Modeling Partitioned MPI Communication Performance

Jered Dominguez-Trujillo

Prof. Patrick G. Bridges

UNM Computer Science

Background

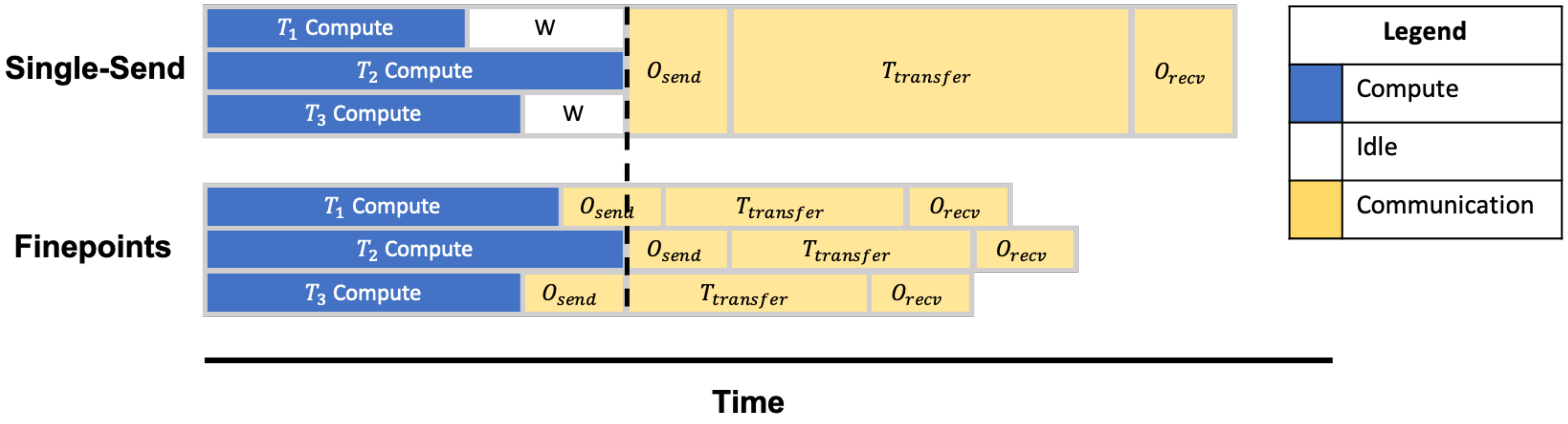
- **Partitioned Communication**
  - A new addition to the MPI specification intended to improve the communication performance on many-core CPUs and GPUs by overlapping communication with computation

- **Current Work**
  - Characterize, model, and predict performance gains able to be realized by using partitioned communication

- **Looking Ahead**
  - Leverage performance models and predictive capability to optimize partitioned communication routines with message aggregation and scheduling in real and proxy applications

Single-Send vs. Finepoints

- **Single-Send:** Single large message after last thread completes
- **Finepoints:** Multiple small messages, each sent once each thread completes

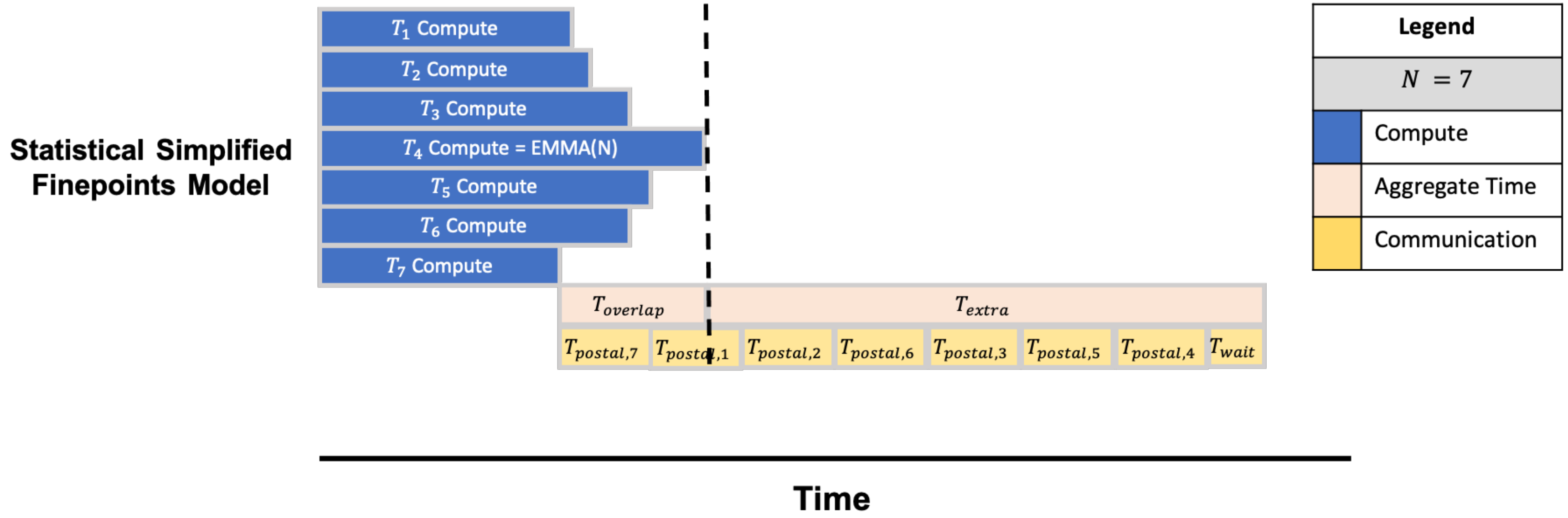


Partitioned Modeling Assumptions

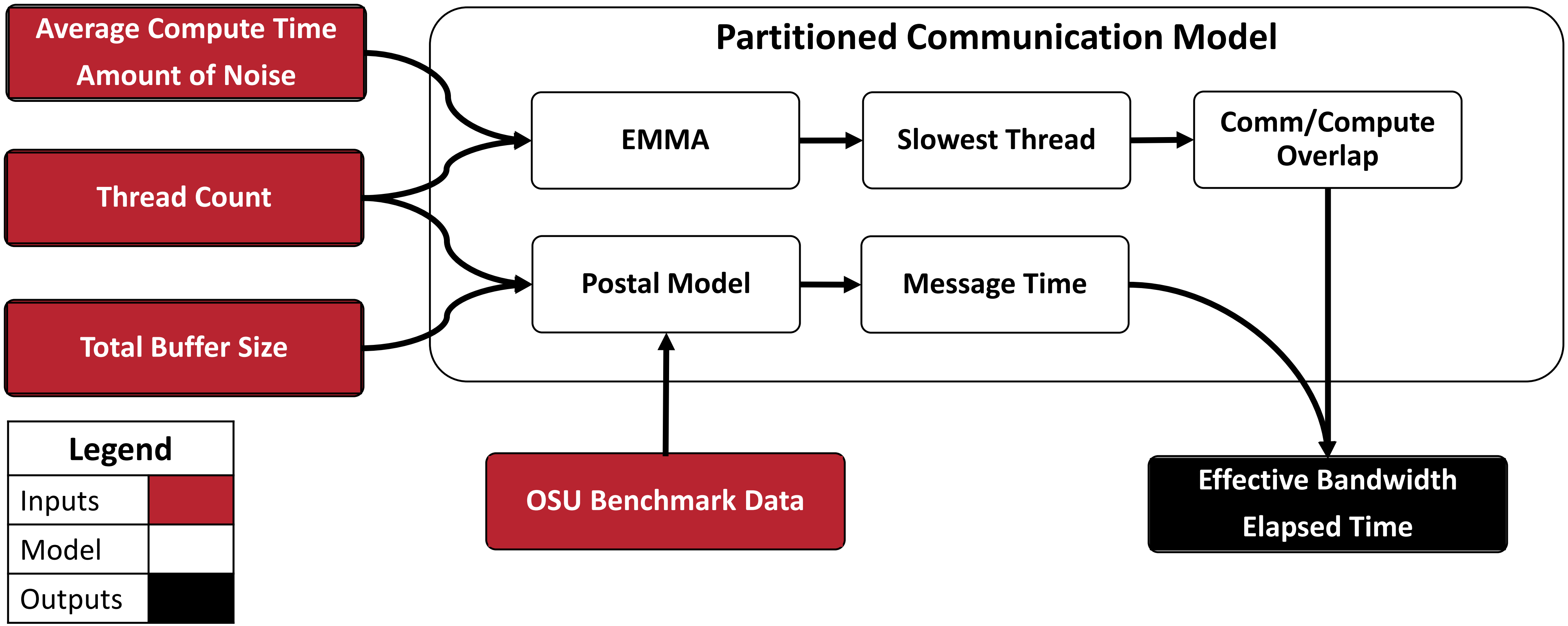
- Load-balanced threads
  - Threads responsible for compute/communication of data partitions of equal size
- Threads individually send partitioned data as single message
- Thread runtimes are distributed normally
  - Allows the simple usage of the expected mean maximum approximation **(EMMA)**
  - $E(max_{i=1}^{m} X_i) \approx F^{-1}(0.57037^{\frac{1}{m}})$, where $F = CDF\ of\ distribution$
- Message transmission times can be described by the postal model
  - **Postal Model:** $T_{comm} = \alpha + \frac{size}{\beta}, \alpha = latency, \beta = bandwidth$

Partitioned Model Specification

- **Further Assumptions:**
  - **Optimistic Sending Assumption:** Data transmission to begin as soon as the fastest thread finishes its compute and will proceed continuously until the slowest thread finishes its compute
  - At least one message will remain unsent at the time that the slowest thread finishes its compute



Partitioned Model Implementation



Partitioned Benchmark Implementation

- Analogous to MPI Ping-Pong Benchmark
- Initializes with warm-up loops
- Send-Side Partitioned Communication with MPIPCL
- Timing and OpenMP reductions to calculate performance

Legend	
Inputs	(red)
Model	(white)
Outputs	(black)

Thread count
Total buffer size
Average compute time
Amount of noise

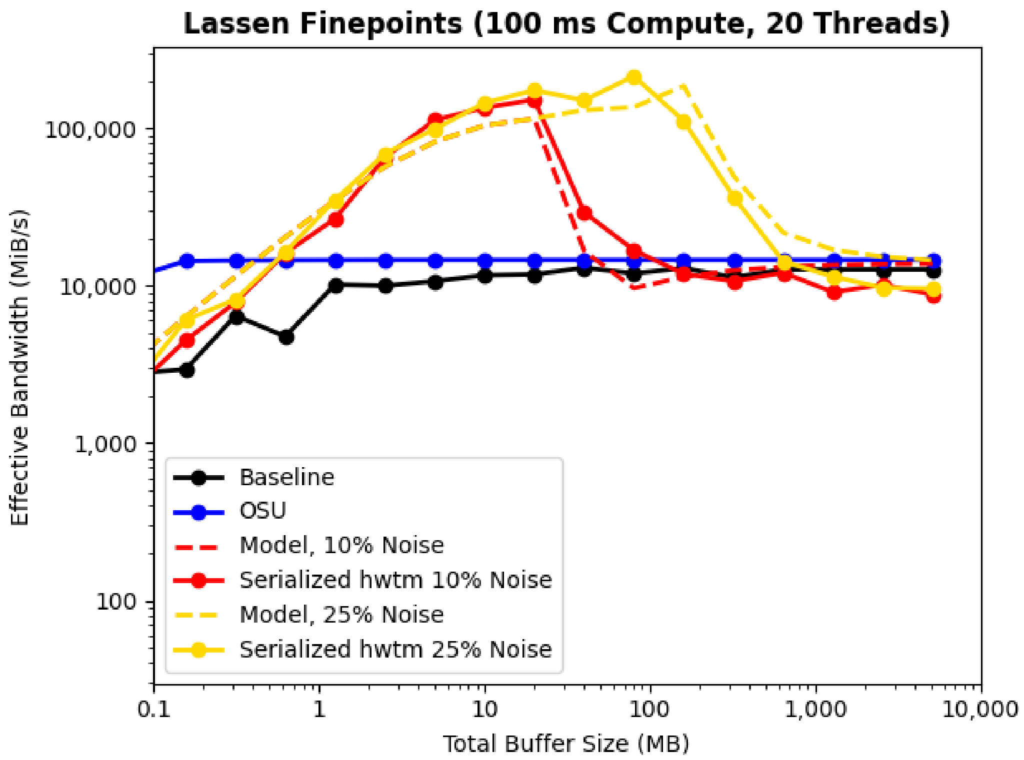
→

Partitioned Communication Benchmark

→

Effective Bandwidth
Elapsed Time

Model Evaluation



- Partitioned Benchmark Performance compared to Model Predicted Performance on Lassen
- Model assumptions investigated by toggling:
  - Async progress thread
  - Hardware tag matching

References and Acknowledgements

References

- Ryan E Grant, Matthew G F Dosanjh, Michael Levenhagen, Ron Brightwell, and Anthony Skjellum. 2019. Finepoints: Partitioned Multithreaded MPI Communication. ISC High Performance Conference (ISC 2019) (2019).

- **Acknowledgements**

  - Ryan Grant and Matthew Dosanjh – For guidance regarding partitioned communication
  - Prof. Purushotham Bangalore, Prof. Anthony Skjellum, and Derek Schafer - For allowing access to MPIPCL, a Partitioned Communication Library
  - Prof. Patrick Bridges and Prof. Amanda Bienz - For technical feedback and support
  - This work was [partially] supported by the U.S. Department of Energy's National Nuclear Security Administration (NNSA) under the Predictive Science Academic Alliance Program (PSAAP-III) Award #DE-NA0003966