Unstructured Communication in AMR Applications

Justin “Tanner” Broaddus

This work was supported in part 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.

Internship Overview

LANL Internship Goal - Establish an understanding of unstructured communication patterns present in AMR applications.

Under the guidance of Galen Shipman - LANL Computer Science

Codebase of interest - LANL's Parthenon

Research questions:

1. How do developers vary in their implementation of unstructured communication in order to perform adaptive mesh refinement (AMR)?
2. What classification of unstructured communication can be derived from inspected AMR codebases?

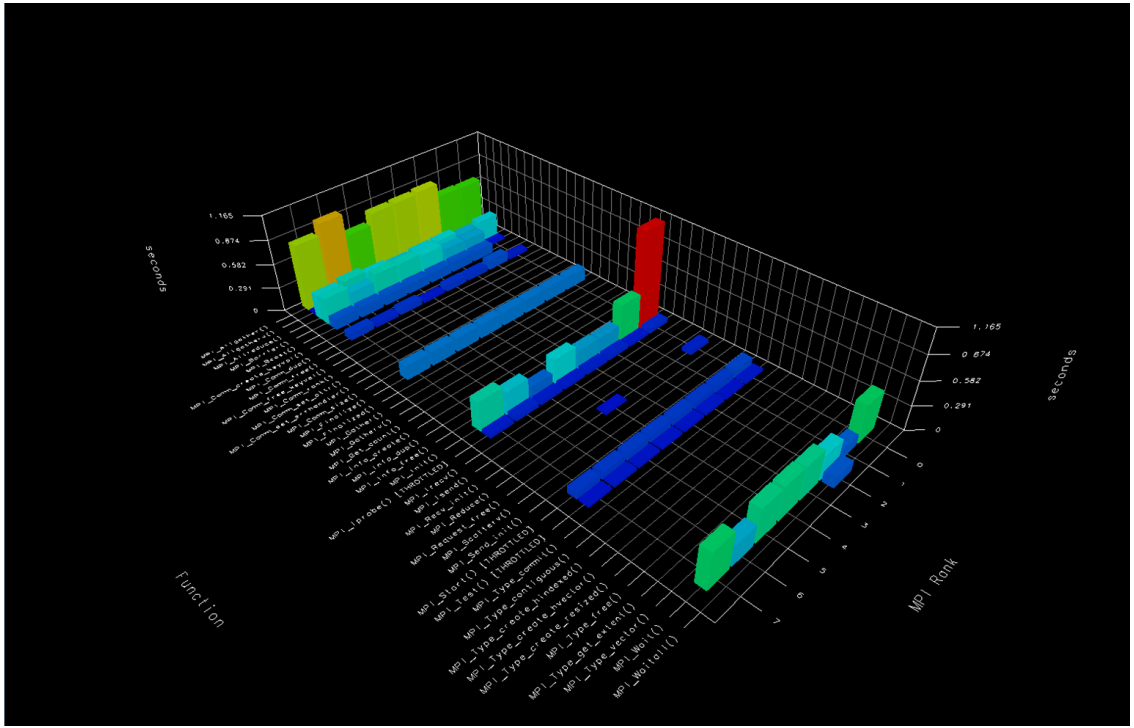
Parthenon

- AMR Infrastructure developed at Los Alamos National Laboratory (LANL), Princeton University, Michigan State University
- Performance portable task-based AMR infrastructure
- Implemented in C++
- Uses Kokkos as the shared memory parallel programming model

Parthenon - Communication Overview

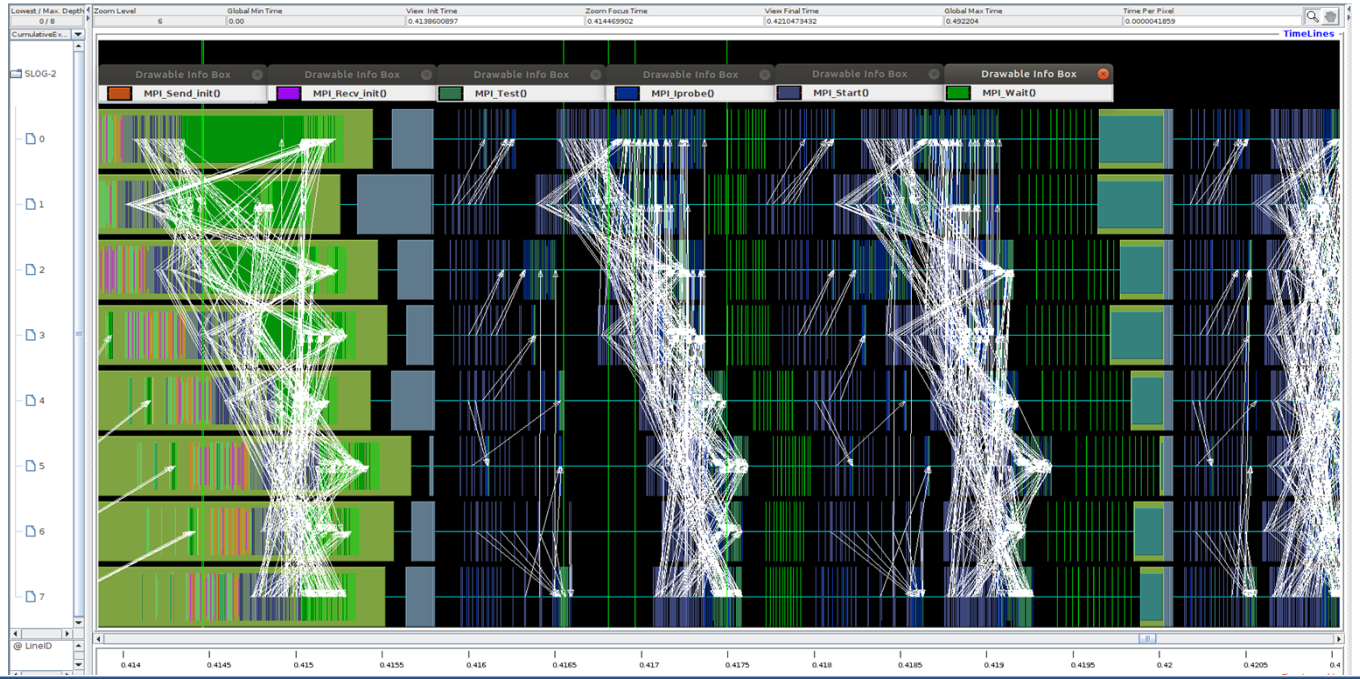
- Halo exchanges
  - Uses persistent communication; initialized by **MPI_Send_Init** and **MPI_Recv_Init**
  - Followed by **MPI_Start** and **MPI_Test**
- Load balancing
  - Considered periodically, load costs communicated by **MPI_Allgather** call
- Mesh block transfers
  - **MPI_Isend** and **MPI_Irecv**; followed by **MPI_Wait**
- Tools for analysis of communication
  - Tau's Paraprof and Jumpshot
  - Caliper and Kokkos annotations

Parthenon - MPI Profiling



Parthenon - Load Balancing Trace

Parthenon - Halo Exchange Trace



CLAMR Overview

- Collection of cell-based AMR mini-apps developed at LANL
- Tests algorithms to be used in heterogeneous computing environments
- OpenCL as the shared memory parallel programming model

CLAMR Observations

- Difference between the min and max times for neighbor boundary updates in MVAPICH2 compiled binaries
- MPI processes stuck in **MPI_Waitall**
- **MPI_Barrier** after **MPI_Waitall** call improves performance for MVAPICH2 compiled binaries
- Issue present L7 function

```
CPU:   state_timer_refine_potential          2.3446          4.2583          4.2754   s min/median/max
CPU:     state_timer_calc_mpot               1.1493          2.9786          3.0291   s min/median/max
CPU:     mesh_timer_refine_smooth            1.0805          1.1493          1.2299   s min/median/max
CPU:   state_timer_rezone_all                3.9106          3.9117          3.9124   s min/median/max
CPU:   mesh_timer_partition                  0.0000          0.0000          0.0000   s min/median/max
CPU:   mesh_timer_calc_neighbors            33.2815         33.2977         33.3073   s min/median/max
CPU:     mesh_timer_hash_setup               7.1314          7.5969          7.9570   s min/median/max
CPU:     mesh_timer_hash_query               3.9666          4.1236          4.3581   s min/median/max
CPU:     mesh_timer_find_boundary            1.4795          1.8370          2.2330   s min/median/max
CPU:     mesh_timer_push_setup               0.6428          0.8085          0.9951   s min/median/max
CPU:     mesh_timer_push_boundary            0.4611          1.0801          9.3026   s min/median/max
CPU:     mesh_timer_local_list               0.0000          0.0000          0.0000   s min/median/max
CPU:     mesh_timer_layer1                   0.5861          2.1061          3.2405   s min/median/max
CPU:     mesh_timer_layer2                   0.3581          1.7464          3.0779   s min/median/max
CPU:     mesh_timer_layer_list               0.0000          0.0000          0.0000   s min/median/max
CPU:     mesh_timer_copy_mesh_data           0.3132          0.3488          0.4402   s min/median/max
CPU:     mesh_timer_fill_mesh_ghost          0.0050          0.0068          0.0080   s min/median/max
CPU:     mesh_timer_fill_neigh_ghost         0.5816          3.3174          3.5235   s min/median/max
CPU:     mesh_timer_set_corner_neigh         0.0000          0.0000          0.0000   s min/median/max
CPU:     mesh_timer_neigh_adjust             0.2602          0.2641          0.2661   s min/median/max
CPU:     mesh_timer_setup_comm               0.7076          9.7836         15.8901   s min/median/max
CPU:   state_timer_mass_sum                  0.0054          0.0295          0.0330   s min/median/max
CPU:   mesh_timer_load_balance               8.9030          9.3173          9.7593   s min/median/max
CPU:   mesh_timer_calc_spatial_coordi        0.0000          0.0000          0.0000   s min/median/max
============================================================
Profiling: Total CPU            time was   62.8369         62.8547         62.8919   s min/median/max
------------------------------------------------------------
Mesh Ops (Neigh+rezone+smooth+balance)   47.2573         47.6461         48.1077   s min/median/max
Mesh Ops Percentage                      75.1697         75.8151         76.5403   percent min/median/max
============================================================
Profiling: Total                time was   64.0156         64.0156         64.0157   s min/median/max
```

CLAMR compiled w/ MVAPICH2

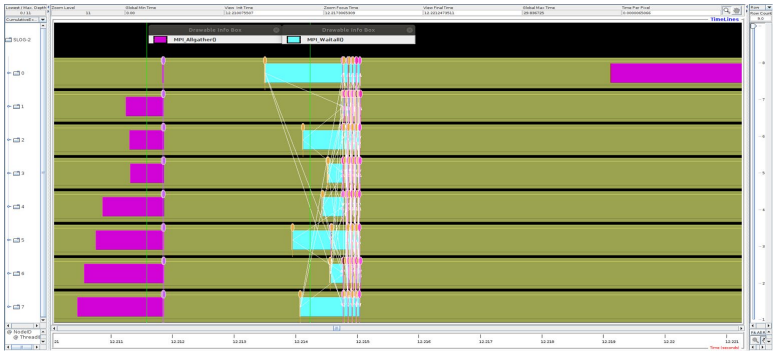
```
CPU:   state_timer_rezone_all                3.9078          3.9091          3.9098   s min/median/max
CPU:   mesh_timer_partition                  0.0000          0.0000          0.0000   s min/median/max
CPU:   mesh_timer_calc_neighbors            25.8390         25.8526         25.8639   s min/median/max
CPU:     mesh_timer_hash_setup               7.1509          7.5878          7.9701   s min/median/max
CPU:     mesh_timer_hash_query               4.0705          4.1575          4.4782   s min/median/max
CPU:     mesh_timer_find_boundary            1.4645          1.8823          2.2533   s min/median/max
CPU:     mesh_timer_push_setup               0.6502          0.8079          1.0069   s min/median/max
CPU:     mesh_timer_push_boundary            0.2565          0.4076          0.5316   s min/median/max
CPU:     mesh_timer_local_list               0.0000          0.0000          0.0000   s min/median/max
CPU:     mesh_timer_layer1                   0.6030          2.2101          3.4201   s min/median/max
CPU:     mesh_timer_layer2                   0.3595          2.0652          3.3769   s min/median/max
CPU:     mesh_timer_layer_list               0.0000          0.0000          0.0000   s min/median/max
CPU:     mesh_timer_copy_mesh_data           0.3334          0.3702          0.4889   s min/median/max
CPU:     mesh_timer_fill_mesh_ghost          0.0050          0.0069          0.0083   s min/median/max
CPU:     mesh_timer_fill_neigh_ghost         0.5799          3.3697          3.4714   s min/median/max
CPU:     mesh_timer_set_corner_neigh         0.0000          0.0000          0.0000   s min/median/max
CPU:     mesh_timer_neigh_adjust             0.2611          0.2660          0.2724   s min/median/max
CPU:     mesh_timer_setup_comm               0.1128          2.3837          8.4010   s min/median/max
CPU:   state_timer_mass_sum                  0.0042          0.0307          0.0342   s min/median/max
CPU:   mesh_timer_load_balance               9.0555          9.3503          9.7628   s min/median/max
CPU:   mesh_timer_calc_spatial_coordi        0.0000          0.0000          0.0000   s min/median/max
============================================================
Profiling: Total CPU            time was   55.5333         55.5521         55.5745   s min/median/max
------------------------------------------------------------
Mesh Ops (Neigh+rezone+smooth+balance)   39.9882         40.2425         40.6864   s min/median/max
Mesh Ops Percentage                      71.9815         72.4353         73.2280   percent min/median/max
```

CLAMR compiled w/ OpenMPI

CLAMR Observations

- Difference between the min and max times for neighbor boundary updates in MVAPICH2 compiled binaries
- MPI processes stuck in **MPI_Waitall**
- **MPI_Barrier** after **MPI_Waitall** call improves performance for MVAPICH2 compiled binaries
- Issue present L7 function

CLAMR compiled w/ MVAPICH2



CLAMR compiled w/ OpenMPI

CLAMR Observations

- Difference between the min and max times for neighbor boundary updates in MVAPICH2 compiled binaries
- MPI processes stuck in **MPI_Waitall**
- **MPI_Barrier** after **MPI_Waitall** call improves performance for MVAPICH2 compiled binaries
- Issue present L7 function

```
CPU:    state_timer_refine_potential                2.3446          4.2583          4.2754   s min/median/max
CPU:      state_timer_calc_mpot                     1.1493          2.9786          3.0291   s min/median/max
CPU:      mesh_timer_refine_smooth                  1.0805          1.1493          1.2299   s min/median/max
CPU:    state_timer_rezone_all                     3.9106          3.9117          3.9124   s min/median/max
CPU:    mesh_timer_partition                       0.0000          0.0000          0.0000   s min/median/max
CPU:    mesh_timer_calc_neighbors                 33.2815         33.2977         33.3073   s min/median/max
CPU:      mesh_timer_hash_setup                     7.1314          7.5969          7.9570   s min/median/max
CPU:      mesh_timer_hash_query                     3.9666          4.1236          4.3581   s min/median/max
CPU:      mesh_timer_find_boundary                  1.4795          1.8370          2.2330   s min/median/max
CPU:      mesh_timer_push_setup                     0.6428          0.8085          0.9951   s min/median/max
CPU:      mesh_timer_push_boundary                  0.4611          1.0801          9.3026   s min/median/max
CPU:      mesh_timer_local_list                     0.0000          0.0000          0.0000   s min/median/max
CPU:      mesh_timer_layer1                         0.5861          2.1061          3.2405   s min/median/max
CPU:      mesh_timer_layer2                         0.3581          1.7464          3.0779   s min/median/max
CPU:      mesh_timer_layer_list                     0.0000          0.0000          0.0000   s min/median/max
CPU:      mesh_timer_copy_mesh_data                 0.3132          0.3488          0.4402   s min/median/max
CPU:      mesh_timer_fill_mesh_ghost                0.0050          0.0068          0.0080   s min/median/max
CPU:      mesh_timer_fill_neigh_ghost               0.5816          3.3174          3.5235   s min/median/max
CPU:      mesh_timer_set_corner_neigh               0.0000          0.0000          0.0000   s min/median/max
CPU:      mesh_timer_neigh_adjust                   0.2602          0.2641          0.2661   s min/median/max
CPU:      mesh_timer_setup_comm                     0.7076          9.7836         15.8901   s min/median/max
CPU:    state_timer_mass_sum                       0.0054          0.0295          0.0330   s min/median/max
CPU:    mesh_timer_load_balance                    8.9030          9.3173          9.7593   s min/median/max
CPU:    mesh_timer_calc_spatial_coordi             0.0000          0.0000          0.0000   s min/median/max
============================================================
Profiling: Total CPU          time was   62.8369         62.8547         62.8919   s min/median/max
------------------------------------------------------------
Mesh Ops (Neigh+rezone+smooth+balance)   47.2573         47.6461         48.1077   s min/median/max
Mesh Ops Percentage                      75.1697         75.8151         76.5403   percent min/median/max
============================================================
Profiling: Total              time was   64.0156         64.0156         64.0157   s min/median/max
```

MVAPICH2 CLAMR w/o MPI_Barrier

```
CPU:    state_timer_rezone_all                     3.9346          3.9356          3.9360   s min/median/max
CPU:    mesh_timer_partition                       0.0000          0.0000          0.0000   s min/median/max
CPU:    mesh_timer_calc_neighbors                 25.9420         25.9515         25.9595   s min/median/max
CPU:      mesh_timer_hash_setup                     7.1185          7.5718          7.9349   s min/median/max
CPU:      mesh_timer_hash_query                     4.0759          4.2055          4.4932   s min/median/max
CPU:      mesh_timer_find_boundary                  1.4763          1.8480          2.2263   s min/median/max
CPU:      mesh_timer_push_setup                     0.6490          0.8009          0.9929   s min/median/max
CPU:      mesh_timer_push_boundary                  0.3218          0.4595          0.5910   s min/median/max
CPU:      mesh_timer_local_list                     0.0000          0.0000          0.0000   s min/median/max
CPU:      mesh_timer_layer1                         0.5893          2.1708          3.3478   s min/median/max
CPU:      mesh_timer_layer2                         0.3654          2.0948          3.4309   s min/median/max
CPU:      mesh_timer_layer_list                     0.0000          0.0000          0.0000   s min/median/max
CPU:      mesh_timer_copy_mesh_data                 0.3374          0.3658          0.4930   s min/median/max
CPU:      mesh_timer_fill_mesh_ghost                0.0050          0.0069          0.0081   s min/median/max
CPU:      mesh_timer_fill_neigh_ghost               0.5944          3.4727          3.6585   s min/median/max
CPU:      mesh_timer_set_corner_neigh               0.0000          0.0000          0.0000   s min/median/max
CPU:      mesh_timer_neigh_adjust                   0.2609          0.2649          0.2676   s min/median/max
CPU:      mesh_timer_setup_comm                     0.0755          2.3068          8.4141   s min/median/max
CPU:    state_timer_mass_sum                       0.0053          0.0290          0.0324   s min/median/max
CPU:    mesh_timer_load_balance                    8.9842          9.3409          9.7475   s min/median/max
CPU:    mesh_timer_calc_spatial_coordi             0.0000          0.0000          0.0000   s min/median/max
============================================================
Profiling: Total CPU          time was   55.5448         55.5582         55.5855   s min/median/max
------------------------------------------------------------
Mesh Ops (Neigh+rezone+smooth+balance)   40.0000         40.3497         40.7863   s min/median/max
Mesh Ops Percentage                      71.9882         72.6276         73.3951   percent min/median/max
============================================================
```

MVAPICH2 CLAMR w/ MPI_Barrier

CLAMR Research Questions

1. How do methods for performing weak progress in MPI implementations affect the performance of an application in different runtime environments?
2. What can be done to optimize the effectiveness of a weak progress engine in order to avoid performance anomalies present amongst different MPI implementations?

Future Work

- Expand the investigation of unstructured communication to different AMR codebases (HIGRAD, xRAGE, CLAMR)
- Begin formulating a classification of unstructured communication
- Begin investigating methods of weak progress
- Assist L7 Benchmark Development (Savannah Camp's work)
- Provide feedback on Dr. Ryan Marshall's reproducibility project

Acknowledgements

- Galen Shipman - LANL Computer Science
- Anthony Skjellum, Patrick Bridges, Amanda Bienz, Puri Bangalore, and Savannah Camp (PSAAP)
- Ryan Marshall - University of Alabama (PSAAP)
- The Parthenon development team (especially Jonah Miller)
- Bob Robey - LANL

References

- CLAMR, https://github.com/lanl/CLAMR
- Parthenon, https://github.com/lanl/parthenon
- Caliper, https://github.com/LLNL/Caliper, https://software.llnl.gov/Caliper/
- Tau, https://www.cs.uoregon.edu/research/tau/home.php