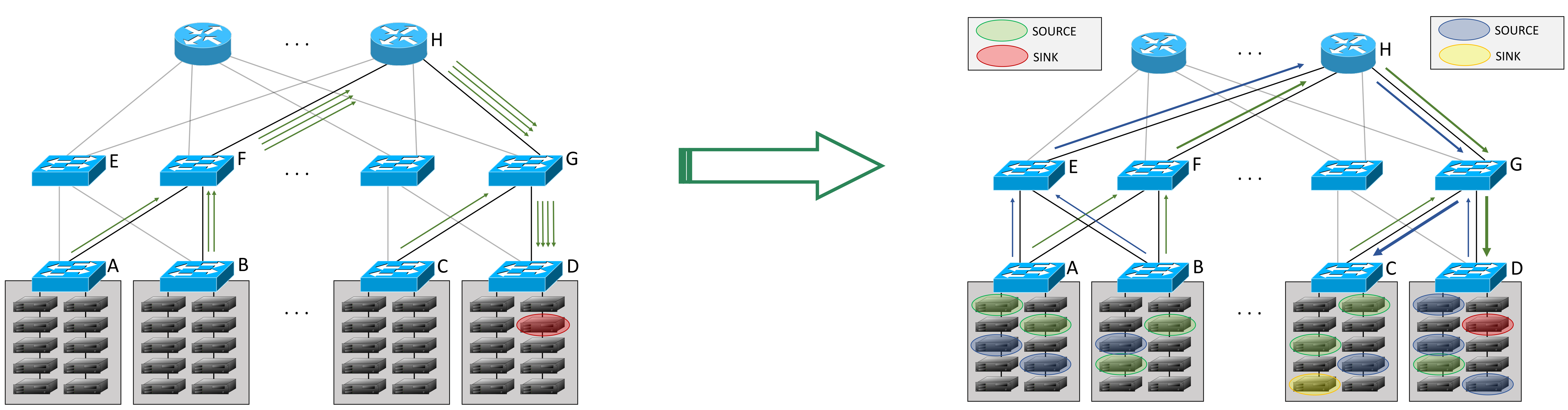
Stochastic Gradient Descent

Approach

DAIET performs data aggregation along network paths using programmable network devices

Offloading to switches and smart NICs part of the computation reduces traffic on the datacenter fabric and the amount of work at destination end hosts

This alleviates communication bottlenecks and can improve overall job completion times

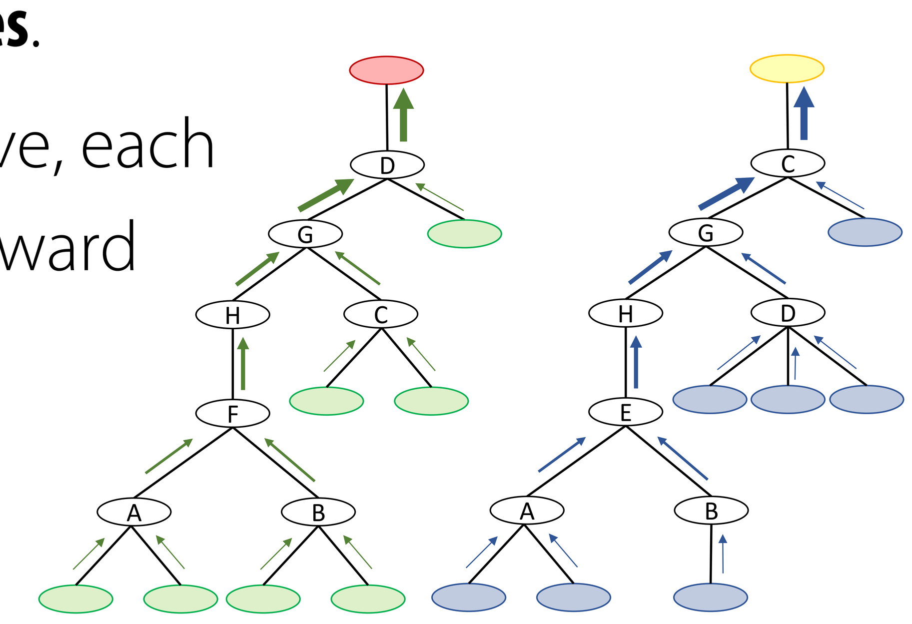


Many-to-one communications are modeled as **aggregation trees**.

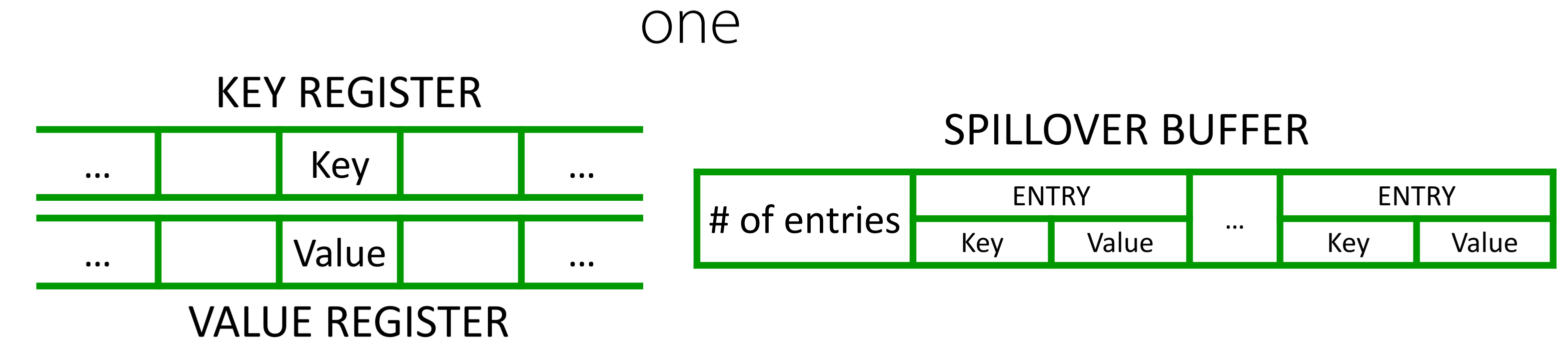
When the aggregation function is commutative and associative, each node can independently aggregate data before sending it forward

Each intermediate node stores a per-tree list of <key, value> pairs until the end of all children's transmissions.

Then the entire list can be forwarded to the next node



Given the limited amount of memory of data-plane devices, only a limited number of pairs can be aggregated by a node, while the remaining ones are plainly forwarded to the next one

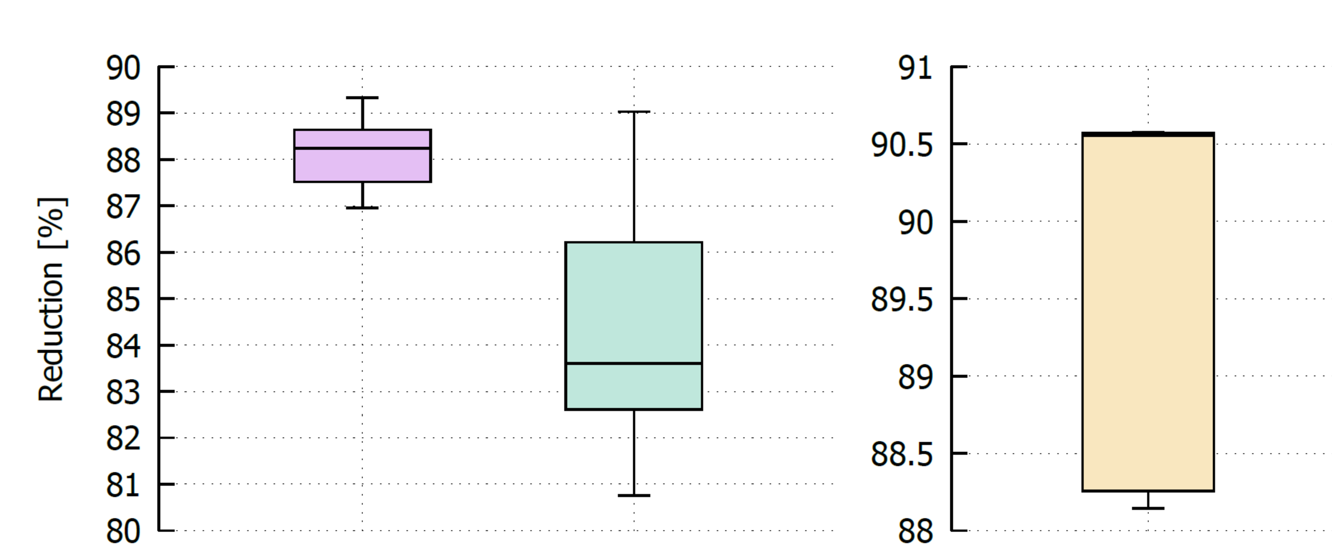


Evaluated a DAIET prototype in P4 for a MapReduce WordCount application with a 500 MB dataset

Measured the reduction in network traffic and in time spent at the reducer compared to no aggregation

Results show **87%-89% reduction of data received** at the reducer and a **median decrease of 84% of the running time** at the reducer

Number of packets received by the reducer is 90% compared to a UDP-based protocol and 42% compared to TCP



https://sands.kaust.edu.sa/daiet/