A Thread-Based Parallel Implementation of the Open Source Framework MATSim Episim

Steffen Fürst & Christian Rakow

Zuse Institut Berlin & Technical University of Berlin

IXPUG Mid-Year Workshop 2021

June 21, 2021

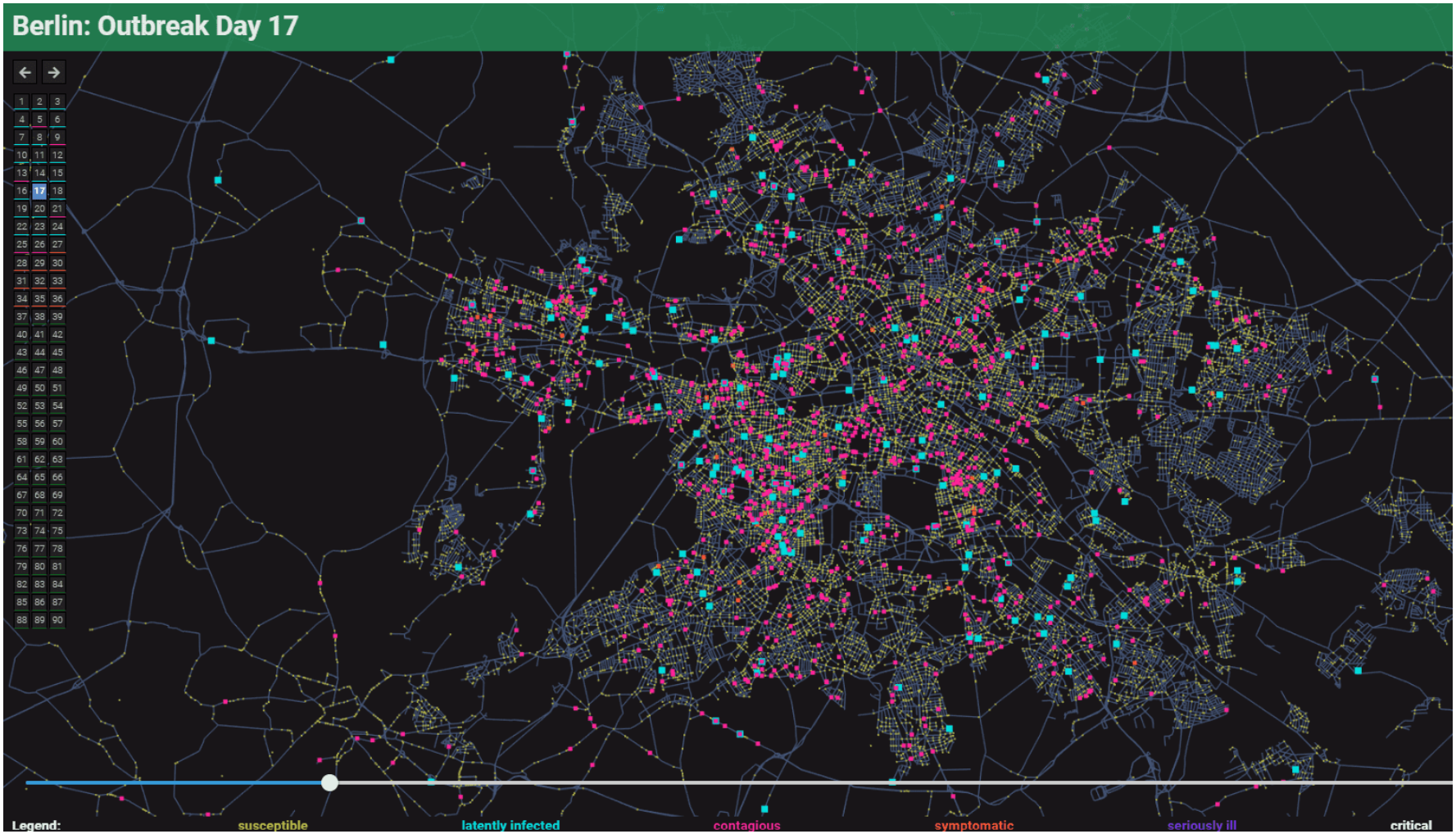
MATSim

MATSim is an open-source framework for implementing large-scale agent-based transport simulations.

Source: [1]

MATSim Episim

MATSim Episim is an epidemic simulation based on MATSim.



Source: [2]

MATSim Episim

MATSim Episim is an epidemic simulation based on MATSim.

Where based means:

- It uses event files created by MATSim as input.
- It uses some parts of MATSim, e.g. to read in these files.

MATSim Episim

MATSim Episim is an epidemic simulation based on MATSim.

Where based means:

- It uses event files created by MATSim as input.
- It uses some parts of MATSim, e.g. to read in these files.

But apart from that, Episim is independent single-threaded Java code.

Epidemiological Simulations

The following elements are in nearly every agent-based epidemiological simulation:

Synthetic population

id	age	gender	household-id
1324019	42	f	382741
1324020	29	m	395832

Epidemiological Simulations

The following elements are in nearly every agent-based epidemiological simulation:

Synthetic population

Locations

id	type
342813	shopping
9856325	public transport

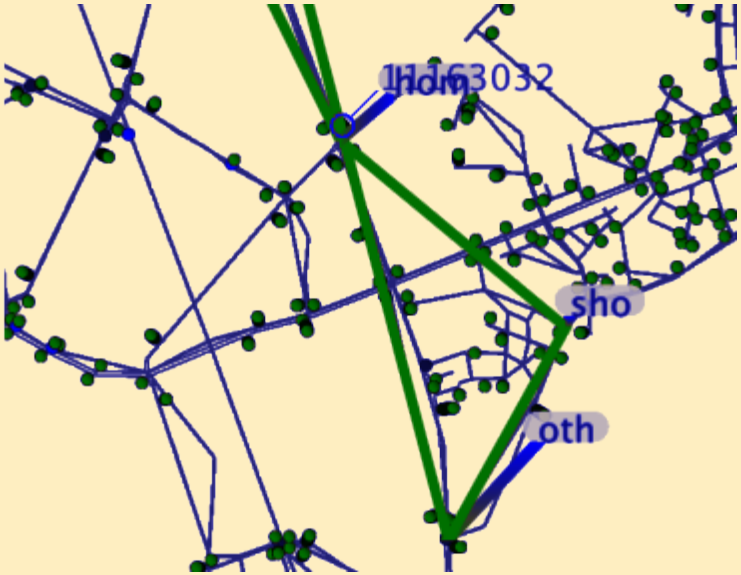
Epidemiological Simulations

The following elements are in nearly every agent-based epidemiological simulation:

Synthetic population

Locations

Movement profile



Epidemiological Simulations

The following elements are in nearly every agent-based epidemiological simulation:

Synthetic population

Locations

Movement profile

person-id	time	location-id
1324020	10:00 - 10:20	9856325
1324020	10:20 - 11:40	342813

Epidemiological Simulations

The following elements are in nearly every agent-based epidemiological simulation:

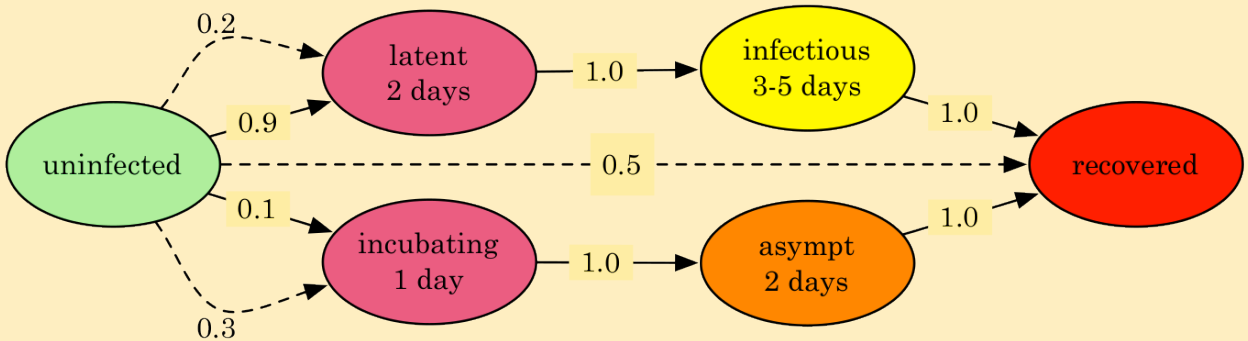
- Synthetic population
- Locations
- Movement profile
- Infection model

$$p(infect|contact) \approx \theta \cdot \frac{sheddingRate \cdot intake}{roomSize \cdot airExchange} \cdot duration$$

Epidemiological Simulations

The following elements are in nearly every agent-based epidemiological simulation:

- Synthetic population
- Locations
- Movement profile
- Infection model
- Disease progression model



Source: [3]

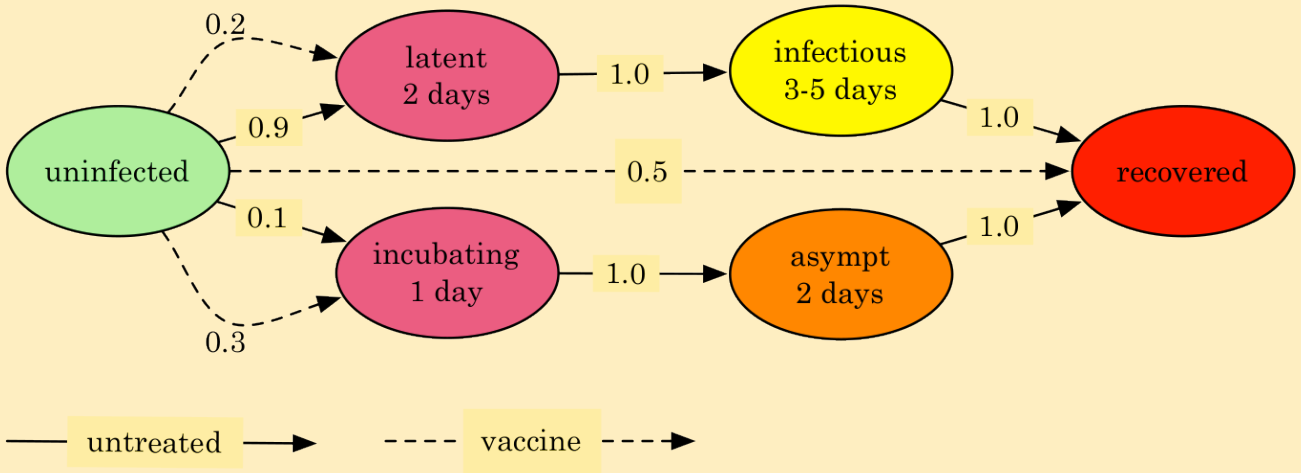
Parallelization Strategy

A promising parallelization approach builds on two properties:

Parallelization Strategy

A promising parallelization approach builds on two properties:

Health state changes are not immediately



Source: [3]

Parallelization Strategy

A promising parallelization approach builds on two properties:

Health state changes are not immediately

The movement profiles can be divided among the different locations

time	action	person-id	location-id
10:00	arrive	1324020	9856325
10:05	leave	493592	9856325
10:07	leave	948539	9856325

$\Rightarrow$ This allows independent processing of events for each location, with a barrier at the end of a simulated day.

Multithreaded Version

About 80% of the runtime is used to process the movement profiles.

Multithreaded Version

About 80% of the runtime is used to process the movement profiles.

For parallelizing the movement profiles:

- In the initialization each locationen is assigned to a task-id.
- The fact that locations are visited by different numbers of people is taken into account.
- To handle the movement profile events an array of `CompletableFutures` is run asynchronously.

Multithreaded Version

About 80% of the runtime is used to process the movement profiles.

For parallelizing the movement profiles:

- In the initialization each locationen is assigned to a task-id.
- The fact that locations are visited by different numbers of people is taken into account.
- To handle the movement profile events an array of `CompletableFutures` is run asynchronously.

In the remaining part of the code, two time-demanding loops could be parallelized by converting them to a Java8 `ParallelStream`.

Serial Performance Degradation

It was necessary to change code to a version with a worse performance:

- For a `ParallelStream`, the order of the nested loops had to be changed, even though the original version was more cache friendly.

Serial Performance Degradation

It was necessary to change code to a version with a worse performance:

- For a `ParallelStream`, the order of the nested loops had to be changed, even though the original version was more cache friendly.
- Events in the movement profile of a person are dependent on the previous and subsequent ones, retrieving them is now more costly.

Serial Performance Degradation

It was necessary to change code to a version with a worse performance:

- For a `ParallelStream`, the order of the nested loops had to be changed, even though the original version was more cache friendly.
- Events in the movement profile of a person are dependent on the previous and subsequent ones, retrieving them is now more costly.

Performance loss: $\approx$ 20%

MATSim Episim Simulations

The most used simulation configuration is for the Berlin metro area (ca. 5 mio people), with one agent for every four persons.

- 10000-20000 simulations per report (every 3 weeks)
- Runtime (single-threaded, one simulation): $\approx$ 2 h
- Memory usage: $\approx$ 11 GB
- File I/O: 500 / 34 MB

MATSim Episim Simulations

The most used simulation configuration is for the Berlin metro area (ca. 5 mio people), with one agent for every four persons.

- 10000-20000 simulations per report (every 3 weeks)
- Runtime (single-threaded, one simulation): $\approx$ 2 h
- Memory usage: $\approx$ 11 GB
- File I/O: 500 / 34 MB

Standard compute node of HLRN's Lise cluster:

- 2 Intel Cascade 9242 (= 96 cores)
- 384 GB Ram

MATSim Episim Simulations

The most used simulation configuration is for the Berlin metro area (ca. 5 mio people), with one agent for every four persons.

- 10000-20000 simulations per report (every 3 weeks)
- Runtime (single-threaded, one simulation): $\approx$ 2 h
- Memory usage: $\approx$ 11 GB
- File I/O: 500 / 34 MB

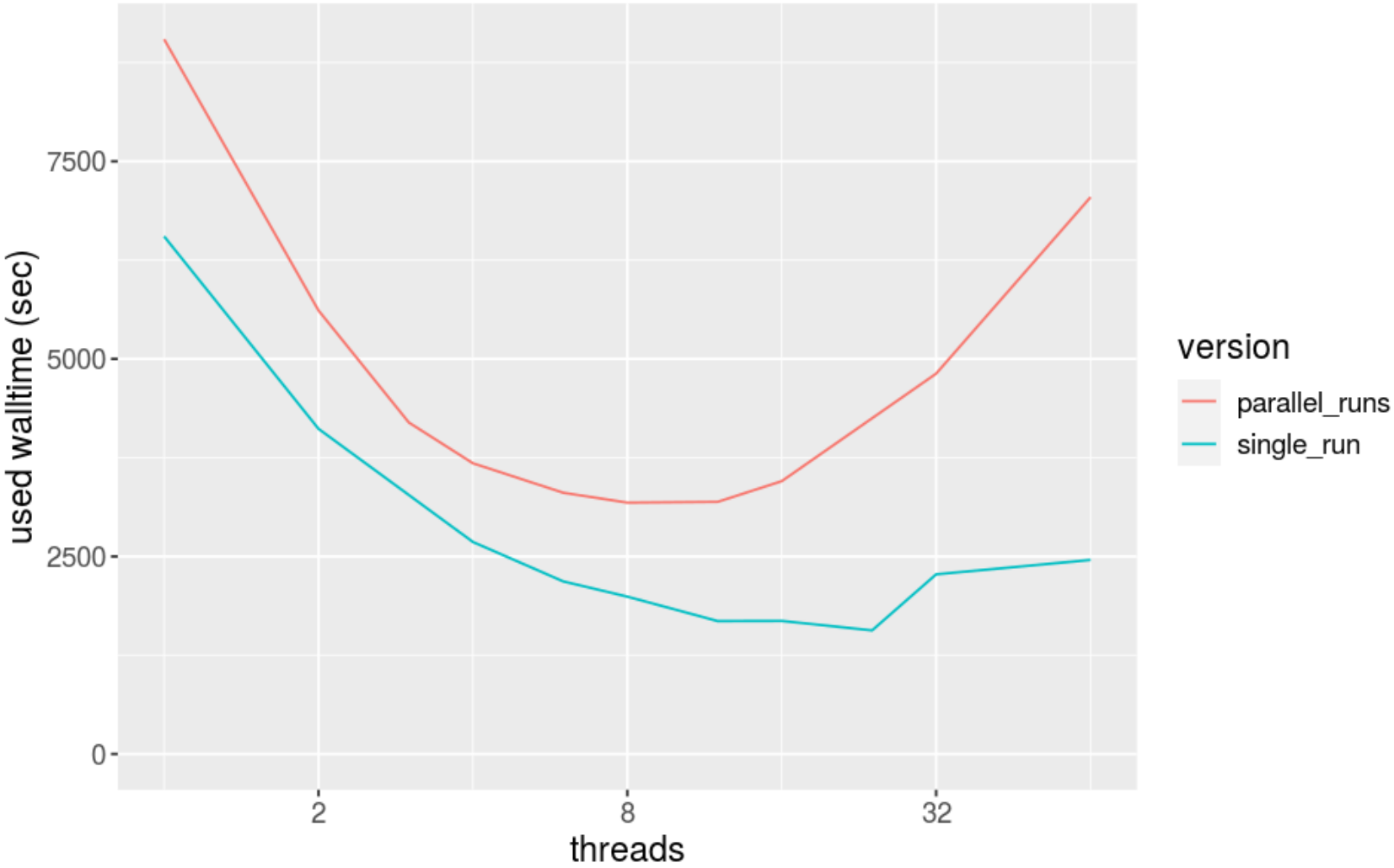
Standard compute node of HLRN's Lise cluster:

- 2 Intel Cascade 9242 (= 96 cores)
- 384 GB Ram

Because of the memory constraint only 32 simulation can be started in parallel.

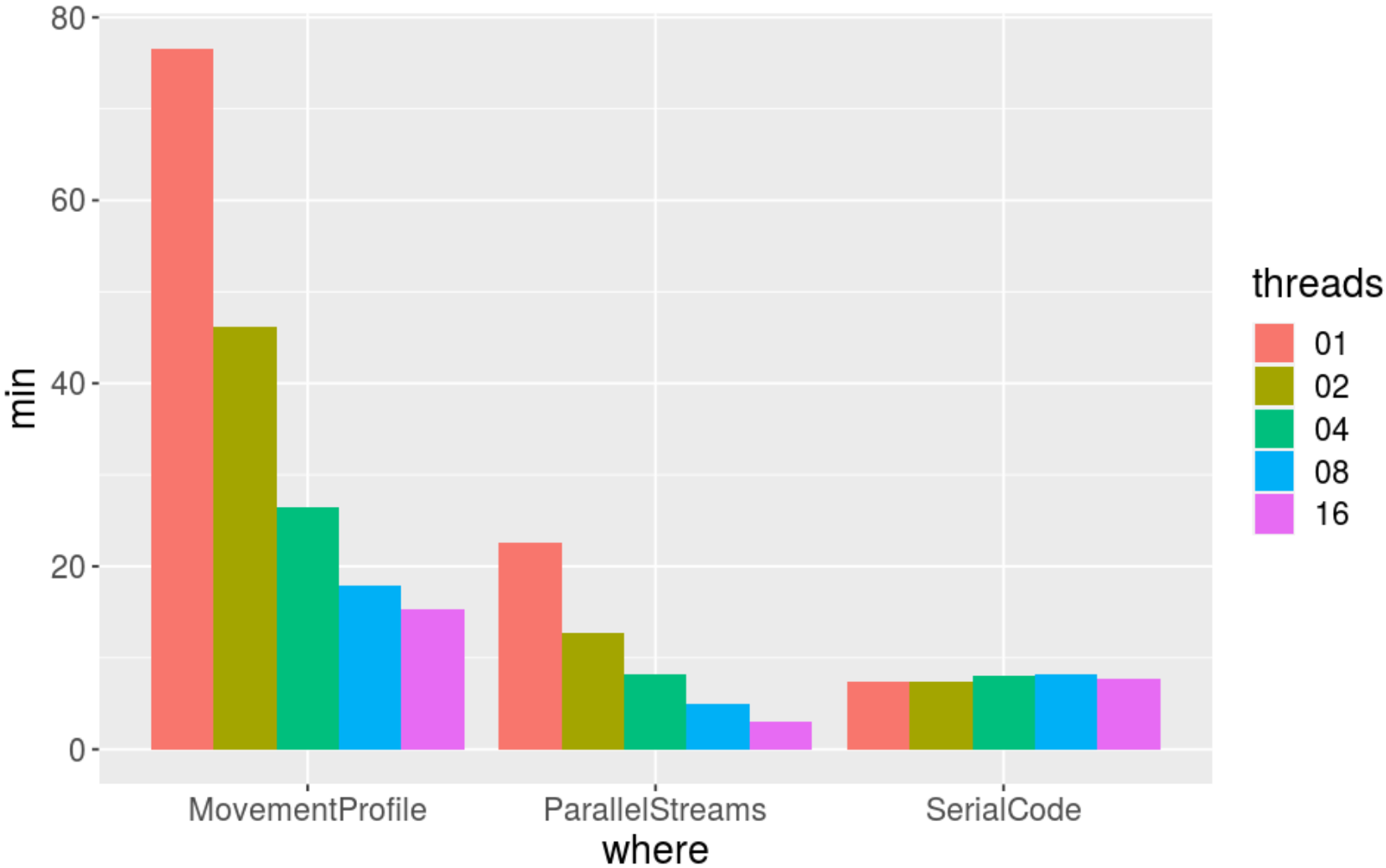
Results Berlin 25%: Walltime

Walltime of Berlin scenario with 25% population



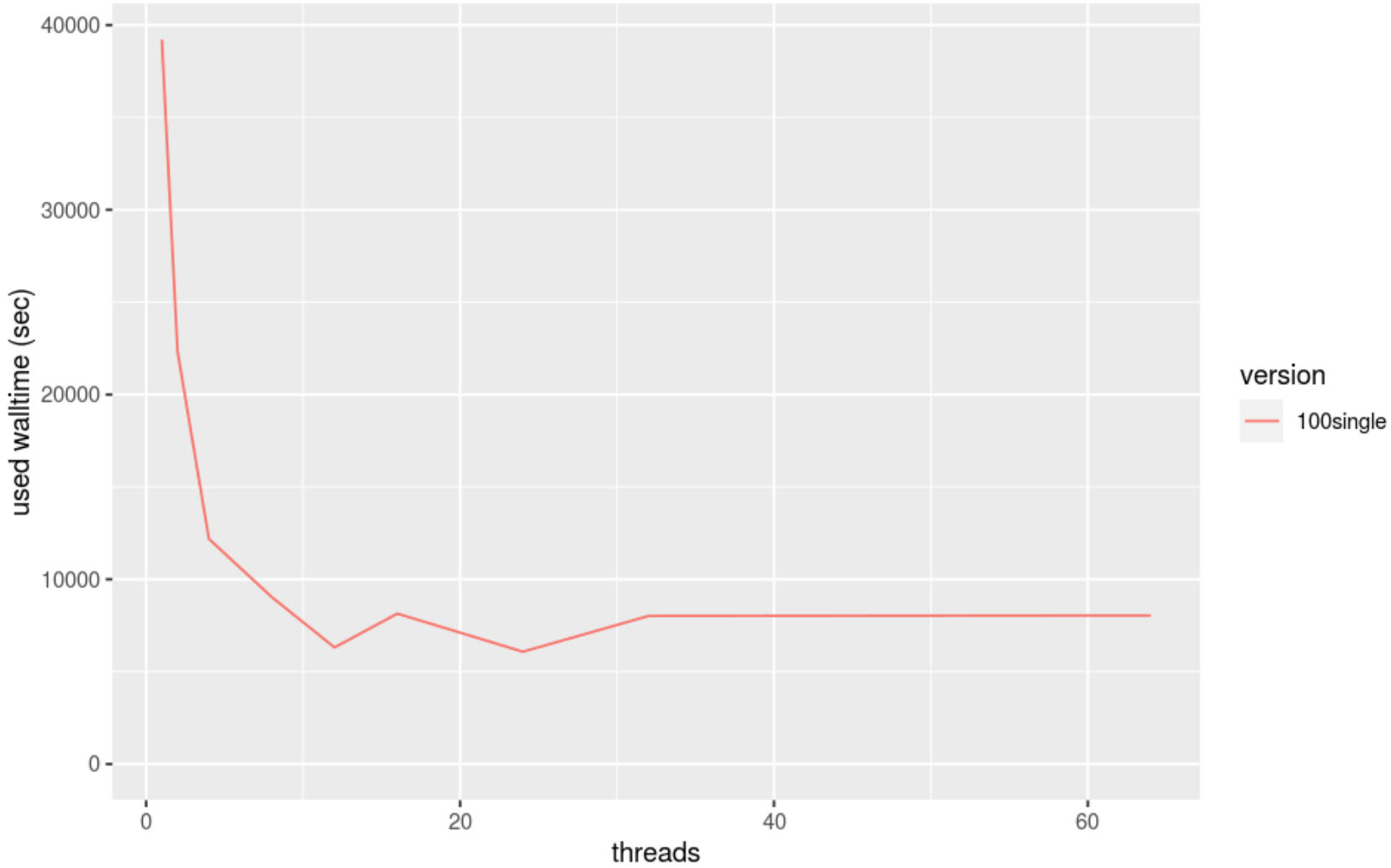
Berlin 25%: Walltime of Different Parts

Walltime of different parts (version: single_run)



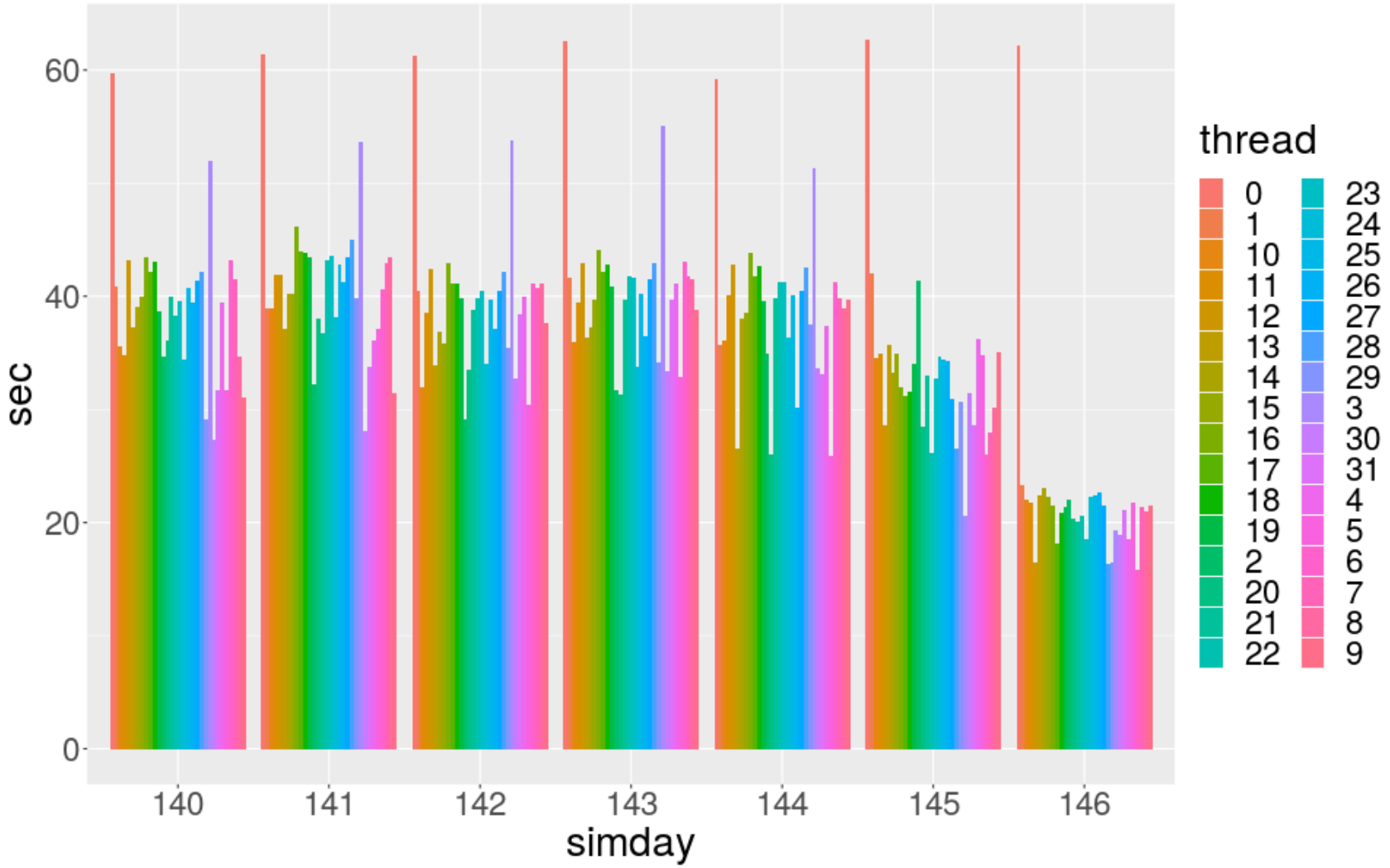
Berlin 100%: Walltime

Walltime of Berlin scenario with 100% population



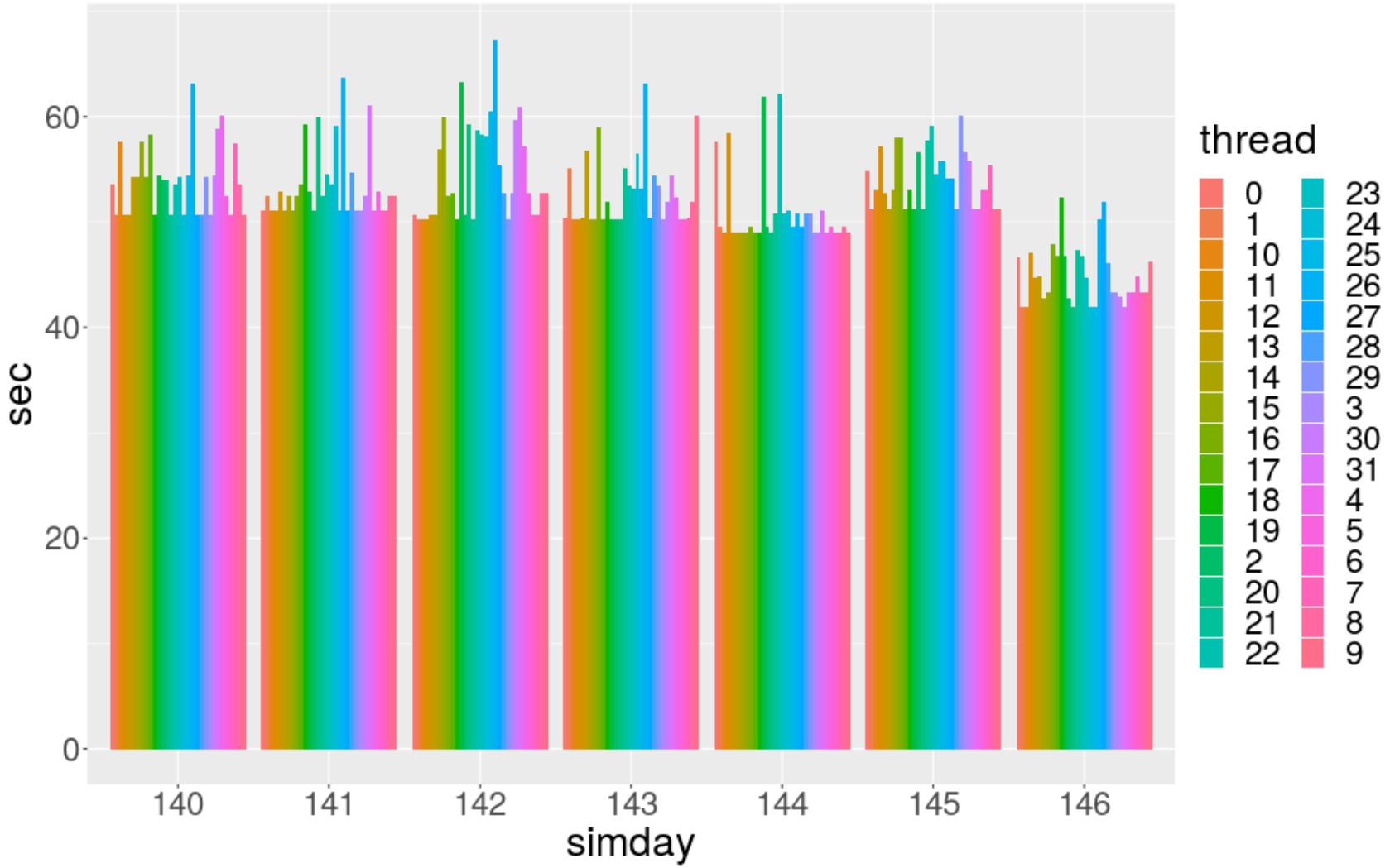
Berlin 100%: Runtime per Thread

Time used for handling the movement profiles for seperate days



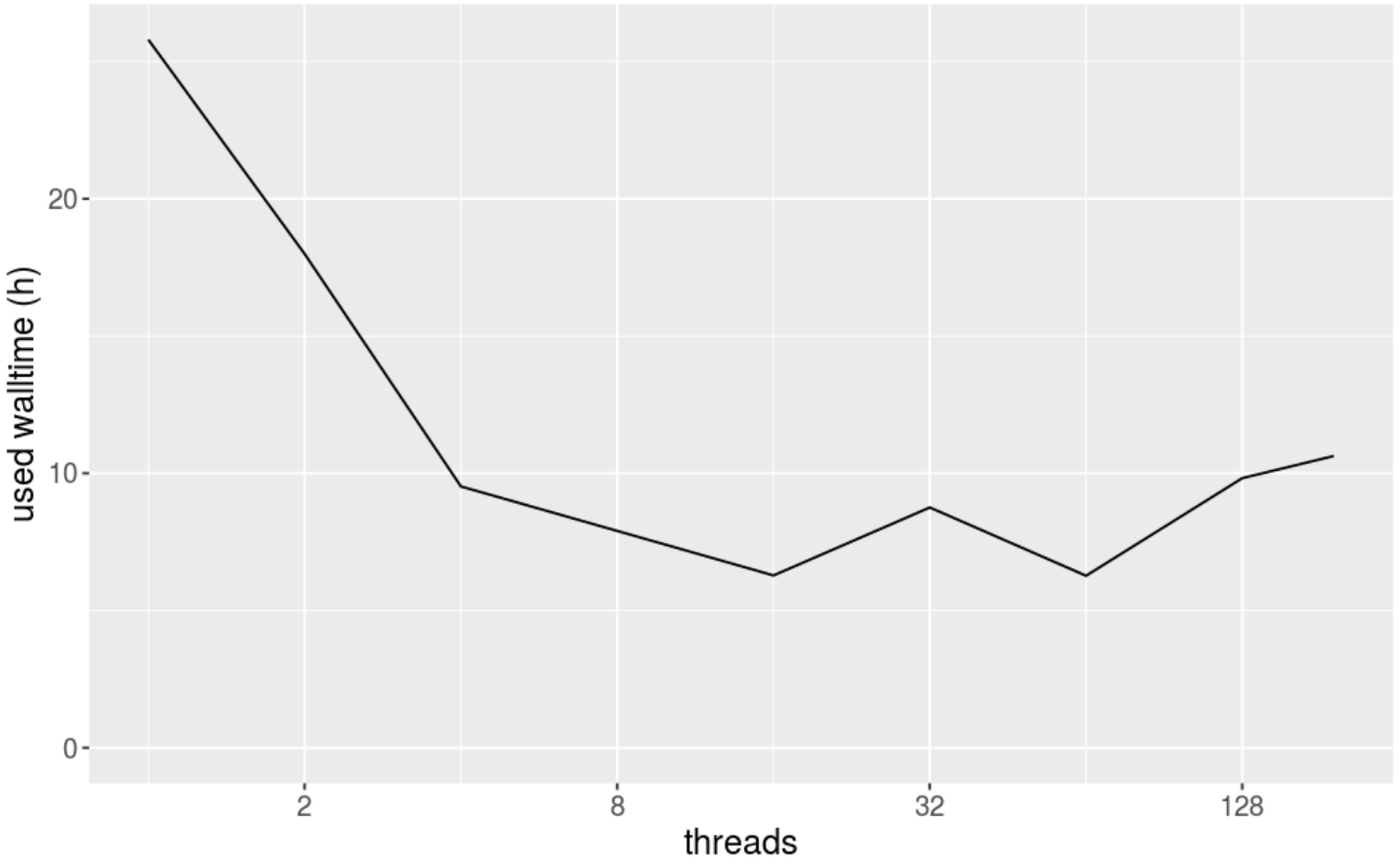
Germany 25%: Runtime per Thread

Time used for handling the movement profiles for seperate days



Germany 25%: Walltime

Walltime of Germany scenario with 25% population



Further Information

Links:

- `https://covid-sim.info/`: MATSim Episim portal with model results, published reports ...
- `https://github.com/matsim-org/matsim-episim-libs`: MATSim Episim source code (the work presented here is in the branch `multithreaded`).

Figure Sources:

[1] `https://www.matsim.org/gallery/poznan`

[2] `https://covid-sim.info/v3?day=17`

[3] C. L. Barrett, K. R. Bisset, S. G. Eubank, Xizhou Feng and M. V. Marathe, "EpiSimdemics: An efficient algorithm for simulating the spread of infectious disease over large realistic social networks," *SC '08: Proceedings of the 2008 ACM/IEEE Conference on Supercomputing*, 2008, pp. 1-12

Acknowledgement: The work on the paper was funded by the Ministry of research and education (BMBF) Germany (01KX2022A) and TU Berlin

Thank you for your attention!