Sparse Communication Simulation

Kutay Tasci

December 2025

1 Communication Graph Generator: Parameter Guide

The communication graph generator produces a synthetic graph in four conceptual stages: (i) process-level communication pattern generation, (ii) message volume assignment, (iii) partition (vertex block) sizing, and (iv) fine-grained vertex-level edge generation. Each stage is controlled by a dedicated set of parameters.

1.1 Process-Level Communication Pattern

```
number_of_processes
average_communication_degree
communication_skew
```

- **number_of_processes**: Number of processes (or partitions) in the system. This determines the number of nodes in the process-level communication graph. Internally, the value is rounded up to the nearest power of two to satisfy R-MAT requirements.
- **average_communication_degree**: Desired average number of communication partners per process. Higher values lead to denser communication patterns.
- **communication_skew** ($\in [0, 1]$): Controls the unevenness of communication. Values close to 0 result in near-uniform communication, while values close to 1 produce highly skewed patterns where a small number of processes dominate communication.

These parameters are used to generate a directed R-MAT graph, where vertices represent processes and edges represent communication relationships.

1.2 Message Volume Assignment

```
message_volumes:
```

```
  max_volume
  mode
  skew
  target_mean
  heavy_tail
  seed
```

- **max_volume**: Maximum allowable message volume on any communication edge. All volumes are clamped to the range $[1, \texttt{max\_volume}]$.
- **mode**: Distribution used to sample message volumes. Supported options include:
  - `power`: Simple and controllable skew via a power-law transform.
  - `lognormal`: Heavy-tailed distribution commonly observed in real communication workloads.
- **skew**: Controls the variance or skewness of the volume distribution. Larger values result in heavier tails and higher variance.
- **heavy_tail**: If enabled, biases the distribution toward larger message volumes.
- **target_mean**: Optional parameter that rescales sampled volumes to match a desired average message size.
- **seed**: Random seed for reproducible message volume generation.

After this stage, the process-level graph becomes a directed, weighted graph, where edge weights correspond to message volumes.

1.3 Partition Size Initialization

```
initialize_graph:
  mean_position
  skew
  seed
```

- **mean_position** ($\in [-1, 1]$): Determines where the expected partition size lies between the minimum and maximum feasible values. A value of -1 biases sizes toward the minimum, 0 toward the midpoint, and $+1$ toward the maximum.
- **skew**: Controls the spread (standard deviation) of partition sizes. A value of 0 produces nearly uniform partitions, while larger values introduce increasing imbalance.
- **seed**: Random seed for reproducible partition size assignment.

Partition sizes are sampled from a truncated normal distribution and determine the number of vertices assigned to each process.

1.4 Fine-Grained Vertex-Level Edge Generation

```
generate_edges:
  edge_connection_prob
  seed
```

- **edge_connection_prob** ($\in [0, 1]$): Probability of adding additional edges between sampled vertex pairs across communicating partitions. Lower values produce sparse bipartite connections, while higher values lead to denser graphs.
- **seed**: Random seed for reproducible edge sampling.

For each communicating process pair (u, v), distinct vertices are sampled from their respective partitions based on the assigned message volumes. The construction guarantees that every sampled vertex participates in at least one inter-partition edge.