SC 2015

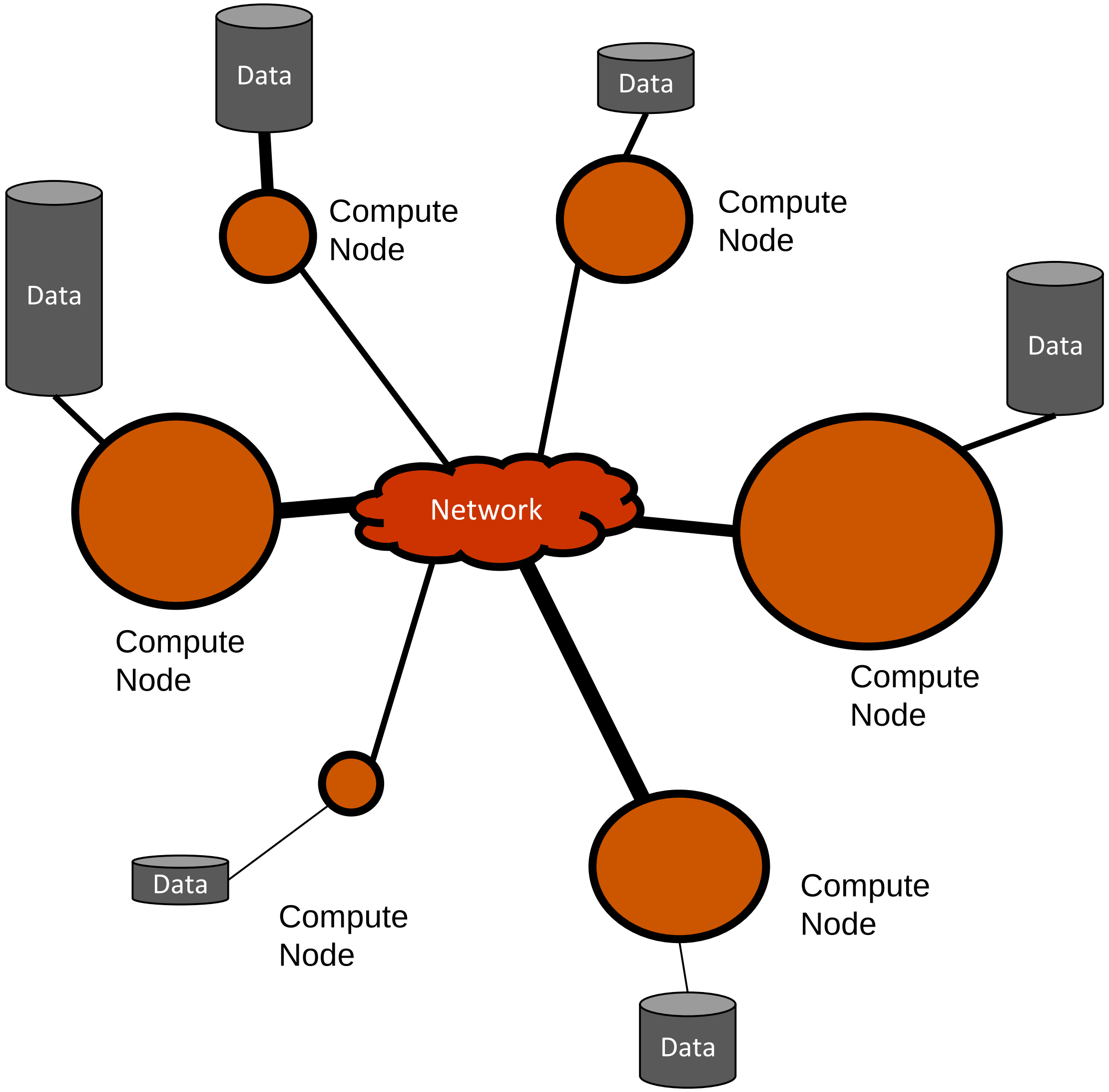
The University of Texas at Austin
Electrical and Computer Engineering

Data Partitioning Strategies for Graph Workloads on Heterogeneous Clusters

Michael LeBeane, Shuang Song, Reena Panda, Jee Ho Ryoo, Lizy K. John
The University of Texas at Austin
mlebeane@utexas.edu

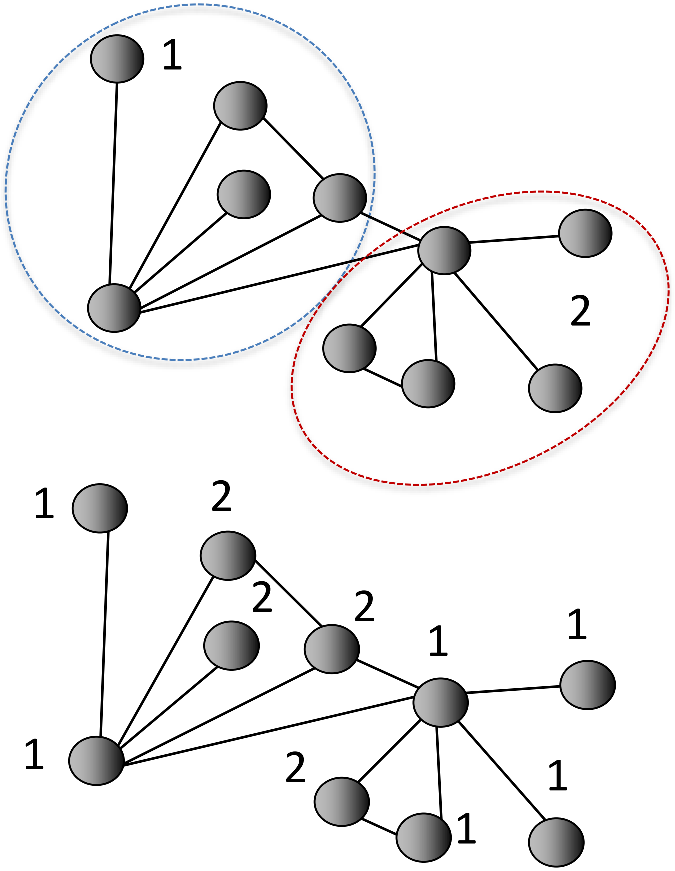
Motivation

- **Heterogeneity is pervasive in modern data centers [][]**
- **Graph analytics are a pervasive workload in the data center []**
  - Many frameworks available to efficiently and easily perform graph analytics [][][][]
- **Most frameworks are not equipped to deal with heterogeneity in the data center**



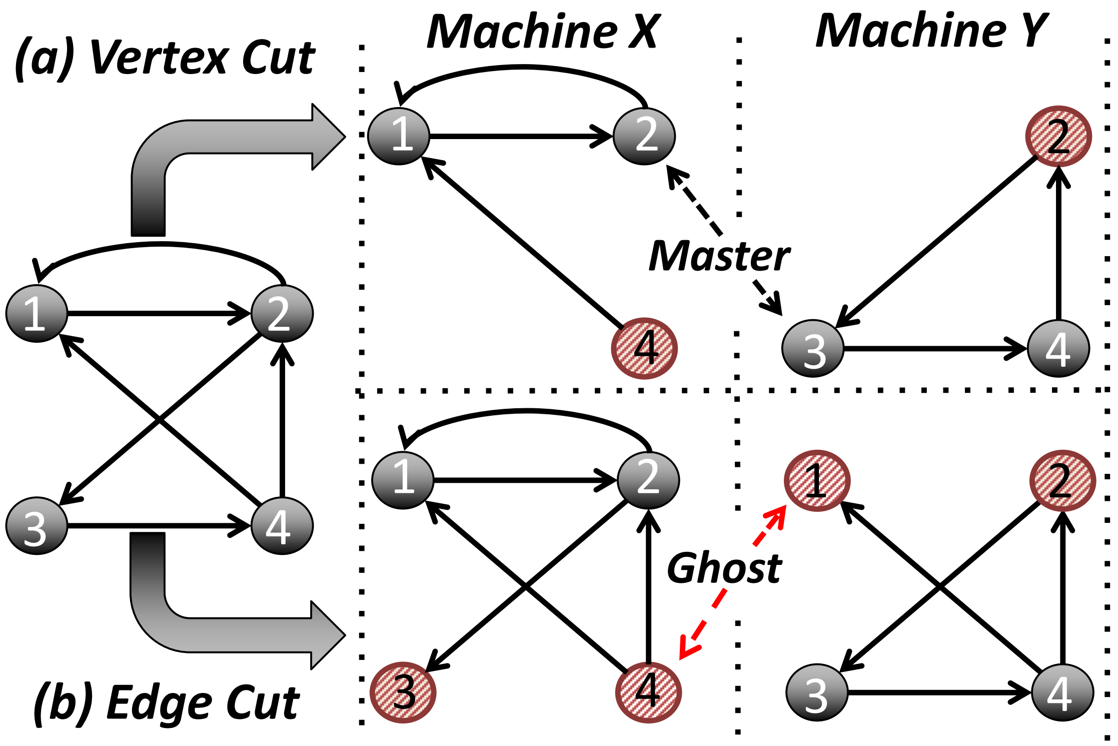
Background

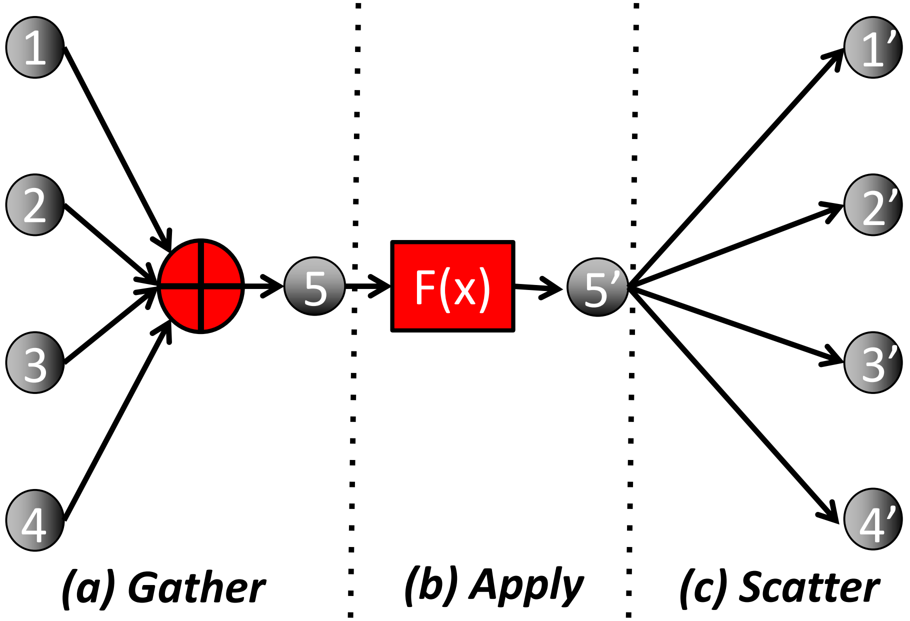
- **All work performed on PowerGraph[] framework**
- **Three relevant graph partitioning topics:**
  - Online vs. Offline Partitioning
  - Vertex vs. Edge Cut
  - Gather/Apply/Scatter



- **Online vs. Offline Partitioning**

Background

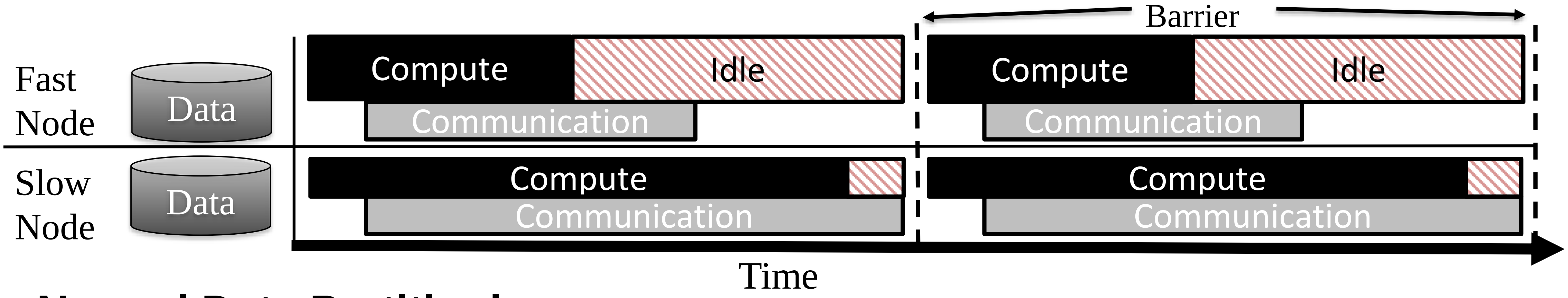




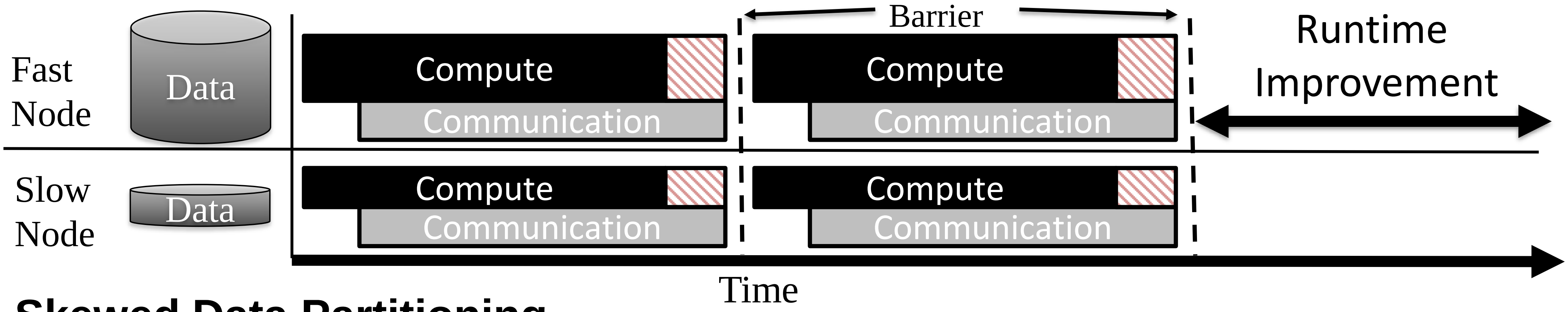
- **Vertex vs. Edge Cut**

- **Gather/Apply/Scatter**

Workload Skew in Heterogeneous Data Centers



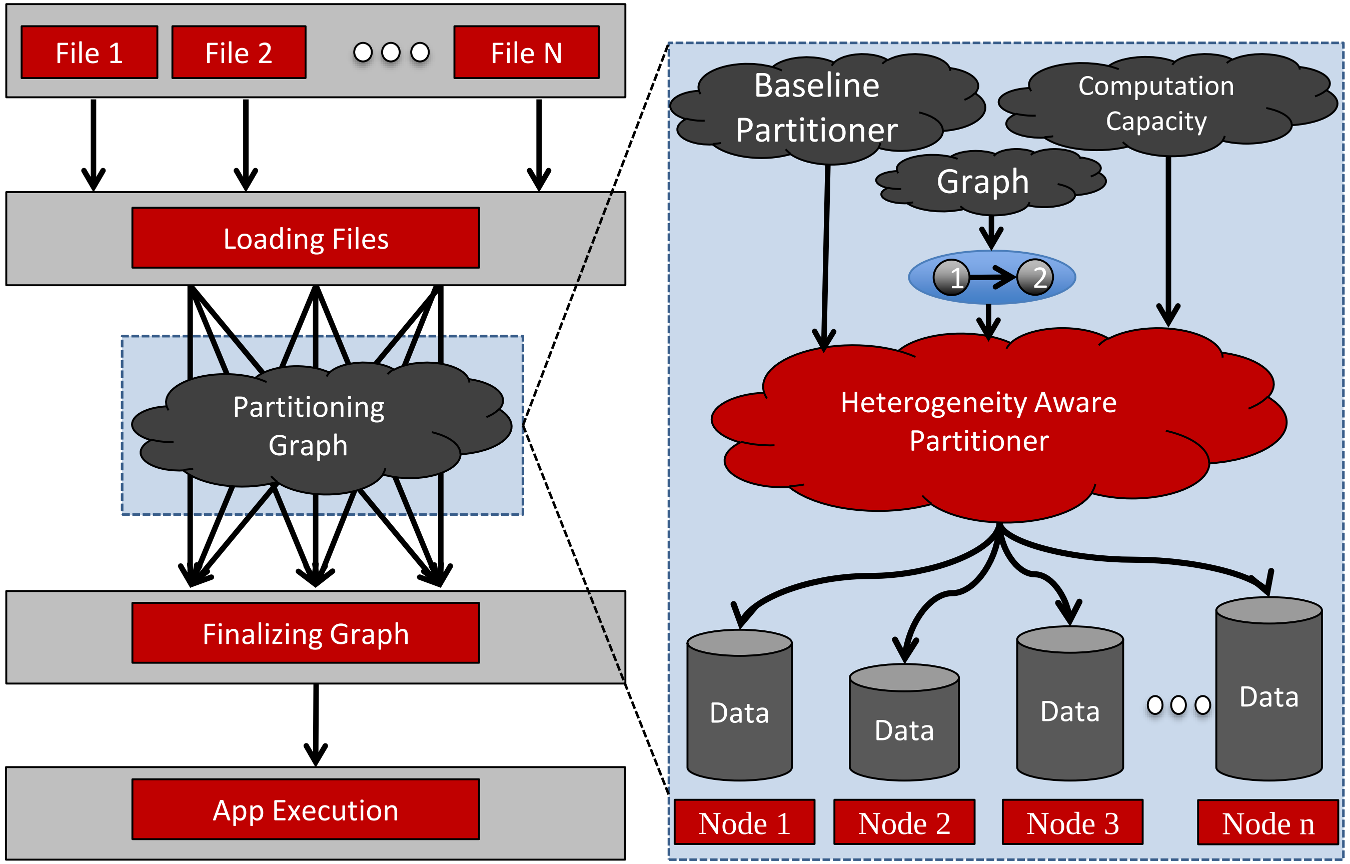
- **Normal Data Partitioning**



- **Skewed Data Partitioning**

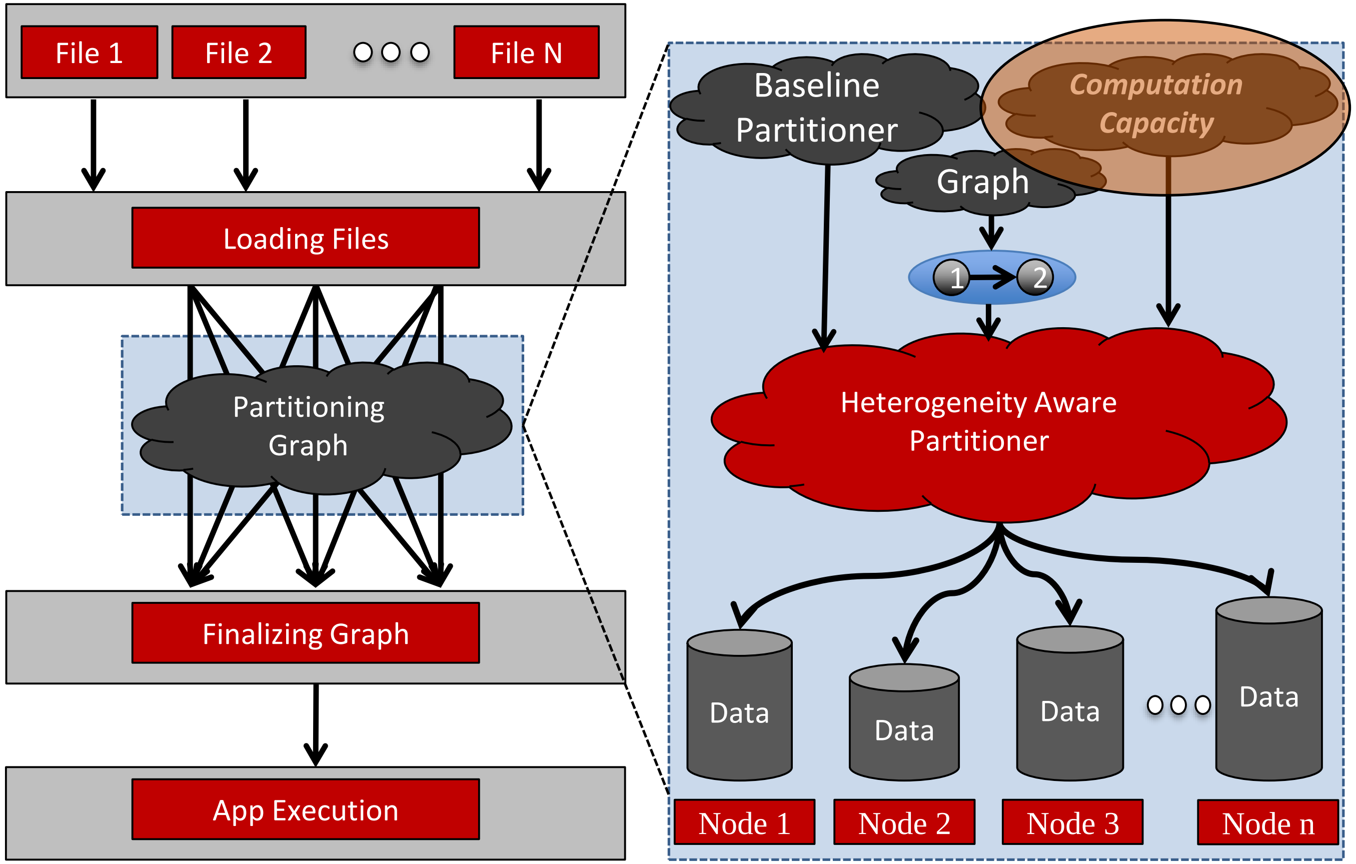
Heterogeneous Graph Analytics

- **Local node computation time dependent on data distribution**
- **To properly balance work, we need:**
  - Estimation of each node’s computational capacity
  - Partitioning algorithms that account for skewed computational capacity



Heterogeneous Computation Capacity

- **Computation capacity is complex**
- **Dependent on many factors:**
  - Hardware of the node
  - Nature of the graph
  - Nature of the algorithm
  - Communication patterns
- **Can we determine a simple, static estimate?**



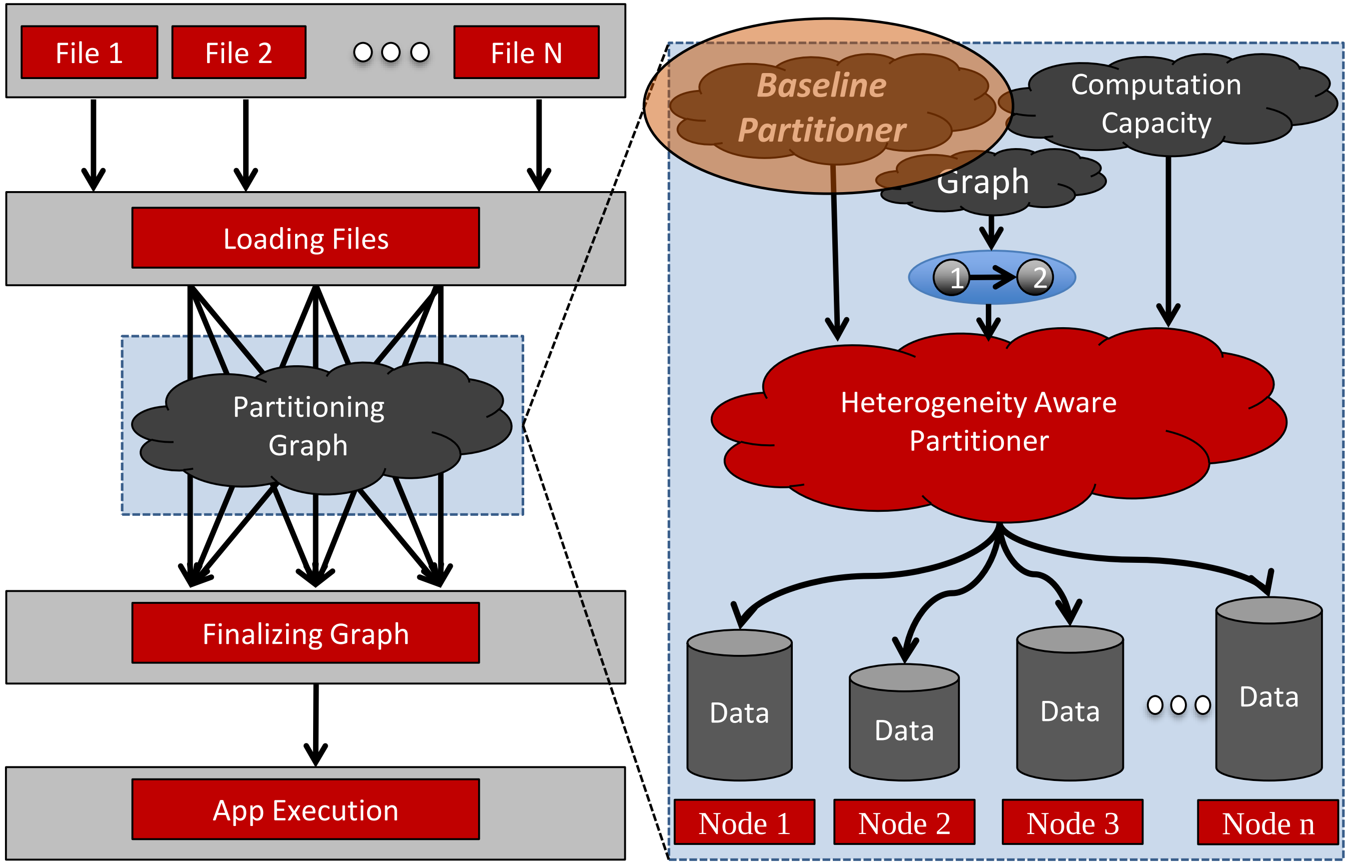
Skew Factor Calculation

Name	HW Threads	Memory	Network	Thread Skew Factor	Memory Skew Factor
c4.xlarge	4	7.5 GB	100 Mbps to 1.86 Gbps	1	1
c4.2xlarge	8	15 GB	100 Mbps to 1.86 Gbps	3	2
c4.4xlarge	16	30 GB	100 Mbps to 1.86 Gbps	7	4
c4.8xlarge	36	60 GB	up to 8.86 Gbps	17	8

- **Static estimate of node computational capacity could be based on:**
  - Threads: Logical compute threads on node (default N – 2)
  - Memory: Physical memory assigned to a node
  - Profiling: Local throughput of graph subset and algorithm
- **We will refer to the estimated ratios of computation capacity as the *skew factor* of the heterogeneous data center**

Heterogeneous Partitioning Algorithm

- **Online partitioning algorithms must be modified to support skew factor**
- **Easy to modify current online partitioning algorithms**
- **We have modified 5 popular algorithms from multiple sources**

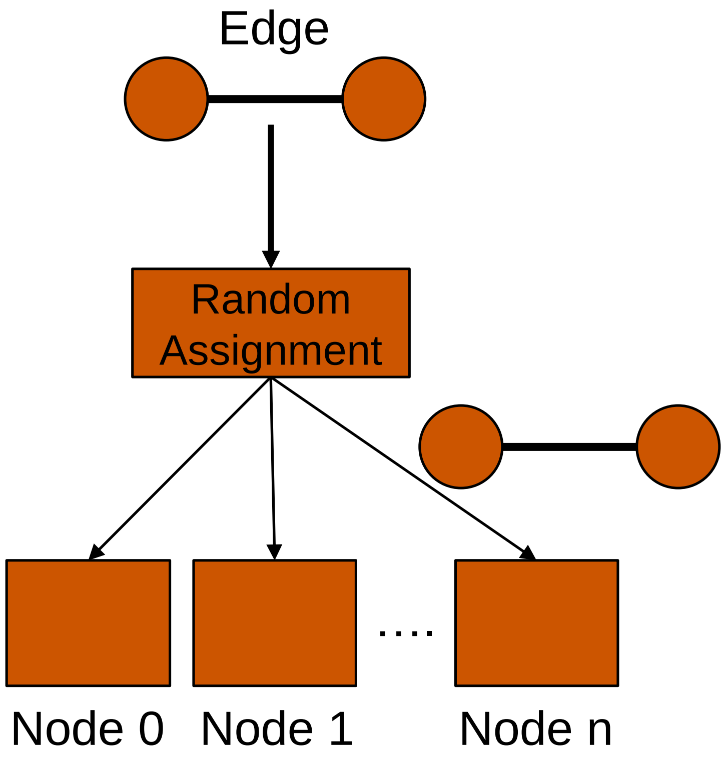


Problem Formulation

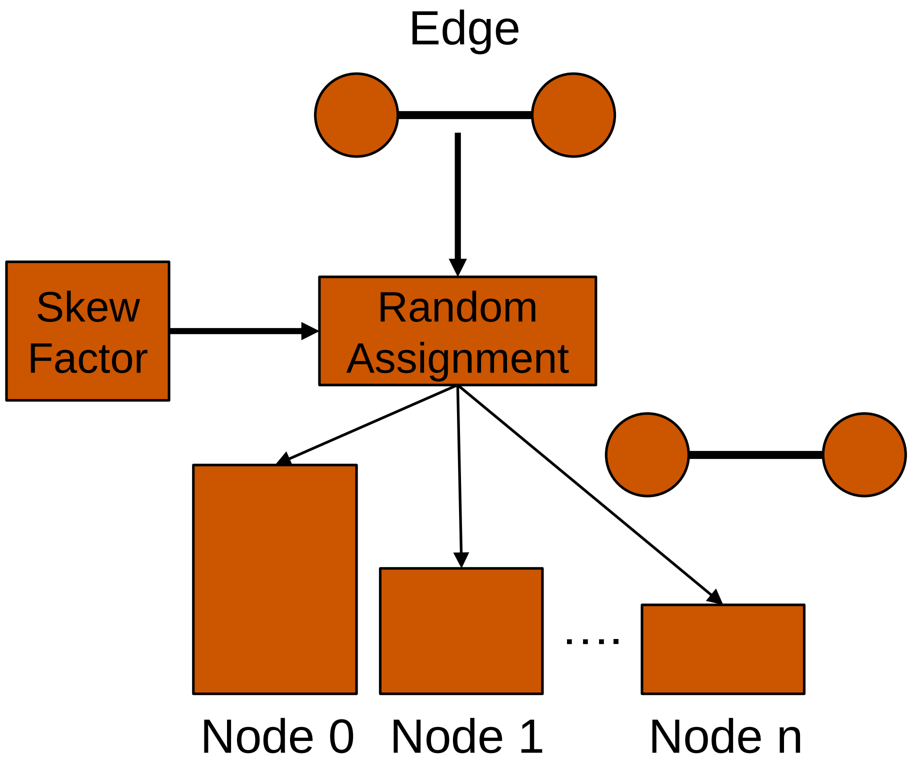
- **Statically estimated based on:**
  - Threads: Logical compute threads on node (default N – 2)
  - Memory: Physical memory assigned to a node
  - Profiling: Local throughput of graph subset and algorithm

- **Statically estimated based on:**

Random Skewed Partitioner



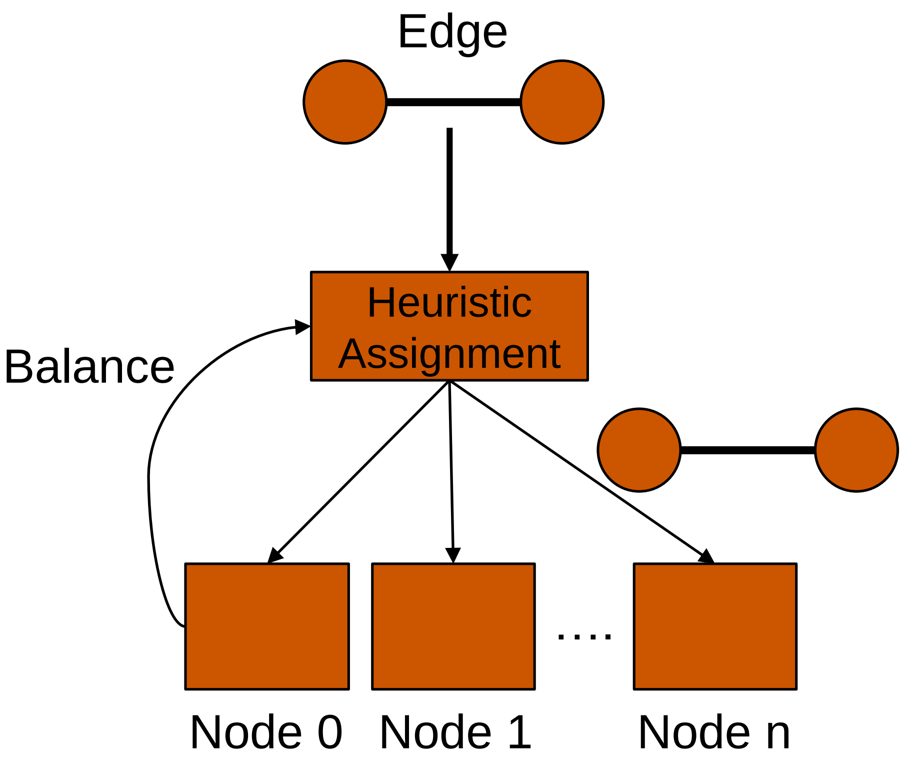
- **Original**



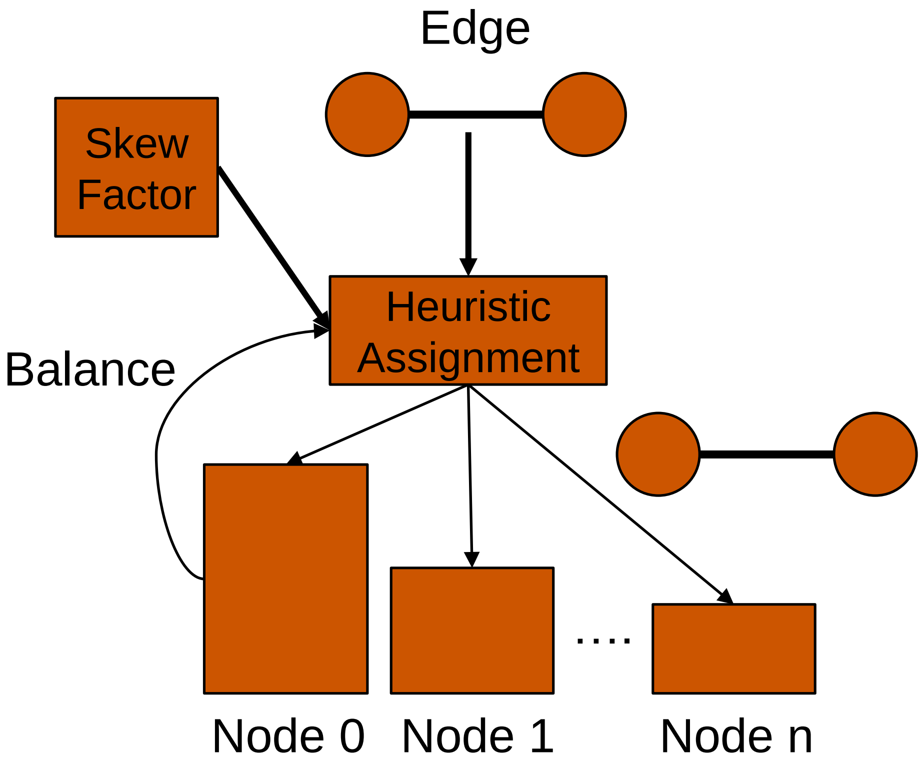
- **Skewed**

- **Random assignment of edges to nodes**

Greedy Skewed Partitioner



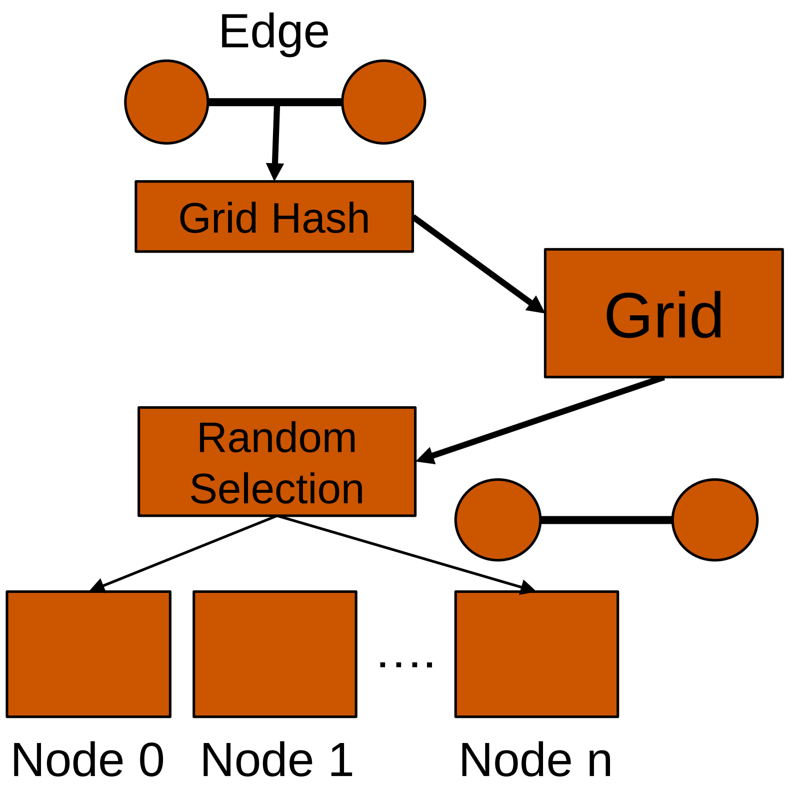
- **Original**



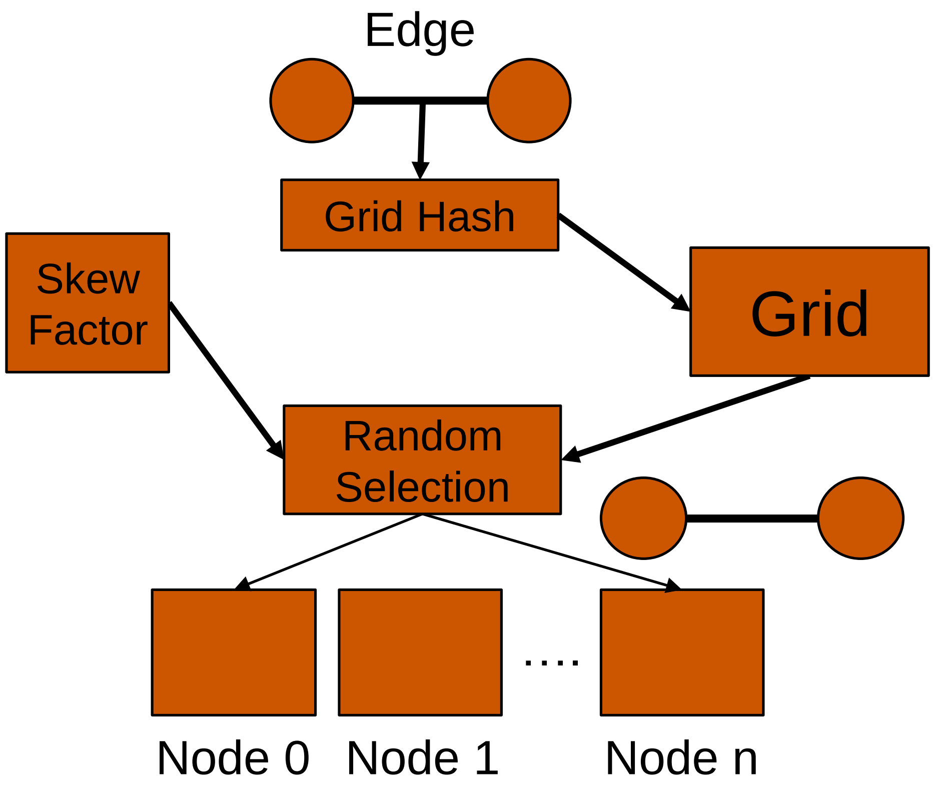
- **Skewed**

- **Greedy decision using current distribution of edges**
  - Either locally or coordinated

Grid Skewed Partitioner



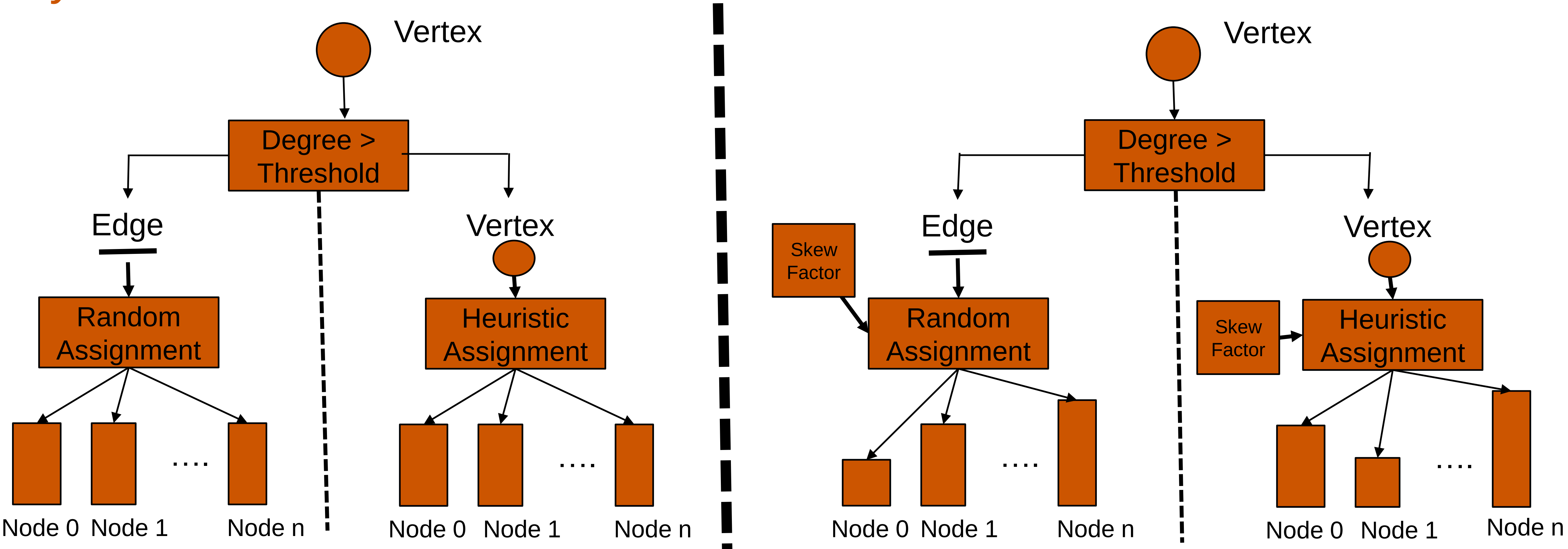
- **Original**



- **Skewed**

- **Greedy decision using current distribution of edges**
  - Either locally or coordinated

Hybrid Skewed Partitioner

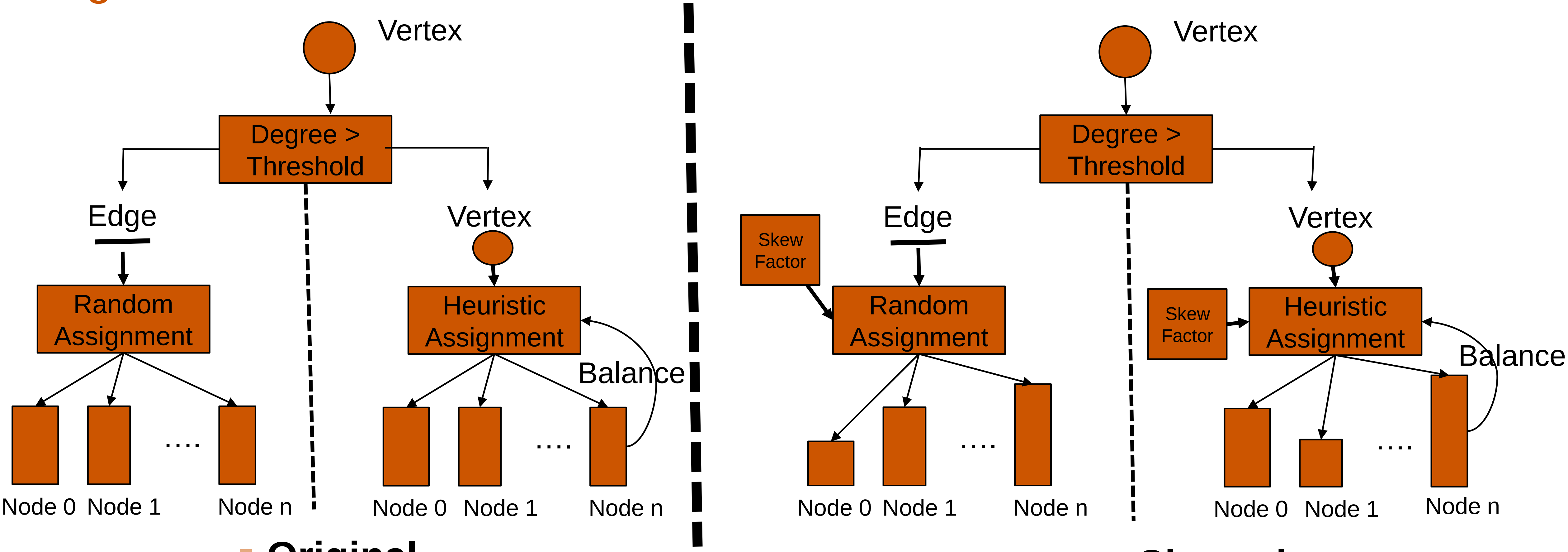


- **Original**

- **Skewed**

- **Random assignment of edges/verticies to nodes based on degree**

Ginger Skewed Partitioner



- **Original**
- **Skewed**

- **Random assignment of edges/verticies to nodes based on degree**

Experimental Setup

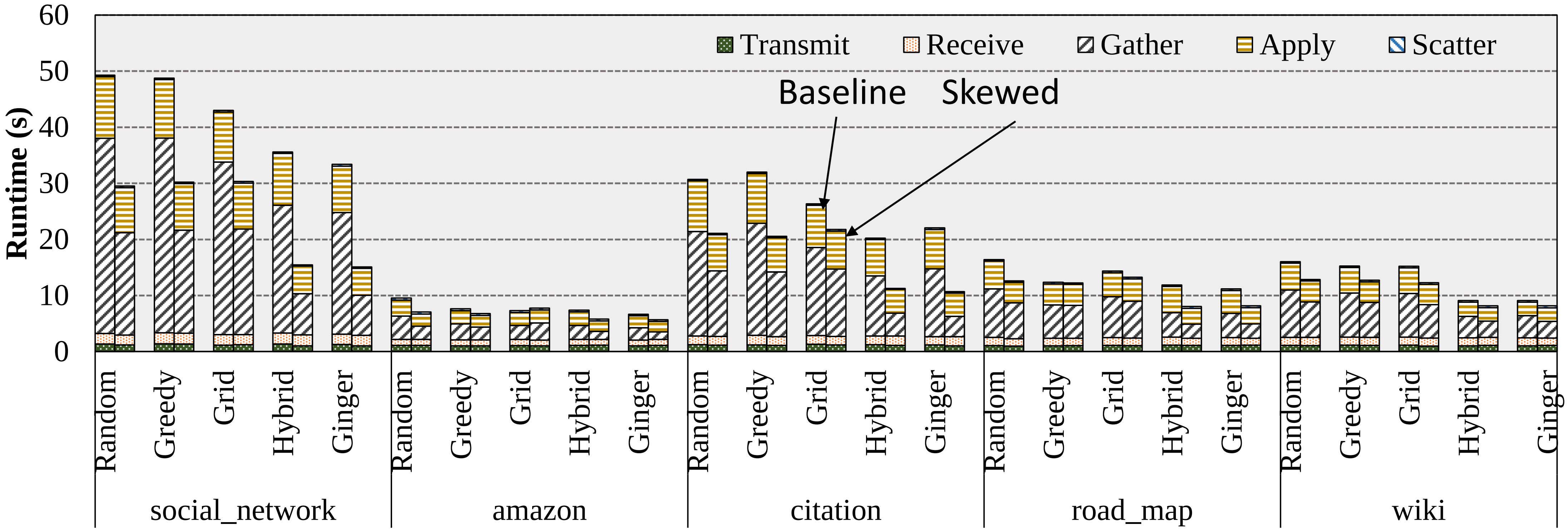
- **Algorithms**
  - Graph: PageRank (PR), Connected Components (CC), Triangle Count (TC)
  - Matrix: Stochastic Gradient Descent (SGD), Alternating Least Squares (ALS)
- **Data Sets**

Name	Vertices	Edges	Size (Uncompressed)	Type	Algorithms
amazon	403,394	3,384,388	46MB	Directed Graph	PR,CC,TC
citation	3,774,768,NA	16,518,948	268MB	Directed Graph	PR,CC,TC
netflix	NA	NA	100MB	Sparse Matrix	ALS,SGD
road-map	1,379,917	1,921,660	84MB	Undirected Graph	PR,CC,TC
social-network	4,847,571	68,993,773	1.1GB	Directed Graph	PR,CC,TC
twitter	41,000,000	1,400,000,000	25GB	Directed Graph	PR,CC,TC
wiki	2,394,385	5,021,410	64MB	Directed Graph	PR,CC,TC

Experimental Setup

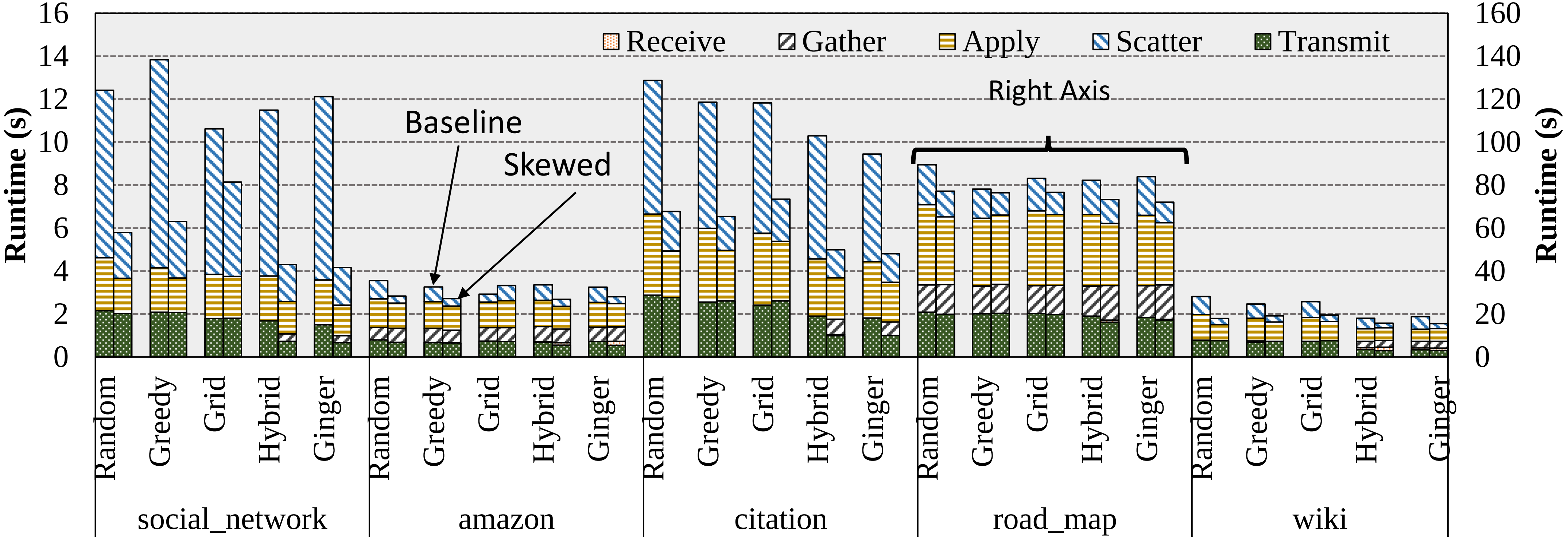
- **Data Center**
  - Graph: PageRank (PR), Connected Components (CC), Triangle Count (TC)
  - Matrix: Stochastic Gradient Descent (SGD), Alternating Least Squares (ALS)
- **Skew Factor**
  - Results use Thread Based Skew Factor

Execution Time



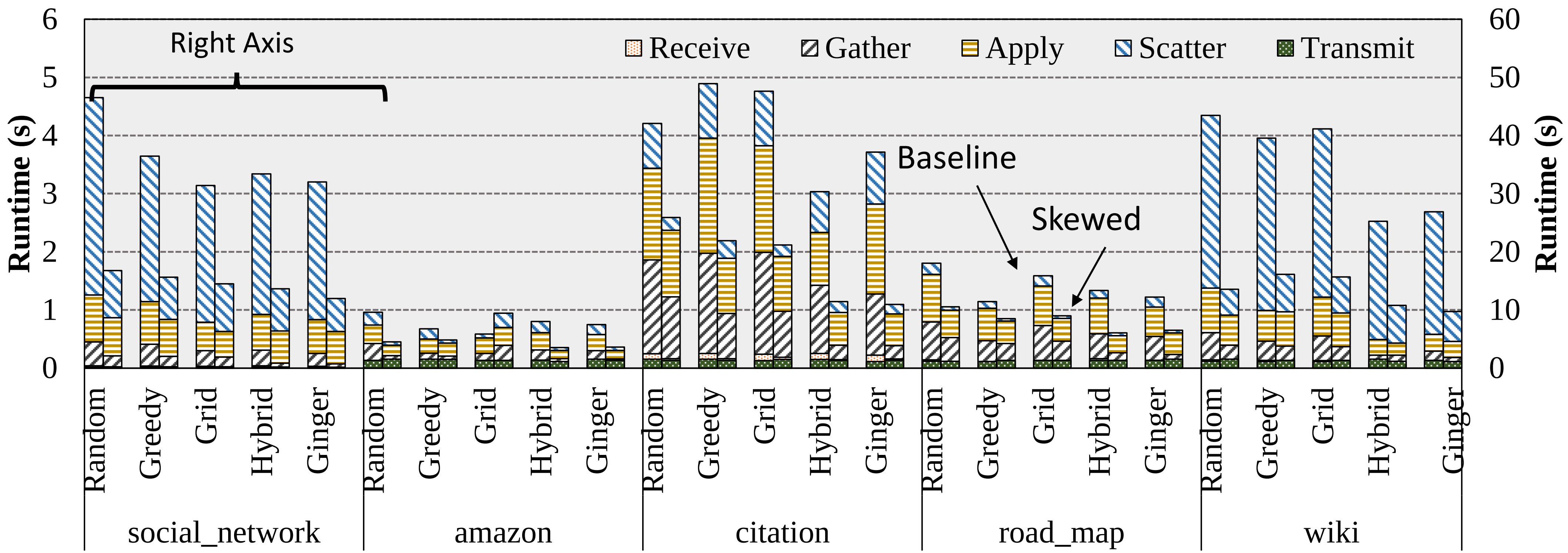
- **Pagerank**

Execution Time



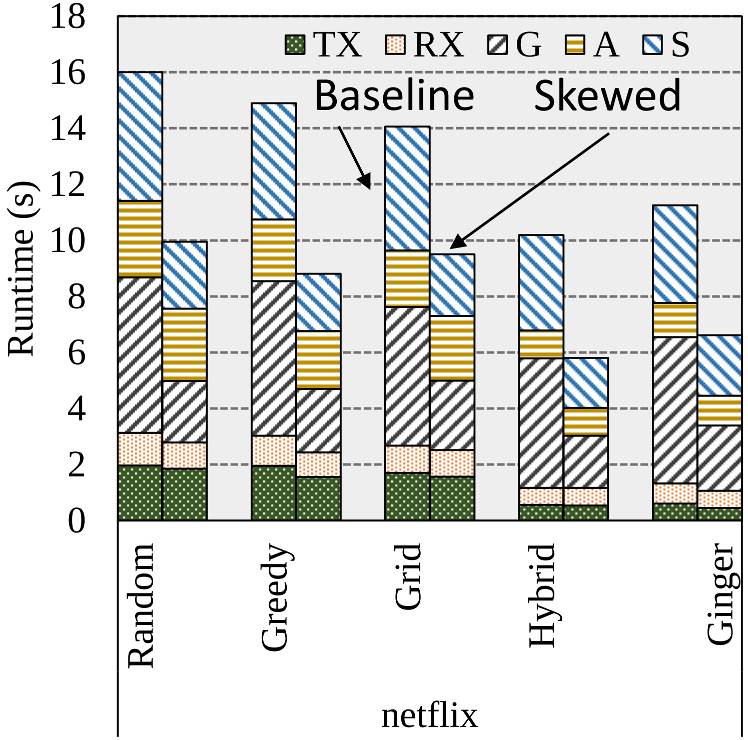
- **Connected Components**

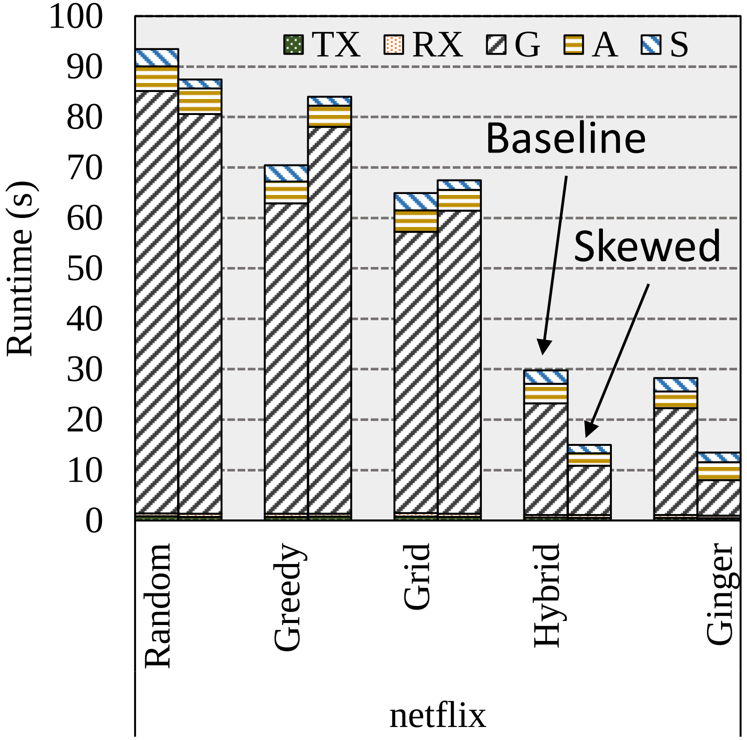
Execution Time



- **Triangle Count**

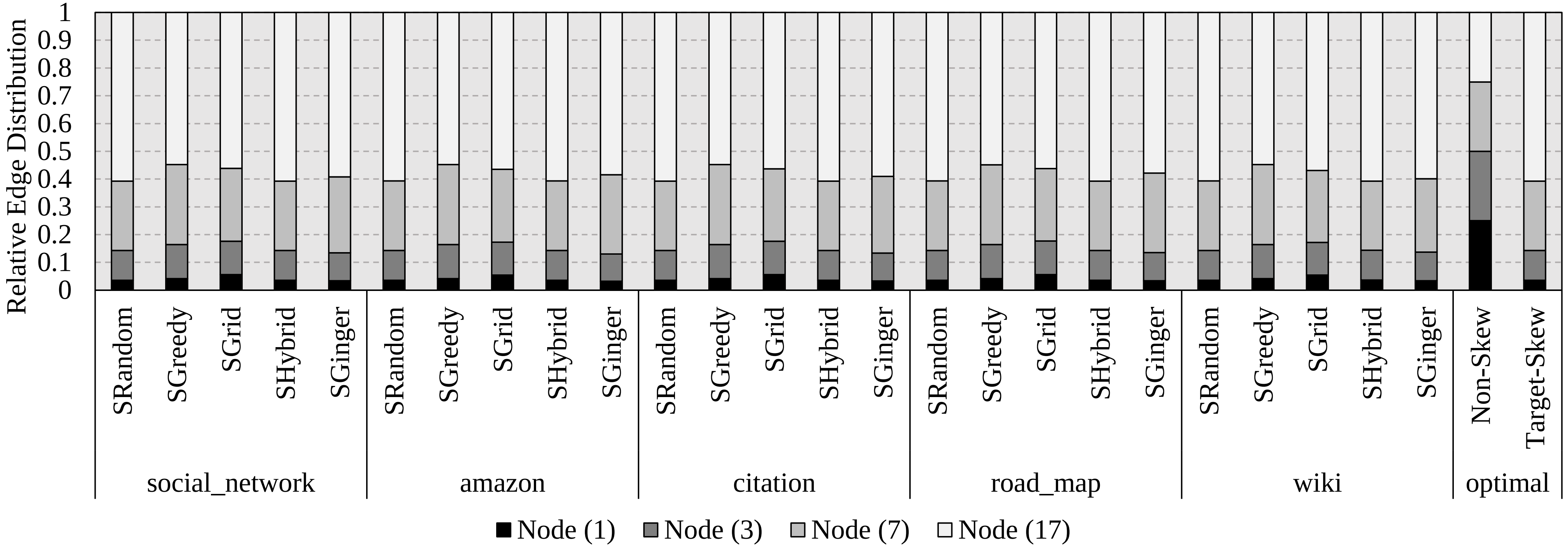
Execution Time





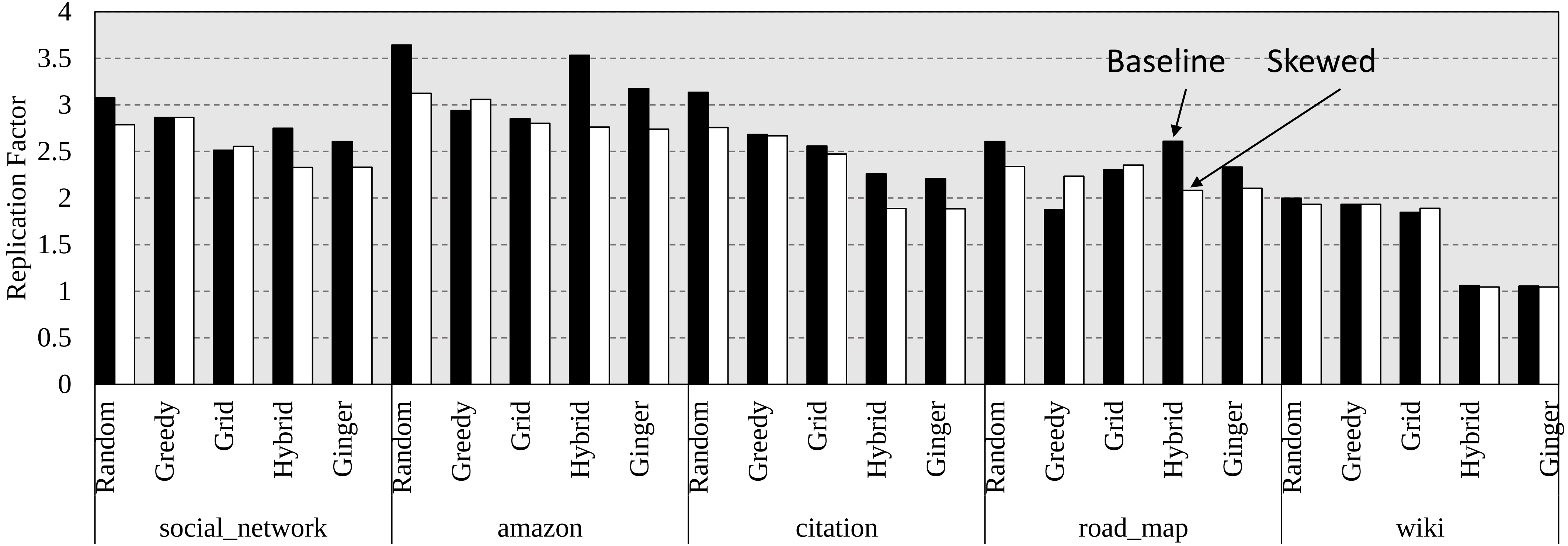
- **Stochastic Gradient Descent**
- **Alternating Least Squares**

Data distribution



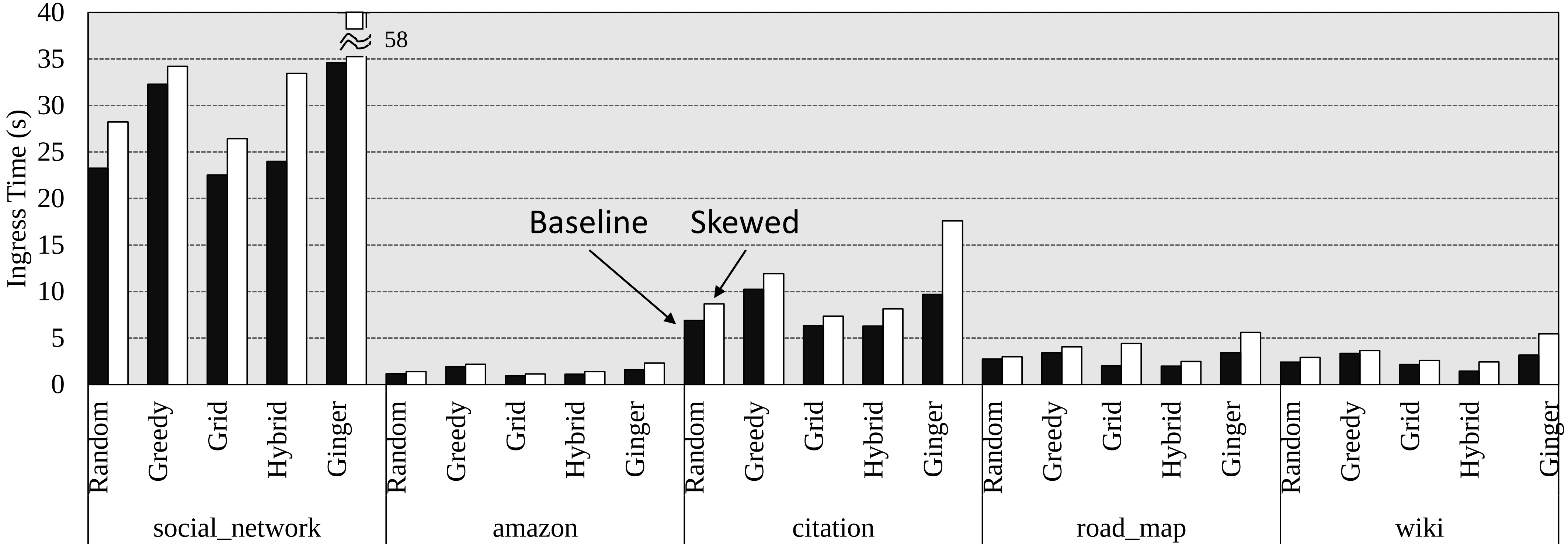
- **Ideal distribution 17-7-3-1**

Results



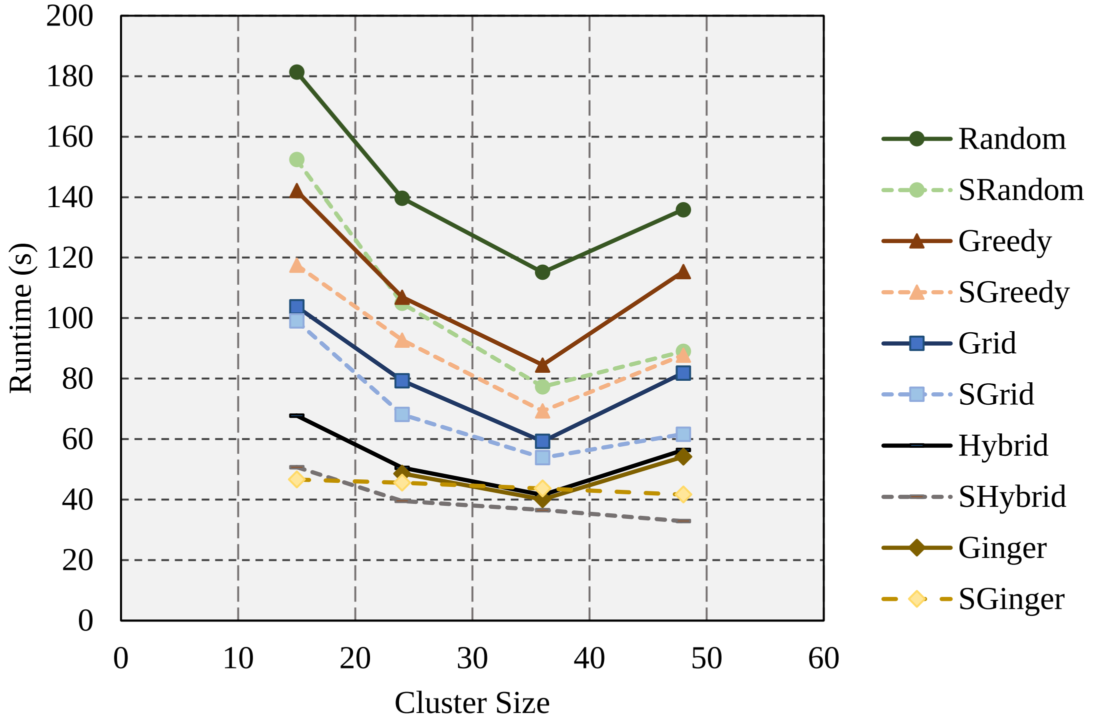
- **Skewed approach generally decreases network communication**

Results

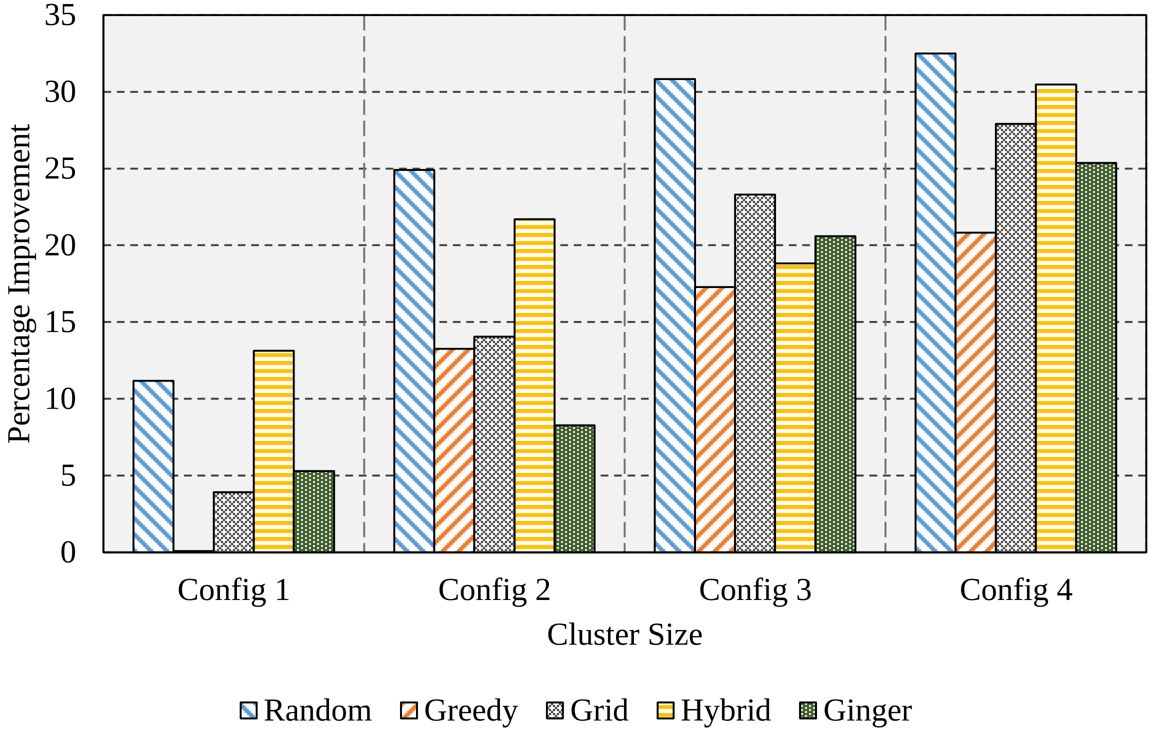


- **Data Ingress Time**

Scale-out Results



- **Extremely large Twitter graph**
- **No benefits after 36 nodes**



Configuration Name	C4.2xlarge	C4.4xlarge	C4.8xlarge
Config 1	12	8	4
Config 2	8	8	8
Config 3	4	8	12
Config 4	3	5	16

Future Work

- **Incorporate better network model**
- **Profile based partitioning scheme**
  - How do we sample graph inputs?

Conclusion

- **Simple, static throughput estimation can greatly improve performance**
- **We modify 5 existing on-line graph partitioning strategies for heterogeneous environments**
- **Our modified algorithms improve runtime by as much as 64% and on average 32% on Amazon EC2**
- **We show that our strategies also work up to 48 nodes, achieving 18% performance improvement on scale-out**

Thank You!

References

[1] S. Garg, S. Sundaram, and H. D. Patel. Robust heterogeneous data center design: A principled approach. *SIGMETRICS Perform. Eval. Rev.*, 39(3):28–30, Dec. 2011.

[2] B.-G. Chun, G. Iannaccone, G. Iannaccone, R. Katz, G. Lee, and L. Niccolini. An energy case for hybrid datacenters. *SIGOPS Oper. Syst. Rev.*, 4(1):76–80, Mar. 2010.

[1] J. E. Gonzalez, Y. Low, H. Gu, D. Bickson, and C. Guestrin. Powergraph: Distributed graph-parallel computation on natural graphs. In *OSDI*, pages 17–30. USENIX Association, 2012.