PLASA – Smart Planning and Safety for a safer and more robust European railway sector

D2.3 – Description of IT environment

Due date of deliverable: 31/08/2017

Actual submission date: 3108/2017

Leader/Responsible of this Deliverable: Torsten Betz (DB)

Reviewed: Y/N

Document status		
Revision	Date	Description
1	12/07/2017	First issue

Project funded from the European Union's Horizon 2020 research and innovation programme		
Dissemination Level		
PU	Public	x
CO	Confidential, restricted under conditions set out in Model Grant Agreement	

Start date: 01/09/2016 Duration: 24 months

REPORT CONTRIBUTORS

Name	Company	Details of Contribution
Markus Zinser	DB	Wrote deliverable
Torsten Betz	DB	Wrote deliverable
Michael Ratschko	DB	Reviewed deliverable

EXECUTIVE SUMMARY

The simulation of a day of train operation in the PLASA Smart Planning model prototype will require computation time on the order of one to a few minutes. Since the model requires repeated simulation of the same day with different random samples drawn from a number of probability distributions, the runtime for achieving meaningful results depends on the number of simulation repeats and the number of CPUs on which these are executed. Runtimes of around one hour, with 1000 repeats, should be achievable for less than 10EUR in a cloud environment with the prototype. There is also additional potential for more efficient implementation, should the prototype prove to be successful.

ABBREVIATIONS AND ACRONYMS

TABLE OF CONTENTS

LIST OF FIGURES

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

LIST OF TABLES

Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.

1. INTRODUCTION

This document describes the IT environment required for the development and testing of the simulation model developed in the Smart Planning subproject of Shift2Rail project PLASA. It gives a brief overview of the modelling framework (more details will be provided in deliverable D2.2), discusses the required input data and data structures, highlights some details of the prototype implementation that are relevant in the context of specifying IT requirements, and gives estimates of the required resources, assuming the simulation of one day of railway traffic in all of Germany as the test scenario. Since the model prototype is still under development, the figures given here are not final, but are our best estimates of the order of magnitude of the required resources and potential runtimes.

2. MODELLING FRAMEWORK

2.1 APPROACH

The proposed smart planning simulation model is a Monte Carlo simulation on a macroscopic railway infrastructure graph. The nodes on the infrastructure graph are stations, stops, and junctions, although the modelling framework is in principle flexible enough to also run on a more microscopic infrastructure graph. Arrival and departure times of all trains contained in a timetable for a given day are simulated at all infrastructure nodes. Delays at each line segment are sampled from probability distributions, and interference between trains and delay propagation is explicitly modelled through so-called “interference functions”. The same day is simulated multiple times, with different random samplings from the delay probability distributions. The output of the model is a posterior distribution of delay probabilities for all trains at all infrastructure nodes. This model allows evaluating the effects of changing delay probabilities at certain line segments or for specific train types on the overall punctuality within the network, including secondary effects.

2.2 REQUIRED INPUT DATA

The fundamental input to a simulation run is a timetable. The level of detail of the timetable determines the level of detail of the simulation. If the timetable is microscopic, the simulation will be microscopic, if the timetable is macroscopic, so will be the simulation. For the prototype, we will be working with a macroscopic timetable.

In order to set up the model for a specific infrastructure graph, historical train operation data is required to calibrate model parameters:

- Disruption data for fitting probability distributions for the occurrence of certain types of disruptions
- Train operation data for fitting probability distributions of delay times incurred for specific types of disruptions
- Infrastructure data including type of signalling system and technical parameters of the trains for computing minimum headways on all segments

- Minimum runtimes of different train types for all segments can either be computed based on infrastructure and train information in a separate microscopic simulation, or be extracted from historical operation data

All parameters of the model can then be adjusted to simulate the effect this has on the punctuality of the entire network, or of specific trains. Alternatively, a new timetable can be tested and evaluated with the parameters reflecting the status quo.

2.3 DATA STRUCTURES

The model builds a “train interference graph” based on the timetable provided. Nodes in this graph are the passages of individual trains at infrastructure graph nodes. Thus, every time specified in the timetable translates into one node on the interference graph. There are two types of edges: First, edges connecting nodes that correspond to consecutive infrastructure nodes on the path that a train takes through the network. And second, interference edges, that connect graph nodes that correspond to passages of different trains at the same infrastructure node, where interference between the trains is possible. The simulation makes use of the following data structures:

- A list of train interference graph nodes. A graph node contains information on planned arrival and departure time, and simulated arrival and departure time can be written into it
- A list of train interference graph edges
- Probabilities for different types of disruptions at each infrastructure edge
- Parameters of a normal distribution describing small time fluctuations of the travel time. These parameters must be specified for each type of train and type of infrastructure.
- Parameters of a log-normal distribution specifying the probability of the duration of a given disruption. These parameters must be specified for each type of disruption and type of infrastructure.
- Parameters of a Weibull distribution specifying the probability of delay times given a train encounters a disruption. These parameters must be specified for each type of train, time interval, type of disruption, and type of infrastructure.
- Minimum headways for each combination of train types at each infrastructure node
- Minimum drive time for each type of train at each infrastructure node

2.4 POTENTIAL FOR PARALLELIZATION

The model repeatedly simulates the same timetable day, drawing new samples from the random distributions of disruptions and delays each time. Since the simulations of individual repetitions of this day are independent, parallelization is trivial. Each simulated day can be run on independent IT infrastructure. It is also possible to parallelize the simulation of an individual day, because the train interference graph can be separated into independent sub-graphs that can be run in parallel, and then recombined at points where the results of one sub-graph are required for the

computations in another one. However, at least in the prototype phase, this more complex approach will not be necessary, because the number of days to be simulated will exceed the number of available CPUs.

2.5 IMPLEMENTATION

The prototype is implemented in the R programming language. This makes prototyping and experimenting relatively quick and easy, and also allows for a natural integration of the data analysis needed for fitting the parameters and evaluating the quality of the simulation. However, the implementation is not fully optimized for speed or memory usage, as premature optimization would slow down development. Additional potential for speed increases exists by switching to a faster language (e.g. C# or C++) once the model reaches a mature state.

3. SCALE OF SIMULATION

The scale of the simulation depends on the size of the network being simulated. For benchmarking purposes, an entire day of operation in all types of rail traffic (long distance passenger trains, regional passenger trains, freight traffic) on the entire DB railway network is used. For this use case, the train interference graph consists of about 1.1 million nodes and 10 million edges. Of these edges, 1 million are for connecting nodes that correspond to consecutive infrastructure nodes on the path that a train takes through the network, 7.5 million are for simulating train interference due to multiple trains using the same infrastructure, 0.4 million are for train interference caused by waiting for connecting passengers, and 1.1 million are required for internal computations.

The number of repeated simulations of the same timetable day required to obtain meaningful and robust output distributions depends on the studied problem. If one is only interested in the total delay distribution across all trains (or all trains of a specific category), the output distribution is already reasonably smooth when only simulating one day, due to the large number of different trains. If one is interested in output distributions for a specific train, line, or station, multiple simulation runs are required for the output distribution to converge. We estimate that for most problems, the number of repetitions should not exceed 1000.

4. COMPUTATIONAL REQUIREMENTS

4.1 CPU

Benchmarks were performed with a preliminary version of the model prototype. Not all types of train interference are implemented yet, so the times reported here only give a general impression of the order of magnitude of runtime requirements of the model. In particular, complex interference functions between multiple trains at larger hubs are not implemented yet, and will increase the runtime. When simulating one day of operation on the German network graph described in Section 3 on a single thread Intel I7-6600U CPU @ 2.6 GHz, the following times where achieved:

- 30 seconds to prepare data sets (required only once per simulation, independent of the number of days simulated)
- 30 seconds to prepare a single day of simulation (distribute disruptions and calculate durations)
- 40 seconds to simulate all train arrival and departure times

4.2 MEMORY

The memory requirements depend very much on the details of the implementation and the representation of the different necessary data structures. With the current prototype implementation, the main data structures require the following memory:

- Simulated disruptions: 800 MB
- Interference graph nodes: 1 GB
- Parameters: 200 MB
- Result: 100 MB

However, during simulation, memory usage can peak at 12 GB due to internal calculations when preparing the input for a single day.

If memory size becomes a critical issue, some of these data structures could be shared across multiple CPU cores, because they are read-only during a simulation run.

4.3 HARD DISK

The results for a single day take up about 500 MB in memory, and 15 MB if stored in compressed format on a hard disk. Thus, saving the entire output of a 1000 day simulation of the entire German railway network could be achieved with 15 GB of storage at the current macroscopic network resolution.

5. PROPOSED INFRASTRUCTURE

The model prototype can be run on any system with an R installation. One flexible solution would be to use on-demand cloud-based servers. A viable example architecture would be Amazon Web Services (AWS), using Intel Xeon processors @2.4 GHz. These processors are slightly slower than those used in our benchmark, so one day of simulation might be achieved in ca. 90 s, ignoring the fixed setup time. This implies that one CPU could run 40 simulated days in one hour, and it would take 25 instances to simulate 1000 days in one hour. A single CPU with 16 GB ram costs $0.24 per hour resulting in a total cost of $6 for the simulation of 1000 days.

Another possibility would be to purchase a dedicated server with 25 CPUs. In that case, the RAM requirement should be lower than 25 * 12 = 300 GB, because not each CPU will have the peak RAM requirement at the same time. However, in that case, only one simulation could be run at any given time to achieve the one hour runtime for 1000 simulated days. The cloud-based solution, on the other hand, could easily be scaled up to run multiple simulations at the same time.

6. CONCLUSIONS

Extrapolating the IT requirements from the performance of the current (July 2017) prototype, our best estimate is that simulating a specific scenario to a high level of accuracy, allowing conclusions about the delay probability distributions of individual trains, should be achievable for under 10EUR. Considering that the manual work by an expert required to set up the scenario and evaluate the results will exceed this price by an order of magnitude, the IT requirements for the PLASA simulation model appear to be well within practical limits for the present use cases. However, it is also clear that the model in its present form is not yet suitable for batch processing of thousands of scenarios differing only slightly, as might be required for optimization.