

A Distributed Multi-Level Simulation Tool for Tunnel Risk Assessment and Cost Optimization

G. Pipelidis*, C. Forster, B. Kohl

ILF Consulting Engineers, Linz, Austria

- To whom correspondence should be addressed: Georgios Pipelidis, Georgios.Pipelidis@ilf.com

Abstract

In this paper, it is presented a workflow management system and a web-based cluster front-end developed for scheduling simulations. It is described how the workflow, generated via a web interface, and is integrated from the web server into the cluster. User interaction was achieved on multiple stages. The user can interact with the system by specifying the input parameters the simulations should vary on. One can add, start and stop the simulation on the cluster, upload or create scripts and parse the resulting data to the server location. The user is also able to execute existing executable scripts locate on the cluster. The process of every sub-simulation can be followed heterogeneously with a maximum of one minute delay. How the core allocation decision is being taken by the scheduler itself, which follows an agent-based protocol inspired by the FIPA-Contract-Net protocol of communication. Thus, the scheduler itself always inspect the level of coverage of the preconditions each of the jobs has to meet.

It is also presented the strategy that was followed for integrating different simulating models that were developed for mesh generation, traffic simulation, airflow simulation, smoke propagation simulation, ventilation strategy simulation and evacuation simulation to estimate the consequences of a fire in a tunnel and finally approximate its risk.

Keywords: Data Acquisition System, Heterogeneous Computing, Human Computer Interaction, Parallel and Distributed Processing, Risk Analysis, Workflow Manager

1 Introduction

Moor's law has become the central driving force of the world's industries [4] with numerous of industrial fields to get benefit of this vast development [5] among them the civil, structural and environmental field of engineering, with many examples have been hosted by the CC conferenced. The significance of its existence together with the consequences of its deceleration to be mentioned by M.M. Resch [8] and the advantages and the opportunities to be highlighted by W. Gentsch and B. Yenier [9].

"ILF Consulting Engineers" has been taking advantage of the rapidly increasing processing power which is one among the reasons that has been emerged to one of the European leading companies on risk analysis in road and rail tunnels for almost one decade. Furthermore, ILF has also developed the Austrian Tunnel Risk Analysis Methodology TuRisMo [6] as the Austrian response in the European 2004/54/EC directive [7].

TuRisMo has been developed able to compare different systems, including all tunnel influencing factors, with goal of calculating the statistically expected risk in a reliable and robust way. Due to the large number of factors and the high number of sub-models TuRisMo is consisted of, in addition to the complex computational workflow which the user has to deal with, the need of developing a workflow manager able of integrating all the sub-models as well as able of interacting with every core of our cluster was emerged. Our cluster consists of 25 nodes with 4 cores each.

Examples of integrating existing simulation tools we can be found in CC. Patzak [10, 11] presented a tool supporting the integration of existing applications. His motivation is to enable further progress of existing knowledge, in various disciplines, that has been integrated into advanced simulation tools. The goal of his framework is to provide data exchange between codes. Examples of enriched interfaces for user to the cluster interaction can be seen on some of the use cases of W. Gentsch and B. Yenier [9] and also from A. Boilley [12]. Finally, G. Bui and G. Meschke [15] has presented a simulation for tunnel as well although this is for its excavation.

This paper consist of three main chapters. The first chapter presents the developed system which enables the interaction with the cluster. The second chapter presents TuRisMo and it fundamental ingredients. Finally, follows the discussion which contains the conclusion and the future work.

2 System Design and Features

The presented framework acts as a middleware over the cluster, as a single system image (SSI), allowing the user of considering the cluster as a cohesive computing unit. It enables to define workflows and job dependencies by an ease of use HTML5 graphical user interface. It is able for automatic submission of executions, it provides the user with interfaces to monitor the executions and allows to define priorities and queues to control the execution order of unrelated jobs. The software is capable of deactivating or activating nodes according to the estimated workload

and automatically restart jobs in case of failures. It has been developed as a web application and it provides an interface to the cluster.

It consists of two main modules, the workflow manager (WFM), responsible for managing the workflow and the communication with the cluster and the Job Management System (JMS) which guides the user for submitting his jobs.

Both modules communicate with the database. The first for intervene the status of the cluster and the second for visualizing the information.

2.1 User module and safety

The User module is responsible for guiding the user, via a wizard-like mechanism, or a recommender mechanism [2], since it provides always with feedback regarding the user's input, to develop his simulation definition file. The mechanism is responsible for crosschecking interdependences that the parameters may have. For example the first tunnel cross-section cannot be located in a distance larger than the tunnel length or the simulation time cannot be shorter than the time it takes for the fire to be fully developed.

It is written in HTML5 technologies in order to make the content accessible by any desktop or mobile device, with no installation or compilation necessary on the target device. HTML5 also to offer a better user experience with a richer design.

An interesting approach for addressing safety issues presented by L. Sellami, D. Idoughi and P.F. Tiako [13], they suggested a framework capable of user identification via detecting the user instructions. In our case user is identified via IP Range Based Authentication and security has been achieved by allowing only IP addresses belong to the internal company network. This way, a high safety level has been achieved in an easy way and user gains access on the system after eliminating the need for user IDs and passwords. This way the overall process has been converted to an intuitive for the user procedure. Users whose access is restricted based on an access rule will get a 403 error on their browser.

2.2 Job module

The Job module provides all the required functionality for adding more nodes into the cluster, as well as to submit new jobs on the cluster, start and stop simulations and generate the necessary pipelines each of the submitted simulation requires.

2.2.1 Dashboard

Interestingly the aeromines dashboard, presented by A. Boilley [12], is capable of visualizing by animating the current simulating state. Since the presented framework is designed for simulating numerous of different cases in parallel, the dashboard provides to the user a visualization of the current status of the simulation using colour coded process bars signifying whether a sub-simulation is running, queueing, finished or has been failed (for more details check: 2.2.2.1), while their proportion of the whole process is described by the length of the bar. It also summarizes

information about the nodes each of the simulations occupies, the job each of the sub-simulation consists of together with information it can be identified (e.g.: The Process ID and the processor core it began).

2.2.2 Job Management

The job management system allows the user to upload his script on the server and execute it on the cluster as well as to download its final results from the server. It also allows the user to specify the maximum number of nodes the simulation he desires to employ. It also enables the user to add nodes on the cluster in an intuitive way, simplifies with this way the scalability possibilities of the cluster. Enables the user to submit jobs and monitors them in real time.

2.2.2.1 Monitoring the jobs

The job monitoring module is responsible for inspecting, in the end of every cycle, whether a connection with each node has been established and the duration of this connection. This information is used for assessing if necessary a local crosscheck for re-examining the accuracy of the information, in case that the process has exceeded the execution time allocated by the user. For this purpose the job monitoring module is responsible for being always updated about the elapsed execution time of every individual job. It is also responsible to inspect if the job is queueing based on the current queue situation. Every job is identified by its Project Name, type and its sequence ID.

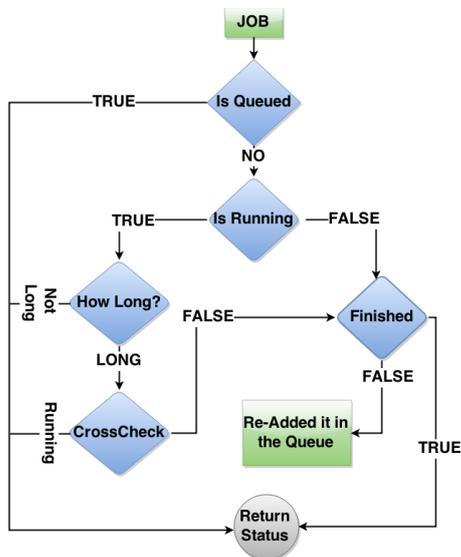


Figure 1: Monitoring the job status

Monitoring the job status: First examines if it is queueing, because it takes the least resources. If not, it checks if it is running. If it is running it examines how long. The boundaries between long and short duration are also depending on the type of simulation (e.g.: If CFD 3 hours would be short). If it is running for longer duration that it has specified, a script will execute on the current node and by using the PID will crosscheck if it is indeed running. If it is not running it will check if it has finished and if not, then it will re-added the job in the queue.

2.2.2.2 Process interaction

Three different possibilities of process interaction are possible. Implicit model of interaction in which the information is being transferred throughout the functions without any storage necessary.

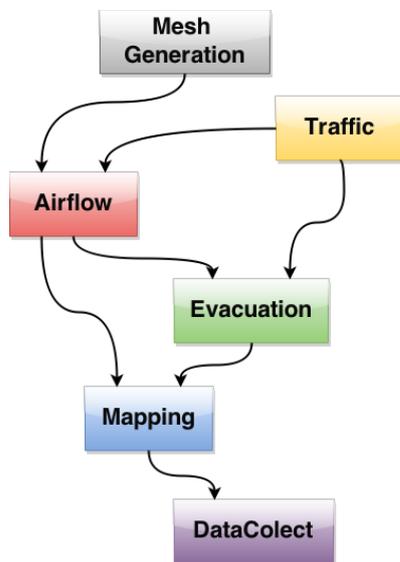
Second is via message passing, in this case parallel tasks exchange data through passing messages. This implies that interconnected sub-models are always running in parallel which does not hold on the presented case.

The last possibility for process interaction is via a shared memory. Shared memory, in our case, is the most efficient way of passing data between programs as it is essential for the integrated models being able of asynchronously read and write data from a global address. It is a frequent phenomenon for interdependent simulations to have difference in days on their start times.

2.2.3 Workflow Management

Workflow Manager, after guides the user to create the simulation definition file, it uses it to generate the necessary pipeline. The pipeline is a collection of simulation definition files that vary based on the boundaries that the user has specified via a wizard-like interface. Parameters like different traffic directions and lines, hazard locations, tunnel closure times, fire start times, different delay times, operating ventilation start time, wind velocities, failures that can occur, like ventilation failure, when ventilation is simulated as broken and time the evacuation process will begin are some among others of the parameters the workflow management will take into consideration in order to generate the corresponding workflow.

The workflow Manager decides whether certain jobs of the pipeline are allowed to run in parallel. The decision is taken based on the interdependences are visualized in Figure 2. Every job that belongs on the same module is allowed to run in parallel, otherwise its precondition has to be covered.



The interdependences of the modules: The traffic and the mesh generation don't have any precondition. The mesh will be generated only once for each tunnel. Airflow simulation preconditions are Traffic (due to piston effect) and Mesh Generation. Evacuation uses as an input the traffic simulation outcome (possible victims) together with the airflow outcome (visibility). Mapping takes as an input evacuation together with airflow results. Datacollect, which is responsible for coupling the results, executes one time over the whole simulation after everything has been finished.

Figure 2: The interdependences of the modules

2.2.3.1 Job initiator Module

After it has been decided which job is going to be executed next, it is time for the job initiator. The job initiator is responsible for finding available resources and allocate them. Its functionality is based on the Contract-net protocol of communication and it is visualized in the Figure 3.

Each core of a node has been taken into consideration as an individual agent. This enables to view the task of job distribution as a kind of contract negotiation that happens in four stages, in contradiction to the contract net where it is done in five steps:

Status recognition: First the system will compute the resource availability. If the status of resources is not explicitly given, a script will be locally executed and update the status in the dashboard as it explained in 2.2.1.

Status declaration: On this step the status of each core will be explicitly specified. If there is a task already allocated on the particular core the return message will contain the sequence id, designed to identify the task, the project name where the task belongs to, the type of simulation and the PID, as it is explained on chapter 2.2.2.1.

Job announcement: The job initiator will explicitly specify the job to the node.

Job status: Currently there is not a system implemented whether a node should accept or reject a job. Hence the possibilities finally are limited to two. The job will be either started or will be failed to start.

The job initiator is equipped with local administrator passwords of the nodes for use whenever these are necessary. The number of jobs allowed to execute has been defined by the user.

Job initiation protocol: First the Job initiator module checks for the status of the current node. If necessary as it mentioned in 2.2.2 even locally. If the node is not disabled / stopped by the user and is not occupied by another job then the job initiator module will send a job from the pipeline on the node. The job can either start or fail.

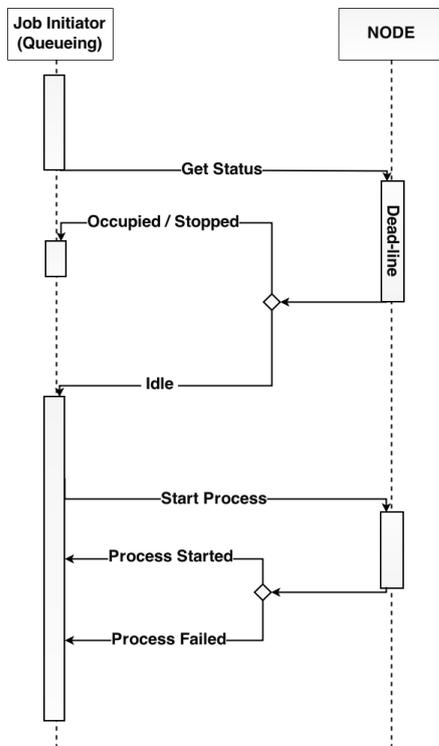


Figure 3: Job initiation protocol.

2.2.4 Cluster Configuration

The system is a multiprocessor system and consists of 25 i5 processors, it is visualized on Figure 4. The nodes are connected with a crossbar switch network. The hosts are sharing a common database to the server where all the information is stored and extracted whenever is necessary. The OS on the server is Windows server 2012 and on each client runs on Windows 7. It follows a Master/Agent architecture

where the job scheduler is responsible for every interaction from the master to the agents.

The communication is done via PsExec. It is a light-weight telnet-replacement that enables the execution of processes on other systems following TCP/IP protocol. Complete with full interactivity for console applications, without having to manually install any client software.

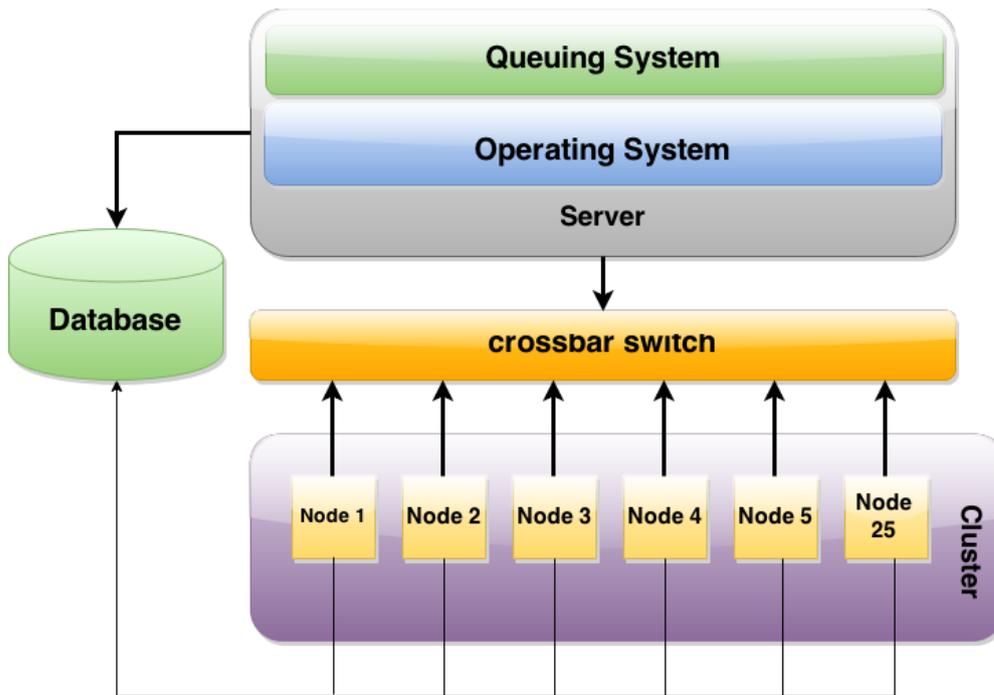


Figure 4: Cluster Configuration

3.1 Problem decomposition

The presented framework is focused on parallel threads of execution. As it is mentioned in the introduction the main problem is to integrate different models designed for different functionalities. Hence, according to Flynn's taxonomy [16], the technique employed to achieve parallelism belongs on Multiple Program Multiple Data technique (MPMD). This allows the system to occupy different executable programs for different task, some written in different programming language. The cores are communicating via shared memory model as it is mentioned in chapter 2.2.2.2.

The Figure 5 presents an example of parallel execution, on the cores of the same node, of different threads in different time steps. An example which is visualised on Figure 5 is the following: on the 3rd core for time t an evacuation simulation will be executed. On the same core for time $t + 1$ an airflow simulation will be executed, while the corresponding mapping simulation, on the evacuation that has been executed on the same core, will be executed on the 1st core for time $t + 2$. It is necessary also to be mentioned that it is not possible for a simulation to be started

without its precondition to be covered. The protocol which specifies the interdependences which can be found on the chapter 2.2.1.

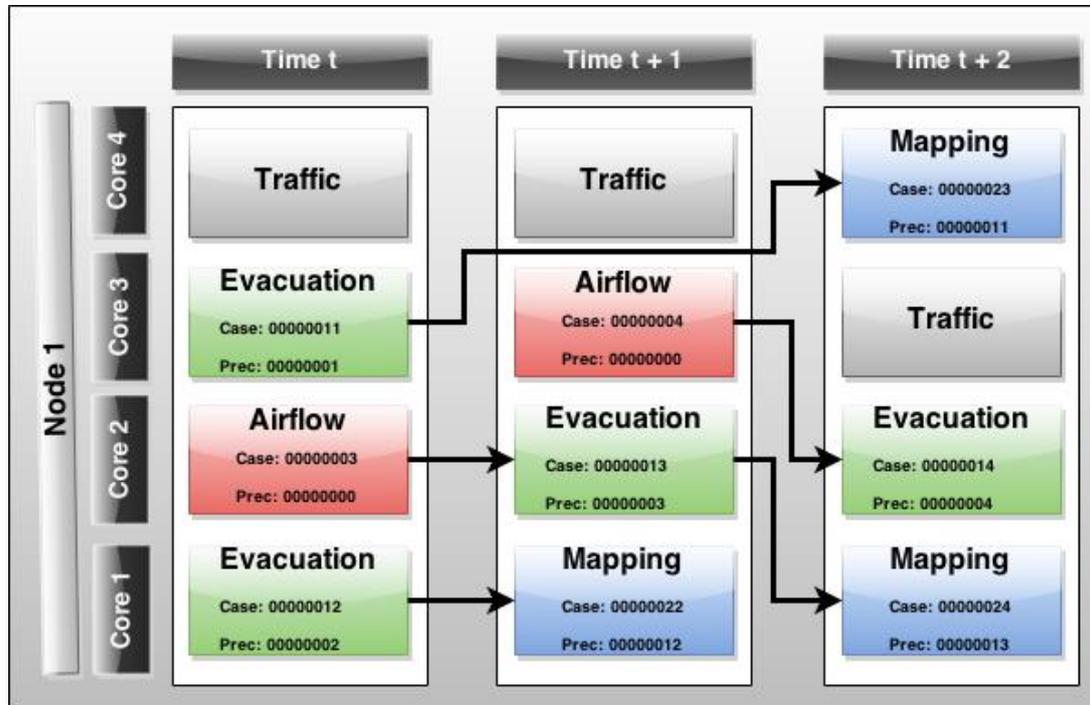


Figure 5: An example of thread execution on the same node in different cores and time steps.

4 TuRisMo

As it has been already mentioned in the introduction, Tunnel Risk analysis Model RuRisMo [6] is the Austrian response in the European 2004/54/EC directive [7]. TuRisMo consist of many integrated sub-models designed for different tasks and goals.

In order to achieve an integration and parallelization of the different sub-models it is essential, as Patzak et. al. [10, 11] have also underlined a global time-step, for solving certain partial differential equations, like Courant-Friedrichs-Lewy (CFL) condition. In the presented system, the start time, which corresponds to the time when the first simulated accident occurs, as well as its difference to the second or a possible third accident and the time step of the whole simulation has to be conformed over all sub-simulations.

4.1 Mesh Generation

Two approaches of mesh generation are essential for TuRisMo. First is the mesh applied in airflow simulation and second is the mesh for the smoke propagation simulation.

The main reason why two different approaches are used is due to the needs of accelerating the airflow simulation but having a rough estimation of the longitudinal air velocity. As a result, the airflow simulation is one dimensional simulation in contradiction to the smoke propagation simulation which is three dimensional.

The airflow simulation is one dimensional also due to the needs of having a realistic output mimicking the output of a sophisticated real time approach that would be used from the tunnel operation centre by using the data of flow measuring devices permanently installed in the tunnel. Hence the 1D mesh is non-uniform, instead it is highly dependent on the number of flow measuring devices and their installed location in the tunnel. The produced object carries the location of every element in relation to its neighbour the section it belongs together with its hydraulic diameter which is used to describe the boundaries each element is allowed to reach.

The 3D mesh is generated only for the estimated affected area by the fire (for more details, check 4.1.2). It must be mentioned that if the cross-sectional geometry varies over the tunnel, each area is considered as a new section. This is essential for the 3D mesh as it is generated for each of the sections individually. Similarly to the 1D mesh the resulting object carries the neighbour elements every node is surrounded by and initial conditions, like environment and tunnel walls temperatures of the current location together with pressure and air density. Possible altitude differences are computed as the weighted average of the given altitude in the beginning of the section to the end. Finally different sections are mapped between each other by smoothing the differences of the more complicate geometry, which is usually the wider cross-section to the narrower.

4.2 Airflow Simulation

The longitudinal airflow in a tunnel is influenced by a number of different characteristics. These can be:

- Pressure is formed in the portal of the tunnel, together with the portal losses. They can be estimated by data are monitoring from wind sensors which are located at the tunnel portals.
- Vehicles which are entering, driving or leaving the tunnel and the number of the stopped vehicles. This is the product of the traffic simulation (for more details, check 4.1.4).
- Inclination of the tunnel. Due to buoyancy.
- Jet fans as well as smoke extraction or injection systems. This is the product of the ventilation simulation (for more details, check 4.1.2)
- Altitude of the tunnel location can also intervene to the air velocity calculations as the air density will change.
- Defined quality standards have highly importance for the air velocity. (for more details, check 4.1.2)
- Initial airflow and the development after the incident happens.
- Aerodynamic parameters of vehicles.
- The drag at the tunnel walls.

The main reasons that make the 1D airflow simulation so important for our model are:

- It allows us to estimate the longitudinal air velocity which is necessary for the 3D smoke propagation simulation.
- It is capable of estimating in a great precision the influenced area from the fire. This similar to F.W. Yang et. al. [14] enables the 3D simulation of simulating only the affected area and dramatically accelerating it.

4.3 Ventilation Simulation

Characteristics capable of influencing the ventilation decisions are:

- Quality limits: Often due to the intoxicated air transmitted by the vehicles in a tunnel [17], the air quality limits have to be ensured in a tunnel. This implies that there is always the case to have operating ventilation system before a fire. This increase the complexity of the simulation and has to be taken into consideration. Usually the air quality limits is kept in low levels (under 2m/s).
- Detection time of the hazard event and the ventilation activation time are not always correspond.
- The lower and upper speed limit the local ventilation is capable of operating, together with the minimum and maximum volume it can affect.
- The ventilation strategy: Regulations are concerning the ventilation strategy have also to be taken into consideration. For example, often ventilation systems in a distance, shorter than a given distance from the incident, are not allowed to operate.
- The amount of time that the local ventilation system needs for operating with its full thrust.
- Possible failures will also be simulated. They can be defined by the user.

The outcome of the ventilation model will pass to the airflow simulation. It will determine whether the fans can affect the airflow velocity for the specified hazard location and also it will estimate if the fan is capable for braking the smoke layer which usually is formed on the top of the tunnel.

4.4 Traffic Model

In the Traffic a histogram with the hourly traffic volume is generating, which is normalized to the respective forecast year. This allows for obtaining model representative traffic scenarios, "Low traffic", "Average traffic" and "High traffic". The data for generating these histograms have been accumulated by permanent installed sensors in the tunnels.

A bidirectional tube has also to be taken into consideration. In this case the traffic has to be measured in terms of its distribution for the two directions. Hence the influence of the traffic conditions over the longitudinal air velocity, due to the piston effect, has to be considering as 0 when it is symmetric. This also implies that nine

different scenarios have to be finally simulated, e.g. “high traffic on the positive direction” etc.

The traffic model is crucial as it can influence many of the other models. As it has already been mentioned, airflow simulation is one of them, as it can be affected due to the piston effect, traffic density, volume and velocity are parameters carry information capable of influencing the airflow model. It can influence the smoke propagation as, first the traffic model will specify the location of the incident, consequently the fire location with homogeneous probability over the tunnel length, and second cars can influence the smoke propagation as obstacles capable of blocking the smoke. Evacuation simulation is also highly depending on traffic as the starting positions of the agents are cars which have been stopped.

4.5 Smoke Propagation Simulation

In the smoke propagation simulation the user can vary between different hazard scenarios:

- Fire: vary between five different fire sizes
 - 5 MW
 - 30 MW
 - 50 MW
 - 100 MW
- Dangerous goods: can be varying between small and high quantities of three types of dangerous goods:
 - Chloric Acid
 - Propane
 - TNT

Characteristics capable of influencing the airflow are:

- The Environment situation:
 - Wall temperatures
 - Ambient temperatures.
- Whether there is a water-mist system installed. The fire power is highly depending on it.
- The desirable duration of the smoke propagation simulation.

4.5.1 Fire model

As a simultaneous ignition of the total fire area is an unrealistic fire growth model, the fire model assumes that each smaller cell has a time-delayed ignition until finally reaches the full fire development time, which strictly depends on the fire size. Hence the fire model is a circular fire growth model where the cell is going to ignite first is located on the centre of the fire ramp while the cell will ignite last is located in a distance from the centred cell equal with the half of the fire ramp length.

4.6 Evacuation Simulation

- The user is able to choose among two evacuation scenarios
 - **Accumulation based evacuation simulation:** Accumulation based evacuation models are mainly used in commercial evacuation environments such as building Exodus [20]. They are normally based on the theory of accumulated intoxication by Purser [18], [19] and assess the effects of concentration and duration of exposure. These programs also allow for simulations on a two dimensional evacuation grid and simulation of bottleneck problems such as doors and sidewalks. The idea is to calculate the total amount of flue gases the evacuating agent has accumulated over the entire period and compare it to a given threshold or profit function.
 - **Limit based evacuation simulation:** The idea behind the model is that self-rescue fails if visibility drops below a certain level. For this purpose the visibility is derived for a height of 1.8m above walking surface by use of three dimensional fluid dynamics simulation environments.

Characteristics capable of influencing the evacuation simulation are:

- Alert time: it carries the time between occurrence of the hazard event and the beginning of the evacuation procedure.
- Water mist starting time: even though the water mist system works in the favour of the people in the smoke propagation simulation, here it works against, as it decrease the visibility which is crucial for the evacuation speed of an agent.
- Position of traffic lights, traffic barriers or any other mean that can stop the traffic flow is also necessary information as it affects the initial conditions of the evacuation simulation.
- The tunnel portals are automatically considered as emergency exits

4 Conclusions

This paper presents the fundamental components which consist Austrian Tunnel Risk Analysis Model, TuRisMo. It is presented the strategy that was followed for integrated these independent applications. It is also presented the cluster characteristics the design and the software that was design for interacting with the cluster. Additional, is presented how the information is exchanged between the different applications and how the status of each of the sub-models is update. Finally, is presented how the parallelization is achieved and in which level.

5 Future work

While the cluster users are increasing the need of a “greener” cluster is emerging. Similar to D.C. Sternel et. al. [21] a classification of users is essential where each user will have limitations on the cluster according to the category he belongs.

It is also essential to estimate the memory demanding that each of the job has, as well as the time each job requires to be executed and taken into consideration when the selection of which job should run next is done.

The current 3D mesh generation is limiting the system from computing complicated geometries this can easily be solved if the Tunnel CAD file is used for the mesh generation and adopt our mesh into it.

A cooperative architecture would be capable of improving the resource consumption and balance the processors temperatures. Since the CFD simulation consumes more processing power while the evacuation simulation consumes more memory RAM. It would be ideal to have an algorithm which “rewards” the more balanced distributed node. For example the node that runs an airflow and an evacuation should be rewarded over the node that runs two airflows.

There will always be the need of running simulations on windows environment. Although, some OS can be changed to Linux which allows fastest communications between the server and the nodes. This implies that the workflow manager system has to be capable of executing on mixed environments which will increase the complexity.

Last but not least, there is an important need of accelerating the smoke propagation simulation which now is the bottleneck. Probably it should be translated to openFOAM where easily will be paralelized on the GPU.

References

- [1] B. Kohl, C. Forster, S. Wiesholzer, “Upgrading of the Austrian Tunnel Risk Model TuRisMo – Methodical and Practical Aspects”, 7th International Conference “Tunnel Safety and Ventilation” 2014, Graz P. 170-179.
- [2] Ricci, F./Rokach, L./Shapira, B., “Introduction to Recommender Systems Handbook”, Recommender Systems Handbook, Springer, 2011, pp. 1-35
- [3] Reid G. Smith “The Contract Net Protocol”, IEEE Transactions on Computers, Dec 1980.
- [4] Schaller, R.R. ; Inst. of Public Policy, George Mason Univ., Fairfax, VA, USA, “Moore's law: past, present and future”, Spectrum, IEEE (Volume:34 , Issue: 6)
- [5] Edited by David C. Brock, Gordon E. Moore, “Understanding Moore's Law: Four Decades of Innovation”, Philadelphia, Pa: Chemical Heritage Press, 2006.
- [6] Österreichische Forschungsgesellschaft Straße-Schiene-Verkehr, „RVS 09.03.11 Tunnel- Risikoanalysemodell,“ 2013.
- [7] “Corrigendum to Directive 2004/54/EC of the European Parliament and of the Council of 29 April 2004 on minimum safety requirements for tunnels in the

- trans-European road network”, Official Journal of the European Union, Official Journal of the European Union L 167 of 30 April 2004.
- [8] M. M. Resch “Future High Performance Computing Strategies”, Techniques for Parallel, Distributed and Cloud Computing in Engineering. Chapter 1
 - [9] W. Gentsch and B. Yenier, “Smart Manufacturing: Engineering Simulations in the Cloud”, Techniques for Parallel, Distributed and Cloud Computing in Engineering. Chapter 2
 - [10] B. Patzak, V. Smilauer and G. Pacquaut, “Design of a Multiscale Modelling Platform”, Fourth International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering, Paper-15
 - [11] B. Patzak, D. Rypl and J. Kruijs, “On the Design of Parallel, Distributed Multi-Physics Integration Tool” Eighth International Conference on Engineering Computational Technology, Paper-3
 - [12] E. Hachem, J. Veysset, R. Klein, A. Boilley and Y. Mesri Center for Materials Forming, MINES ParisTech, Sophia Antipolis, France, “Immersed Methods and Parallel Anisotropic Mesh Adaptation to Simplify the Simulation Setup using the Cloud”, Fourth International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering, Paper-24
 - [13] L. Sellami, D. Idoughi and P.F. Tiako, “An Intrusion Detection System Based on Nodes in Cloud Computing Environments”, Fourth International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering-Paper 22
 - [14] F.W. Yang, C.E. Goodyer, M.E. Hubbard and P.K. Jimack, “Parallel Implementation of an Adaptive, Multigrid Solver for the Implicit Solution of Nonlinear Parabolic Systems, with Application to a Multi-Phase-Field Model for Tumor Growth” , Fourth International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering-Paper 39
 - [15] G. Bui and G. Meschke, “A Study on the Performance of Parallel Solvers for Coupled Simulations of Partially Saturated Soils in Tunnel Engineering Applications”, Fourth International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering-Paper 36
 - [16] Flynn, M. J., "Some Computer Organizations and Their Effectiveness", IEEE Trans. Comput. C-21 (9): 948-960. doi:10.1109/TC.1972.5009071. edit
 - [17] Hans Orru, Boel Lövenheim, Christer Johansson and Bertil Forsberg, “Potential health impacts of changes in air pollution exposure associated with

moving traffic into a road tunnel”, Journal of Exposure Science and Environmental Epidemiology , (29 April 2015)

- [18] D. A. Purser, W. D. Woolley, "Biological Studies of Combustion Atmosphere", Journal of Fire Science 1, p. 118 - 144, 1983
- [19] D. A. Purser, P. Grimshaw, "The Incapacitative Effects of Exposure to the Thermal Decomposition Products of Polyurethane Foams", Fire Master 8, p10 - 16, 1984
- [20] Handbook building Exodus v3.0, University of Greenwich, 2000
- [21] D.C. Sternel, C. Iwainsky, T. Opfer and A. Feith The Hessian Competence Centre for High Performance Computing: “Brainware” for Green IT, Fourth International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering-Paper 37