POP CoE

JMI Assessment (POP2_AR_148)

Jonathan Boyle (jonathan.boyle@nag.co.uk), NAG - March 2022

EU H2020 Centre of Excellence (CoE)

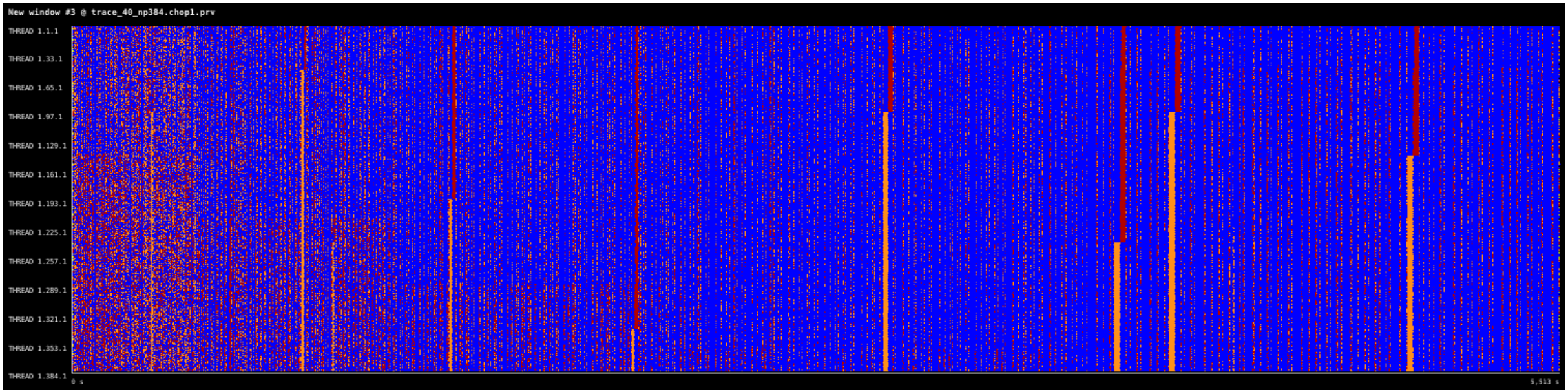
1 December 2018 – 31 May 2022

Grant Agreement No 824080

Background

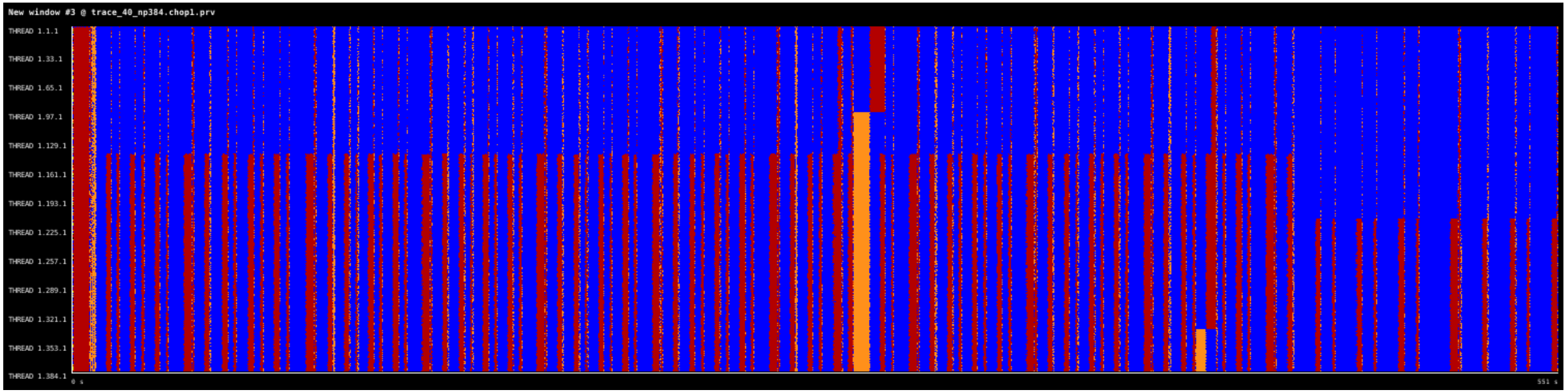
- Applicant: Gerald Eisenberg-Klein (TEEC GmbH)
- Name of the code: JMI
- Scientific/technical area: Earth and atmospheric sciences
- Programming: C,C++ and MPI
- Platform: MareNostrum IV (MN4)
  - 2x 24 core Xeon Platinum 8160 processors per compute node
    - Processor base frequency: 2.10 GHz
- The source code was provided by TEEC GmbH and compiled on MN4
  - GCC 8.1.0, OpenMPI 4.0.1, MKL 2020.1
- POP collected the trace data for test case 40_run_full_demo.sh
  - fmax_lower=10., fmax_upper=30. & niter=30,30,30,30,30
- Scale: 1 - 32 nodes (48 - 1536 cores)
  - **1 - 8 nodes is typical of production runs**
- Tools: Extrae 3.8.3, Paraver 4.9.2, Dimemas 5.4.2 & PyPOP 0.3.3

Focus of Analysis (FOA)



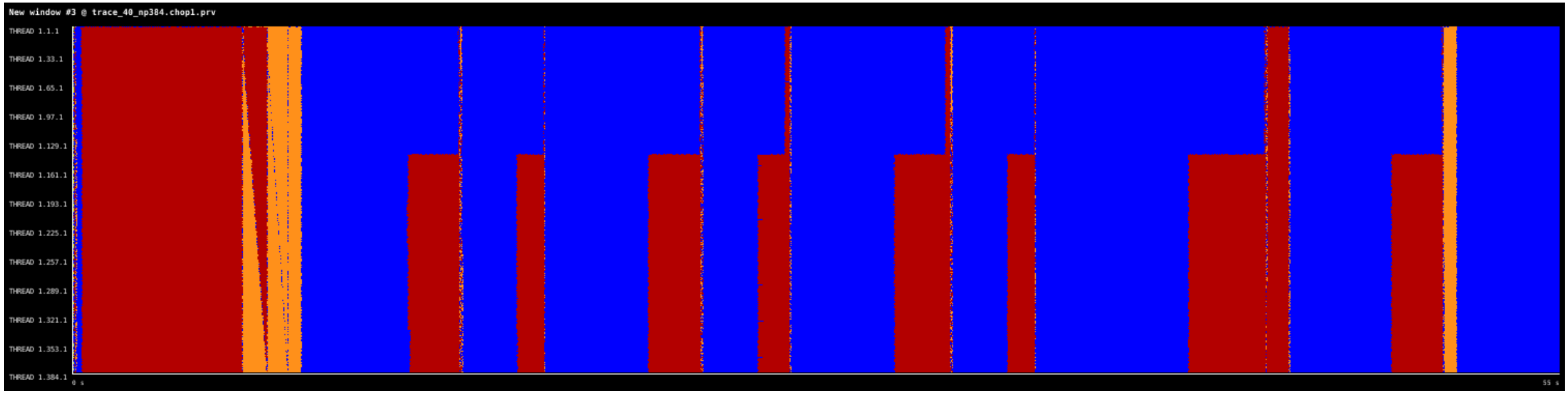
- The FOA was taken to be the whole execution from MPI_Init onwards
- Image shows the trace timeline for **8 compute nodes**
  - Time runs from left to right
  - Each horizonal line represents one process
- Details are unresolved at this scale

Zoom to first 1/10th of FOA



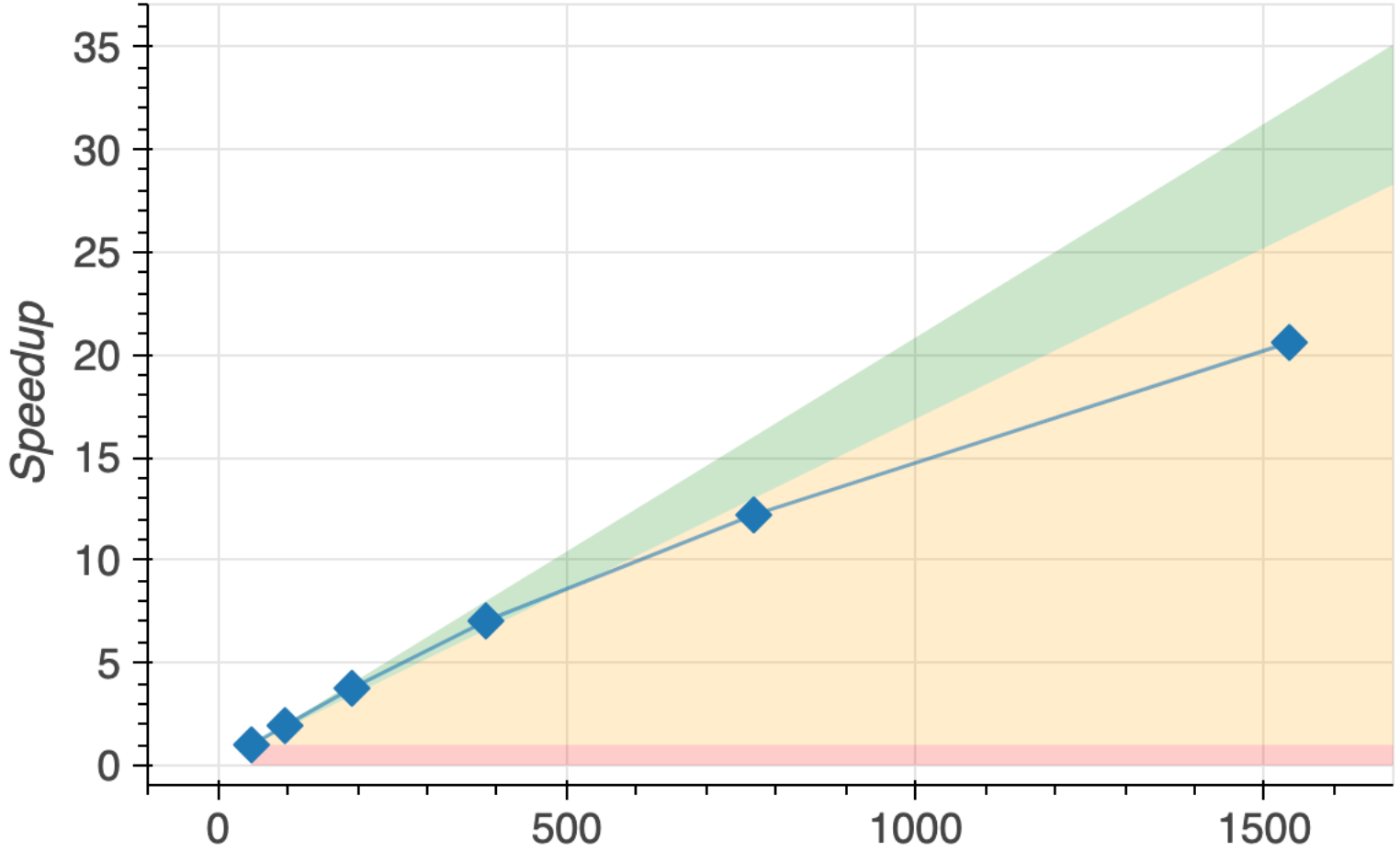
- Note the regions of imbalanced computation ending in MPI_Barrier and some combination of the following collective MPI calls
  - MPI_Bcast
  - MPI_Reduce
  - MPI_Scatter
- This structure is typical of the whole execution, with longer computation in some regions and variable imbalance

Zoom to first 1/100th of FOA



- Although this region shows imbalance, the overall load balance on 8 nodes is good
  - See Load Balance Efficiency on later slide

Speedup versus cores for FOA



- For hardware typical of production runs (up to 384 cores) scaling is good
- For higher core counts, scaling starts to reduce

POP Scaling Metrics

- **Scaling** metrics measure a ‘useful’ quantity relative to the value on 1 compute node
  - ‘Useful’ excludes time in MPI
  - The quantity in question is derived by summing over all processes
  - Values less than 1 indicate a drop in performance
- **Computation Scaling** measures any increase in total useful computation
- This can be split into contributions
  - **Instruction Scaling**
  - **IPC (instructions per cycle) Scaling**
  - **Frequency Scaling**
  - IPC Scaling x Instruction Scaling x Frequency Scaling = Computation Scaling

POP additive efficiency metrics

- **Efficiency** tells us what fraction of our actual execution time would remain after removing a specific bottleneck or set of bottlenecks
  - A value of 1 indicates ideal performance
- **Inefficiency** = 1 - Efficiency
  - This tells us what fraction of our actual execution time would be removed by eliminating a specific bottleneck or set of bottlenecks
- They are additive in the sense a ‘parent’ inefficiency can be split into additive ‘child’ inefficiency values
  - The sum of the child inefficiency values = the parent inefficiency value

POP MPI Efficiency Metrics

- **Parallel Efficiency** = average useful computation / runtime
  - Measures total cost of parallelization
    - For ideal parallelization: runtime = average useful computation
- **Global Efficiency** = Parallel Efficiency x Computation Scaling
  - This measures the combined cost of parallelization plus any increase in useful computation
- Parallel Efficiency is split into:
  - Communication Efficiency
  - Load Balance Efficiency
- **Communication Efficiency** = maximum useful computation / runtime
  - If MPI cost is zero then: runtime = max useful computation
    - i.e. 100% Communication Efficiency
- **Load Balance Eff.** = [Runtime - max comp + avg comp] / Runtime
  - Measures cost of computation imbalance (in absence of MPI synchronizations)

POP MPI Efficiency Metrics

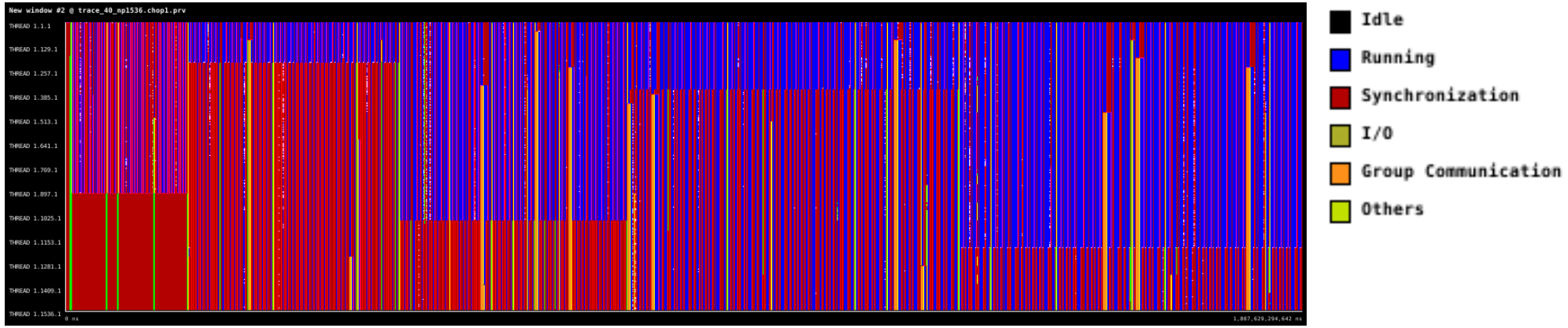
- Communication Efficiency measures total cost of adding MPI
- Communication Efficiency is split into:
  - Transfer Efficiency
  - Serialisation Efficiency
- **Transfer Efficiency** = runtime on ideal network / actual runtime
  - Ideal network has zero latency & infinite bandwidth
    - Requires Dimemas tool to simulate using Extrae traces
  - Measures cost of data transfer over the network
- **Serialisation Efficiency**
  = [Runtime - ideal network runtime + max comp] / Runtime
  - Measures the cost of dependencies introduced by MPI synchronization
    - This time in MPI adds to the runtime even on an ideal network

POP Efficiency Metrics

Number of Processes	48	96	192	384	768	1536
Global Efficiency	0.99	0.96	0.93	0.87	0.75	0.64
↪ Parallel Efficiency	0.99	0.97	0.94	0.88	0.77	0.69
↪ Load Balance Efficiency	0.99	0.98	0.95	0.92	0.83	0.75
↪ Communication Efficiency	1.00	0.99	0.99	0.97	0.94	0.94
↪ Transfer Efficiency	1.00	1.00	0.99	0.97	0.96	0.96
↪ Serialisation Efficiency	1.00	0.99	0.99	1.00	0.99	0.98
↪ Computation Scaling	1.00	0.98	0.98	0.98	0.98	0.92
↪ Instruction Scaling	1.00	1.00	1.00	1.00	1.00	1.00
↪ IPC Scaling	1.00	0.99	0.99	0.99	0.98	0.98
↪ Frequency Scaling	1.00	1.00	1.00	1.00	1.00	0.94

- For hardware typical of production runs (up to 384 cores) metrics are high
- For higher core counts, Load Balance Efficiency reduces & impacts performance
- IPC=2.77 on 48 processes

Load imbalance on 32 nodes



- This timeline ‘draw mode’ overemphasises MPI_Barrier calls (i.e. Synchronization) to help visualise the imbalance
  - Note the variability in load balance over the execution time
- Imbalance in the table on the right is measured on 32 nodes as average / maximum over the processes
  - We see the computational imbalance is caused by an imbalance in the number of useful instructions

	Imbalance
Useful computation	0.72
Useful instructions	0.73
Useful IPC	0.99
Useful frequency	1.0

Summary and Recommendations

- Scaling and efficiency are good for hardware typical of production runs
- For the larger node counts, performance starts to degrade due to computational imbalance
  - This is caused by imbalance in useful instructions
- If it becomes necessary to run this computation on larger node counts, then it is recommended to improve load imbalance

Performance Optimisation and Productivity

A Centre of Excellence in HPC

Contact:

https://www.pop-coe.eu

mailto:pop@bsc.es

@POP_HPC

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 676553 and 824080.