Alya-combustion Audit (POP2_AR_016)

Guillem Ramirez-Miranda and Marta Garcia-Gasulla, BSC

EU H2020 Centre of Excellence (CoE)

1 December 2018 – 30 November 2021

Grant Agreement No 824080

Background

- Applicant: Daniel Mira
- Name of the code: Alya + Cantera
- Scientific/technical area: Combustion CFD
- Programming: Fortran; MPI

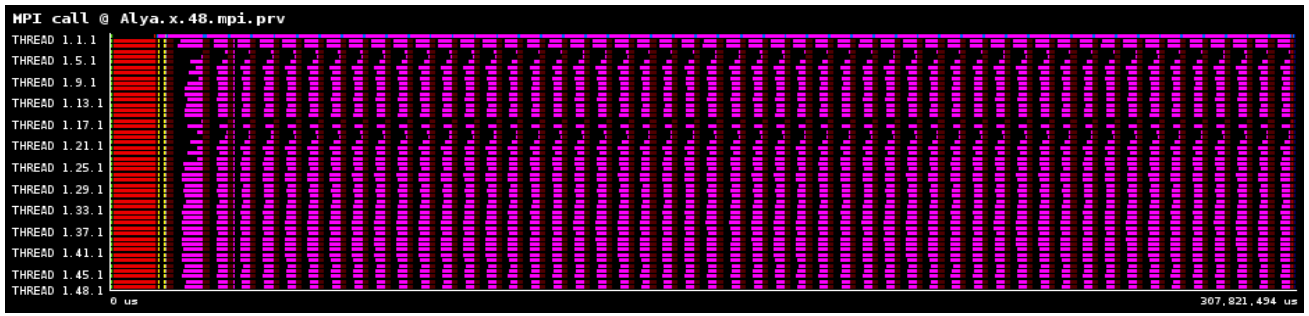
- Input case: methane/air premixed bunsen flame in a confined domain at ambient conditions T= 300K and p=1 bar. The chemical kinetics is based on a GRI3.0 chemical scheme with 53 species and 325 reactions and the problem is solved with a finite rate chemistry model.
- Platform: BSC MareNostrum (2 x Intel Xeon Platinum 8160 24C at 2.1 GHz per node)
- Scale: from 8 MPI ranks to 768 MPI ranks.
- Initial set-up: 50 iterations of simulation. Reduced to 7 iterations
- BSC collected the performance data

Application structure

- First analysis with 50 iterations and 48 MPI

  - Phases clearly identified in the tracefile: initialization and iterative computation.

  - Great density of communications, better focus on less iterations.



Initialization

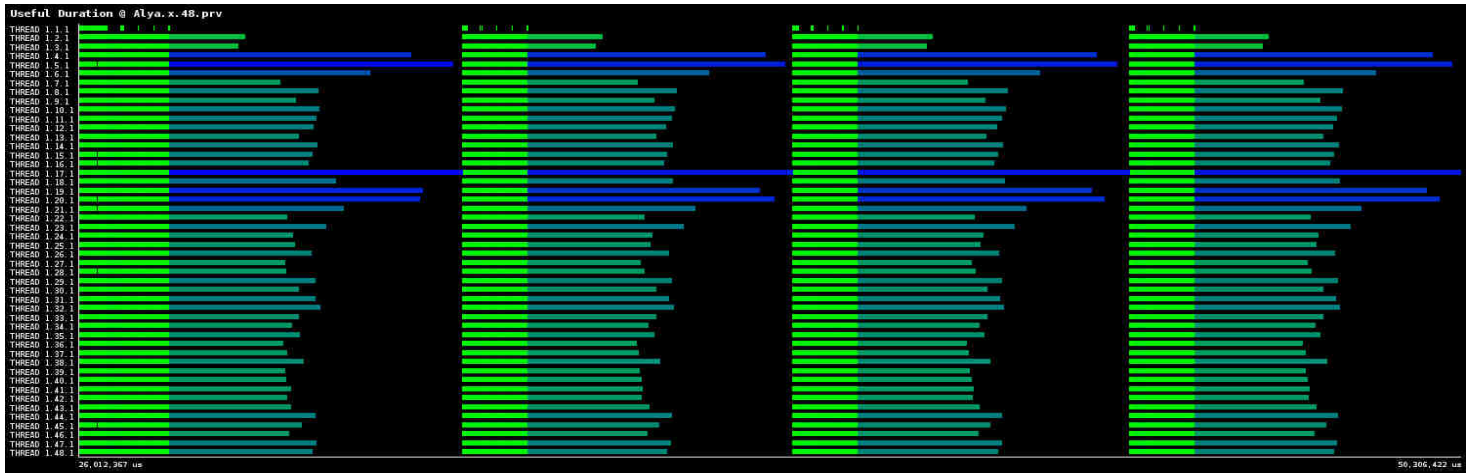
Relevant iterative phase

- The application is structured with a master task that does not perform useful computation, it communicates with all the processes using MPI collectives.

- The Useful Duration view suggest a load balance problem, some processes have more density of useful computation that the rest.

Focus of Analysis (FOA)

Useful duration

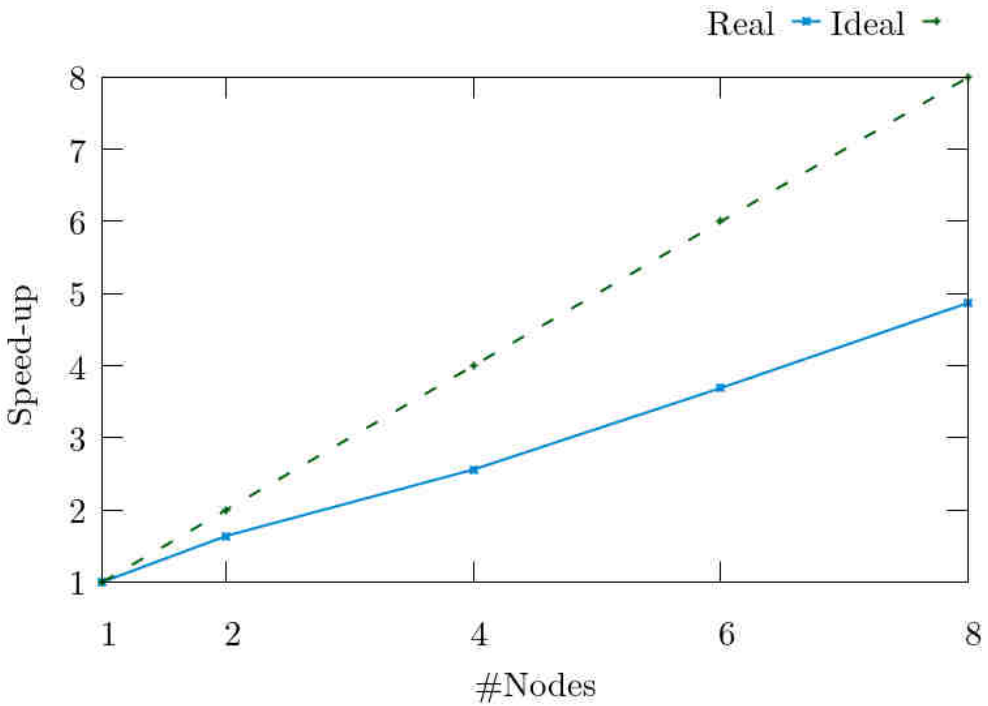


	Useful
Average	50.28%
Minimum	19.31%
Maximum	80.84%

- We see the same pattern in all the iterations of the iterative phase.
- We will use as Focus of Analysis 4 iterations.
- We discard the first iteration as it is slightly longer than the others.
- We observe a high load imbalance in each iteration.

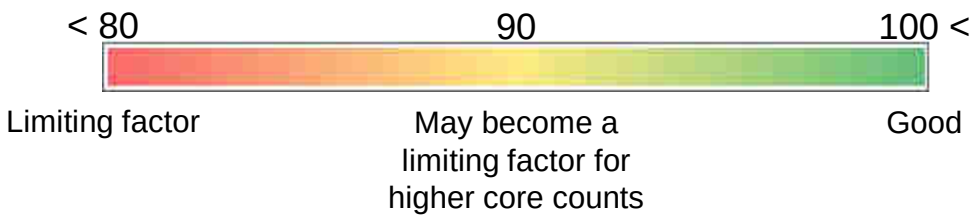
Scalability

- Scalability is far from ideal.
- With 8 nodes speed up is below 5x with respect to using one node.



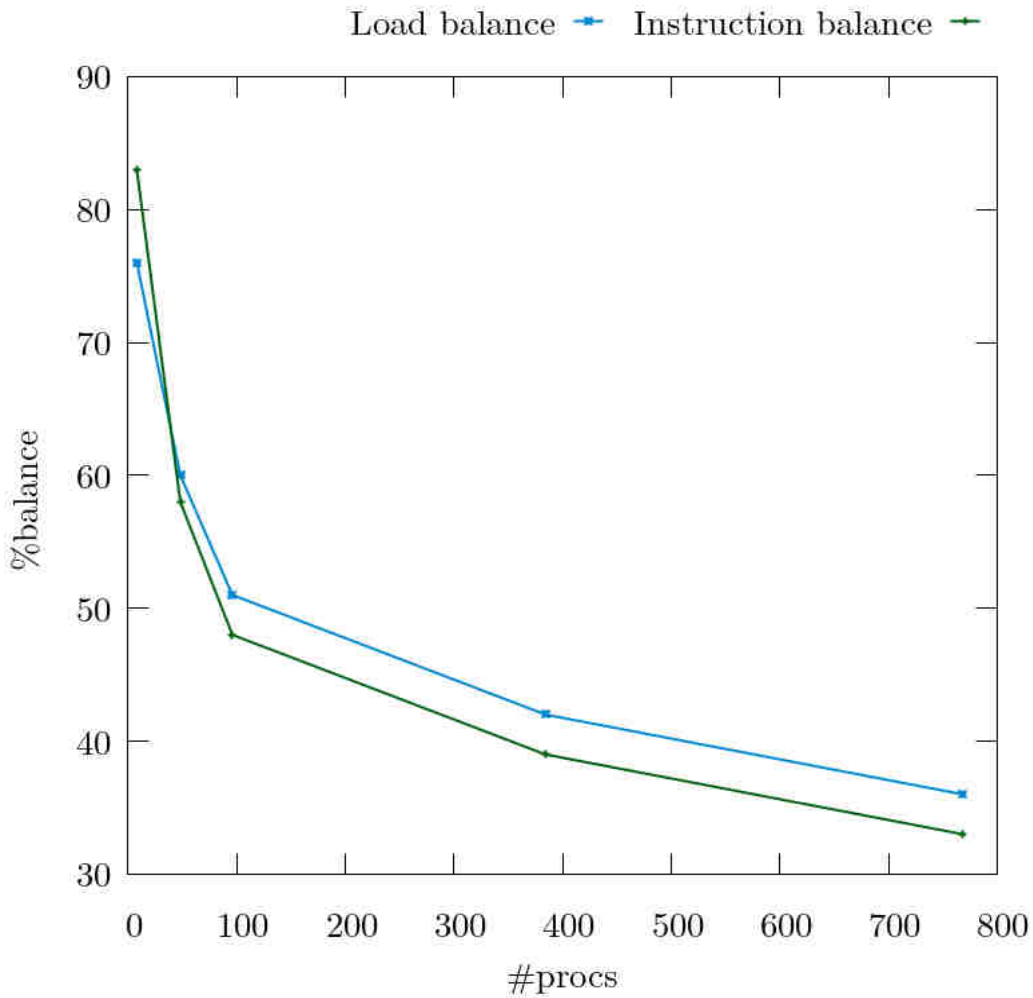
Efficiency model analysis

Number of processes	8	48	96	384	768
└Parallel efficiency	75.63	59.10	49.40	36.14	26.28
└Load balance	75.82	59.99	50.83	41.71	35.75
└Communication efficiency	99.74	98.53	97.18	86.65	73.52
└Serialization efficiency	99.86	99.32	99.13	95.83	89.39
└Transfer efficiency	99.88	99.21	98.03	90.42	82.25
└Computation scalability	100.00	91.37	89.62	83.80	78.30
└IPC scalability	100.00	93.59	93.55	93.36	92.90
└Instruction scalability	100.00	97.61	95.79	89.75	84.27

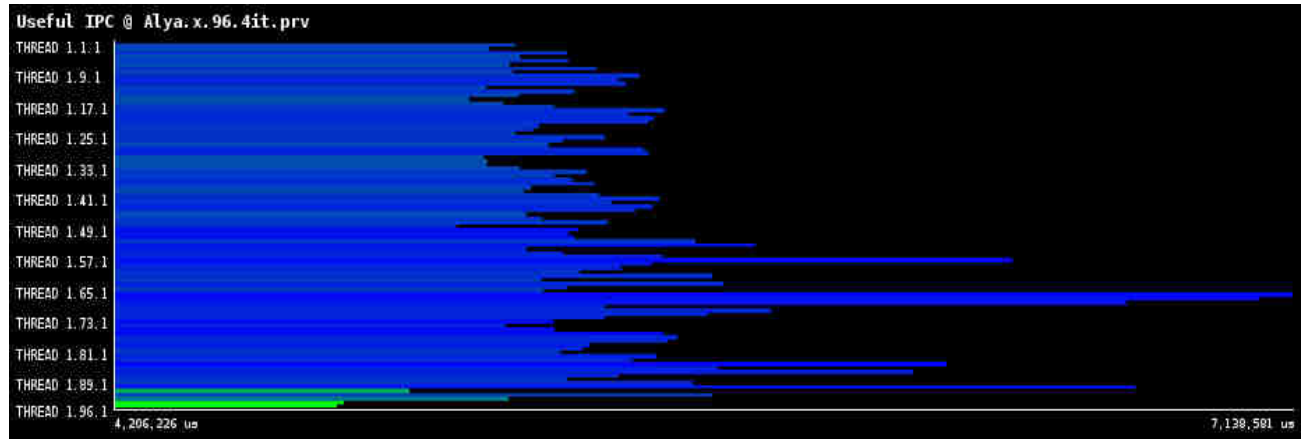


- Efficiencies lower than 80% indicate space for improvement. Lower than 60% there is a clear need for improvement.
- The main factor limiting scalability identified by the efficiencies analysis is **load imbalance**.
- When scaling the number of nodes the transfer inefficiency has a low value
- The IPC of the application is reduced when the node is full of processes.
- The number of useful instructions required to solve the same problem increases substantially when the number of processes increases.

Load balance



- Instructions are the leading cause of the load imbalance.
- This means that the work is not getting distributed between the processors evenly.
- The load balance of the application is better than the instruction balance
  - Processors that have more work do not share node resources → have a better IPC.



Summary of observations

- The main factor affecting efficiency is the load imbalance
  - Load imbalance is algorithmically → Some computation is only performed by some processes
- We suggest to implement a shared memory parallel version of the function causing the load imbalance using OmpSs or OpenMP and finally apply Dynamic Load Balancing (DLB)
  - This can be addressed in a Proof-of-Concept within POP2
- For further analysis on the communication efficiency and instruction scalability we recommend repeating the analysis on a bigger input set.
  - This can be addressed in a Performance Plan

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.