A note on parallel efficiency of fire simulation on cluster

L Valasek, J Glasa
Slovak Academy of Sciences/Institute of Informatics, Dubravska cesta 9, 845 07 Bratislava, Slovakia
E-mail: lukas.valasek@savba.sk

Abstract Current HPC clusters are capable to reduce execution time of parallelized tasks significantly. The paper discusses the use of two selected strategies of cluster computational resources allocation and their impact on parallel efficiency of fire simulation. Simulation of a simple corridor fire scenario by Fire Dynamics Simulator parallelized by the MPI programming model is tested on the HPC cluster at the Institute of Informatics of Slovak Academy of Sciences in Bratislava (Slovakia). The tests confirm that parallelization has a great potential to reduce execution times achieving promising values of parallel efficiency of the simulation, however, the results also show that the use of increasing numbers of computational meshes resulting in increasing numbers of used computational cores does not necessarily decrease the execution time nor the parallel efficiency of simulation. The results obtained indicate that the simulation achieves different values of the execution time and the parallel efficiency in regard of the used strategy for cluster computational resources allocation.

1. Introduction
Current fire simulation systems are capable to utilize the knowledge of the CFD (Computational Fluid Dynamics) theory and model fires and their consequences involving many physical and chemical processes related to fire such as combustion, pyrolysis, heat transfer, thermal radiation, turbulence, fluid dynamics and so on. However, simulation of fires in large areas generally requires carrying out the calculation in parallel on HPC (high performance computing) systems.

The FDS (Fire Dynamics Simulator) system [1] is an advanced CFD-based fire simulator intended for fires in various environments. It was developed by NIST (National Institute of Standards and Technology, U. S. Department of Commerce) in cooperation with VTT (Technical Research Centre of Finland). FDS numerically solves a form of Navier-Stokes equations for low-speed fire induced flows with emphasis on transport of smoke and heat from fire. The system is capable to utilize various types of parallelization. To take advantage of computational resources available, FDS supports four programming models: the sequential model designed for sequential computers, the parallel MPI (Message Passing Interface) model designed for systems with distributed memory, the multithreading OpenMP (Open Multi-Processing) model designed for systems with shared memory and the hybrid MPI&OpenMP model designed for systems with distributed shared memory. In previous research we investigated applicability of FDS to simulate fires in various environments [2-4].

In this paper, we focus on the MPI model of parallelization of fire simulation on the HPC cluster at the Institute of Informatics of Slovak Academy of Sciences in Bratislava (Slovakia).

1 To whom any correspondence should be addressed.
There were 52 IBM dx360 M3 computational nodes available for testing purposes; each node consisted of 2 sockets with 6-core processors Intel E5645, 2.4GHz (12 cores per node). Within each node, 48 GB RAM was available (24 GB of RAM per processor). In the MPI model, the computational domain is divided into several computational meshes and the computation on each mesh is considered as a single MPI process assigned to a single computational core. The communication between MPI processes is provided by MPI. We use the 64-bit MPI version 6.3.2 of FDS for Linux and the open-source Open MPI version 1.10.0 of MPI. The script mpirun (a part of the Open MPI) allows determining exactly how the MPI processes representing a task will be mapped by individual computational nodes and then bound to individual sockets or cores. The aim of this paper is to compare the impact of 2 selected computational resources allocation strategies on parallel efficiency of a corridor fire simulation.

2. Simulation of a corridor fire
We consider a 10-second fire of a corridor with the dimensions of 7.2 m x 1.2 m x 2.7 m and an 100 kW rectangular fire source with the dimensions of 0.1 m x 1 m located about 1 mm above the floor at the 3.64 m distance from the left end of the corridor (figure 1, left). Its boundaries are 20 cm far from the corridor walls. The computational domain with the dimensions of the corridor is divided into 2916000 cube cells forming the computational mesh of the 2 cm resolution (360 cells in the x-direction, 60 cells in the y-direction and 135 in the z-direction). The corridor walls and the corridor ends are represented by INERT and OPEN boundary conditions on the domain borders. The fire source is represented by the corresponding VENT with the 1000 kW/m² HRRPUA (heat release rate per unit area) boundary condition. We assume the ambient temperature and initial temperature inside the corridor being equal to 20°C, the default dominant chemical reaction and the use of the LES (Large Eddy Simulation) method in regard of the corridor dimensions and mesh resolution. The chosen computational mesh fulfils the conditions required for usage of FFT for solving the pressure equation. The mesh sensitivity study realized (figure 1, middle) shows that the chosen 2 cm mesh corresponds to "fine mesh" in the sense of [5] already since the 2 s to the fire. The simulation (denoted by 1M) was realized sequentially on a single computational core of the HPC cluster with the execution time of 17.45 hours. The simulated fire behaviour was as follows (figure 1). At first, smoke spread from the fire source upwards to the corridor ceiling. At the 2 s, the fire intensity reached the value of 100 kW and then fluctuated around the value (figure 1, right). At the 3 s to the fire, smoke hit the ceiling and began to spread under the ceiling in all directions from the hit point at the ceiling. At the 6th s, smoke reached both ends of the corridor.

Figure 1. The course of fire at the 3 s and 6 s to its start of fire (left); mesh sensitivity study illustrated by the D*/dx curves corresponding to the 1 cm, 2 cm and 3 cm mesh resolutions, where D* is the characteristic fire diameter, dx is the nominal size of a mesh cell and t is time, and the horizontal dashed line represents the values corresponding to “fine mesh” in sense of [5] (middle); and the heat release rate (HRR) curve of fire (right)

3. Parallelization of the simulation and its realization
We consider a decomposition of the computational domain in x-direction onto m subdomains of the same size and 9 parallel simulations mM, where m = 2, 3, ..., 6, 8, 9, 10, 12 (figure 2). The
decomposition for \( m = 7 \) and \( m = 11 \) does not fulfil the conditions required for solving a Poisson equation using FFT, therefore, we do not consider the simulations 7M and 11M. The tested simulations have the same 2 cm resolution with the total number of cells equal to 2916000 (2916000/m cells per individual mesh). Since the chosen mesh resolution is the same as in the sequential calculation and we simulate the same fire scenario, the mesh sensitivity study performed for the sequential calculation and analysis of the HRR curves of executed individual parallel simulations showed that we can suppose that also the computational meshes in parallel simulations correspond with “fine mesh” in the sense of the study [5].

In Table 1, the values of execution times (in hours, rows 1 and 2) and parallel efficiency (in %, rows 3 and 4) of the considered simulations carried out with the CC and SS strategies, and their differences (in %, row 5) are shown (see figure 3). The parallel efficiency was calculated according to the number of computational meshes. The comparison of values of the parallel efficiency of simulations carried out with the CC and SS strategies shows that the tests do not confirm the original expectation that increasing number of computational meshes (or used computational cores) decreases the parallel efficiency of simulations (see the increase of the parallel efficiency of the couples of simulations 2M, 3M; 4M, 5M and 8M, 9M carried out with the CC strategy and the couples of simulations 2M, 3M and 4M, 5M carried out with the SS strategy).

Figure 3 also indicates that the values of parallel efficiency of the simulations realized by the CC strategy are smaller than the ones realized by the SS strategy (except the case of \( m = 12 \)). Their difference varies from -15.33 to -4.12 (in the case of \( m = 12 \) it equals to 0.10). Thus, it can be concluded that the simulations carried out with the CC strategy reach substantially greater execution
times and substantially smaller parallel efficiencies as compared with the ones realized by the SS strategy.

![Figure 3](image)

Figure 3. Execution times $t_m$ (left) and parallel efficiency $E_m$ of the simulations $mM$ carried out with the CC and SS strategies, where $m$ means the number of computational meshes. The dashed line represents values of ideal parallel efficiency of simulation.

4. Conclusion

In this paper, the effect of the 2 strategies for allocation of the MPI processes representing the given simulation carried out on computer cluster to individual cores or sockets on parallel efficiency of fire simulation was examined. The series of one sequential and 9 parallel simulations of simple corridor fire scenario was realized on the HPC cluster at the Institute of Informatics of Slovak Academy of Sciences in Bratislava using the parallel MPI programming model of FDS and the strategies --map-by core, --bind-to core and --map-by socket, --bind-to socket. The parallel efficiencies of parallel simulations were compared. The tests confirm a great potential of parallelization to reduce execution times and achieve promising values of parallel efficiency of simulation. The results indicate that the simulations carried out with the CC strategy reach markedly smaller values of parallel efficiency as compared with the simulations carried out with the SS strategy. Analysis of the simulation results also shows that parallelization of the simulation does not necessarily decrease the execution time nor the parallel efficiency of simulation with increasing number of computational meshes. The tests indicate that the problem of efficient parallel realization of fire simulation on computer clusters using FDS and the MPI programming model must be carefully considered.

Acknowledgement

This paper was supported by the Slovak Science Foundation VEGA (project No. 2/0184/14).

5. References

[1] McGrattan K, Hostikka S, McDermott R, Floyd J, Weinschenk C and Overholt K 2015 Fire Dynamics Simulator (Version 6): User’s Guide, NIST Special Publication 1019-6, NIST, Washington

[2] Weisenpacher P, Glasa J, Halada L, Valasek L and Sipkova V 2014 Computing and Informatics 33 6 1237-1268

[3] Glasa J, Valasek L, Halada L and Weisenpacher P 2014 J. Phys.: Con. Ser. 490 012067 DOI: 10.1088/1742-6596/490/1/012067

[4] Valasek L 2015 J. Phys.: Con. Ser. 633 012083 DOI: 10.1088/1742-6596/633/1/012083

[5] Hill K, Dreisbach J, Joglar F, Najafi B, McGrattan K, Peacock R and Hamins A 2007 Verification and Validation of Selected Fire Models for Nuclear Power Plant Applications, NUREG 1824. United States Nuclear Regulatory Commission, Washington, DC.

[6] Documentation of OpenMPI, v1.10.0, Aug. 25, 2015. [https://www.open-mpi.org/doc/v1.10/](https://www.open-mpi.org/doc/v1.10/).