Boosting Complex-Systems Research through RSE Collaboration

Jeffrey Kelling[1,2], Richa Tripathi[3], Justin M. Calabrese[3]

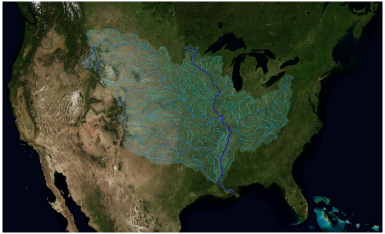
[1] Department of Information Services and Computing, Helmholtz-Zentrum Dresden - Rossendorf (HZDR)
mailto:j.kelling@hzdr.de, https://www.hzdr.de

[2] Faculty of Natural Sciences, Chemnitz University of Technology
[3] Center for Advanced Systems Understanding (CASUS), Helmholtz-Zentrum Dresden - Rossendorf (HZDR)

February 20, 2023

Context: Biodiversity in River Networks

Mississippi–Missouri River System

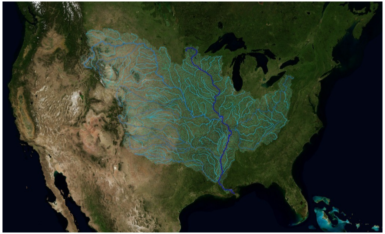


https://svs.gsfc.nasa.gov/4493

⟶

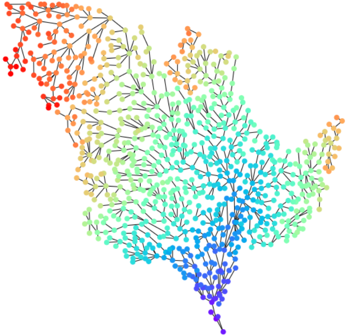
Context: Biodiversity in River Networks

Mississippi–Missouri River System



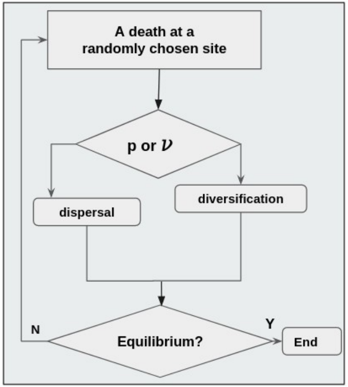
https://svs.gsfc.nasa.gov/4493

⟶



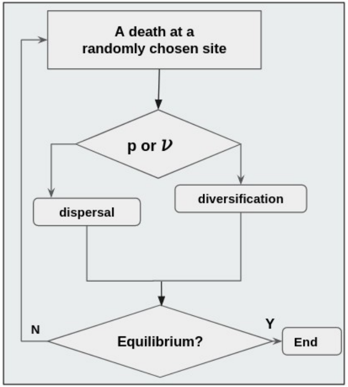
reduction to ∼ 800 nodes

Stochastic Model: Neutral Model of Biodiversity



- Models death → replacement → death-cycle.
- Neutral with respect to species (no niches).

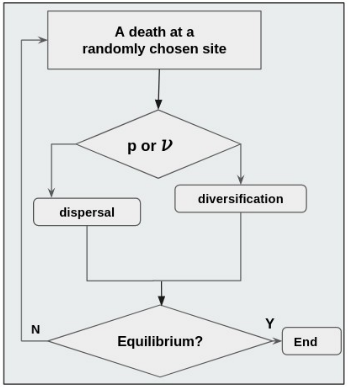
Stochastic Model: Neutral Model of Biodiversity



- Models death → replacement → death-cycle.
- Neutral with respect to species (no niches).

- Single individual replaced by
  - p dispersal— species from different site with probability $p_{i,j}$, or
  - ν diversification— new species with probability ν (small).

Stochastic Model: Neutral Model of Biodiversity



A death at a randomly chosen site

p or ν

dispersal

diversification

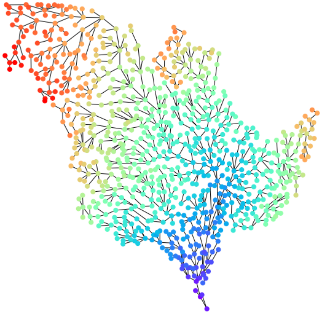
Equilibrium?

N

Y

End

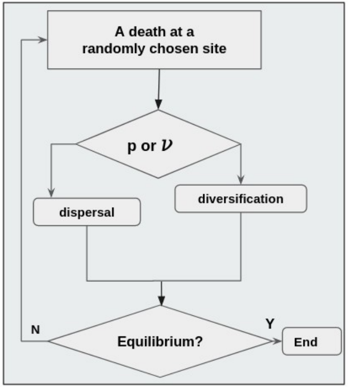
- Models death → replacement → death-cycle.
- Neutral with respect to species (no niches).
- Single individual replaced by
  - p dispersal— species from different site with probability $p_{i,j}$, or
  - ν diversification— new species with probability ν (small).



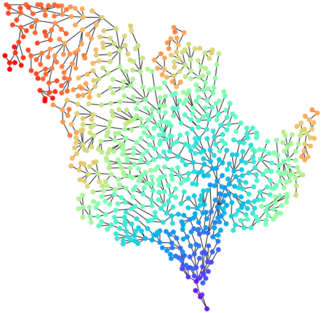
$\sim$ 800 nodes

$\sim$ 20 000 total sites

Stochastic Model: Neutral Model of Biodiversity



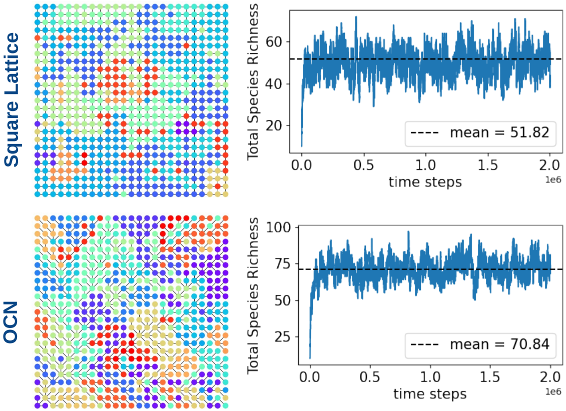
- Models death → replacement → death-cycle.
- Neutral with respect to species (no niches).
- Single individual replaced by
  - p dispersal—species from different site with probability $p_{i,j}$, or
  - ν diversification—new species with probability ν (small).
- $p_{i,j}$ applies to nodes.
- Nodes contain multiple sites.



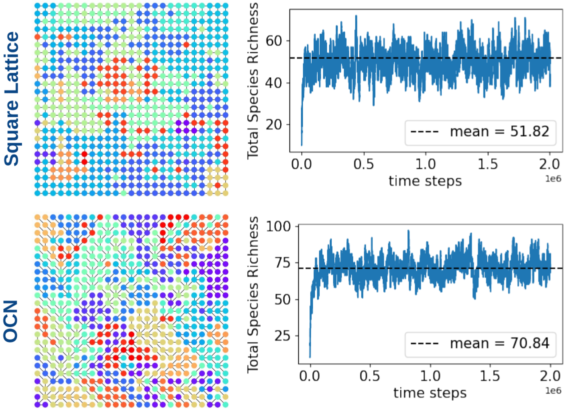
$\sim$ 800 nodes

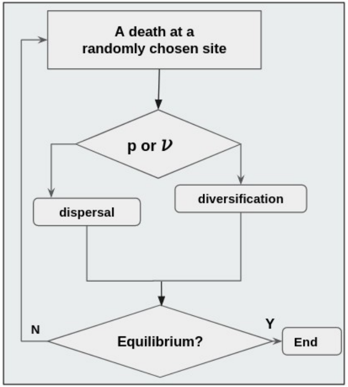
$\sim$ 20 000 total sites

Example Observable: Total Species Richness



Example Observable: Total Species Richness




A death at a randomly chosen site

p or ν

dispersal

diversification

Equilibrium?

N

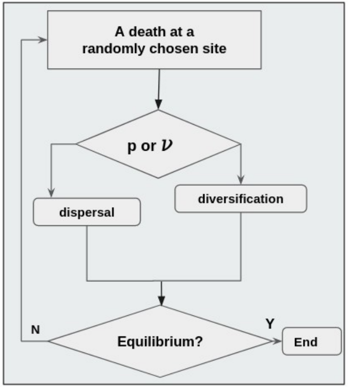
Y

End

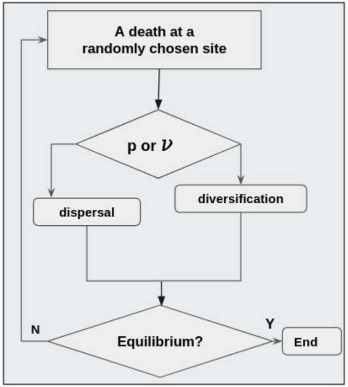
$> 10^6$ loop iterations

Starting Point and Goal

Starting Point: Python code



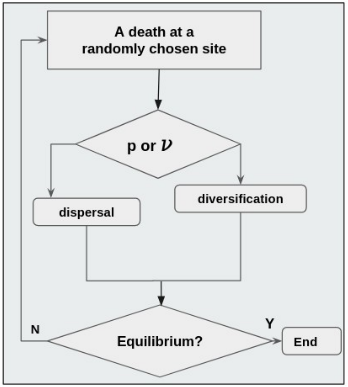
Starting Point and Goal



Starting Point: Python code

- inefficiencies in random-event generation

Starting Point and Goal

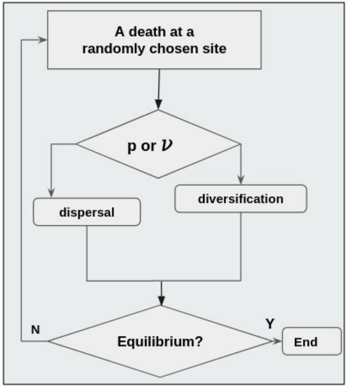


Starting Point: Python code

- inefficiencies in random-event generation
- $\sim 0.03\,\mathrm{ms}$ per update
- Algorithm does not allow much use of numpy & friends.
- Large amount of time spend on analysis.

⇒ main problem: python

Starting Point and Goal



Starting Point: Python code

- inefficiencies in random-event generation
- $\sim 0.03\,\mathrm{ms}$ per update
- Algorithm does not allow much use of numpy & friends.
- Large amount of time spend on analysis.

⇒ main problem: python

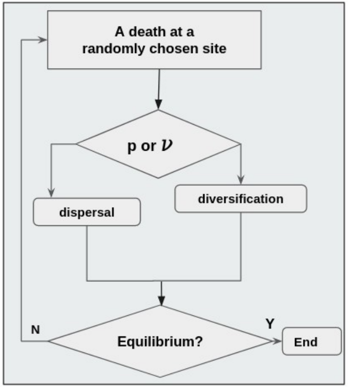
Goal: GPU code

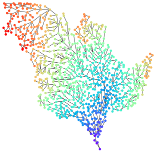
→ larger graphs may be interesting in the future

→ large parameter studies

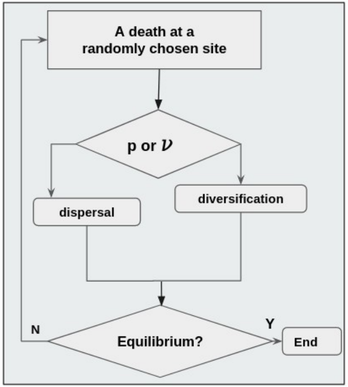
- GPUs are available to the institute.
- Energy-efficient compute.

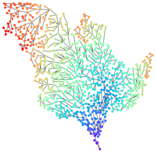
Parallelism





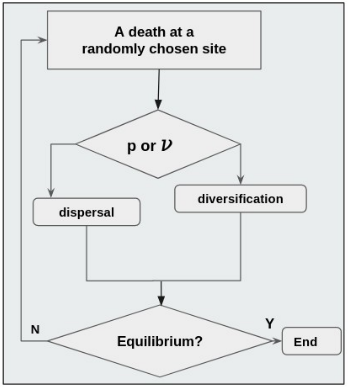
Parallelism





1. Replace a site in each node simultaneously.
   - Need to add process to avoid or resolve logical conflicts…
   - … not everyone is comfortable casually modifying accepted models in their field.

Parallelism



A death at a randomly chosen site

p or ν

dispersal

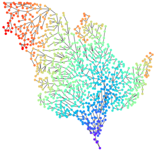
diversification

Equilibrium?

N

Y

End

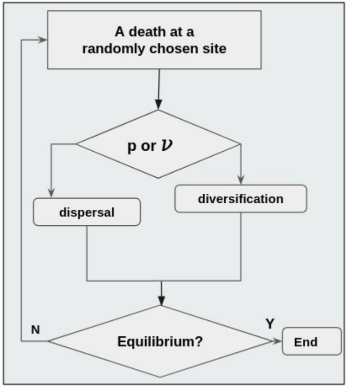


1. Replace a site in each node simultaneously.
   - Need to add process to avoid or resolve logical conflicts...
   - ... not everyone is comfortable casually modifying accepted models in their field.
   - Small graph means limited paralellism.

"Here is a GPU implementation, won't do much for your puny graph, but will be fast when you get to real big ones."

— True, but not very helpful.

Parallelism



A death at a randomly chosen site

p or ν

dispersal

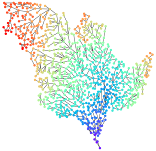
diversification

Equilibrium?

N

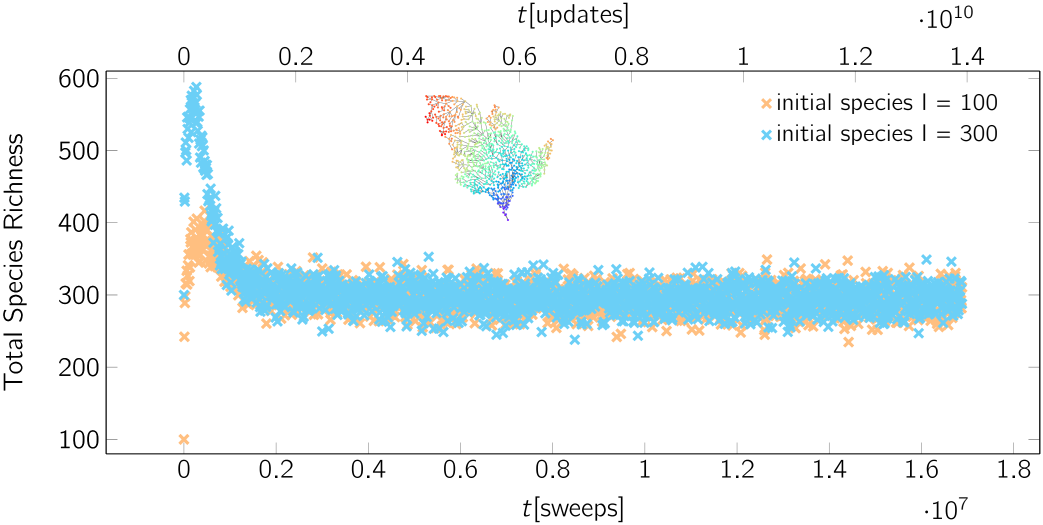
Y

End



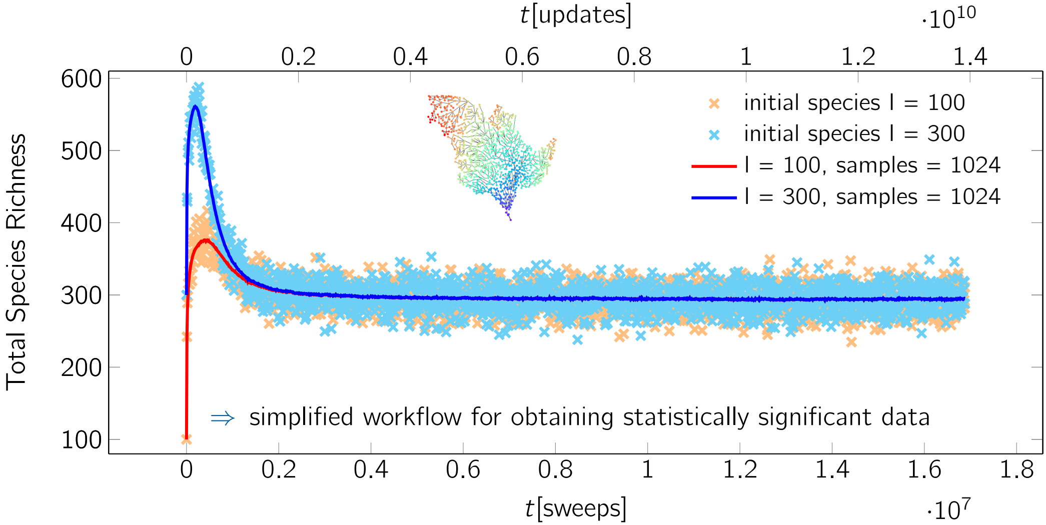
1. Replace a site in each node simultaneously.
   - Need to add process to avoid or resolve logical conflicts…
   - … not everyone is comfortable casually modifying accepted models in their field.
   - Small graph means limited paralellism.
2. Run multiple (partially) independent simulations
   - for averaging,
   - parameter studies or
   - response functions.

Averaging



t[updates]
$\cdot 10^{10}$
0
0.2
0.4
0.6
0.8
1
1.2
1.4
Total Species Richness
600
500
400
300
200
100
× initial species I = 100
× initial species I = 300
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
t[sweeps]
$\cdot 10^{7}$

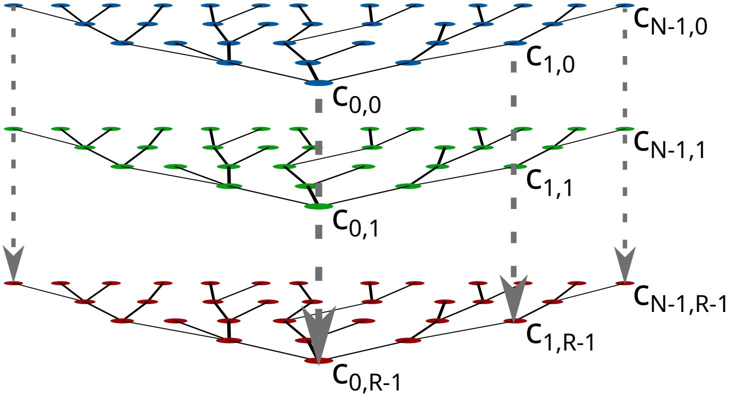
Averaging



⇒ simplified workflow for obtaining statistically significant data

Replica Stacking

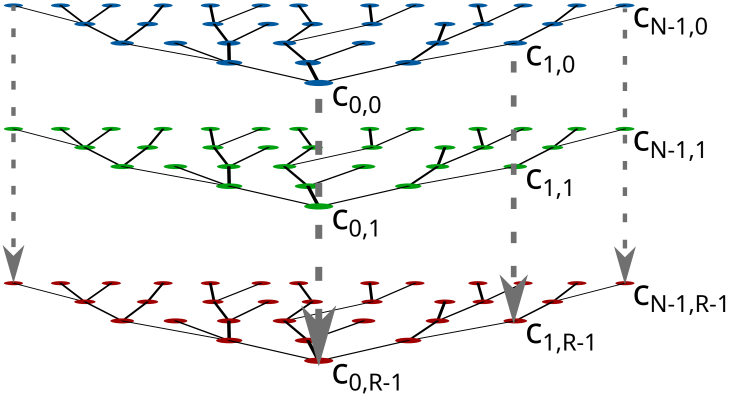
- Variables from different replica are adjecent in linear memory:



$$\left[c_{0,0}, c_{0,1}, \ldots, c_{0,R-1}, c_{0,1}, c_{1,1}, \ldots, c_{1,R-1}, \ldots, c_{N-1,0}, c_{N-1,1}, \ldots, c_{N-1,R-1}\right]$$

Replica Stacking

- Variables from different replica are adjecent in linear memory:

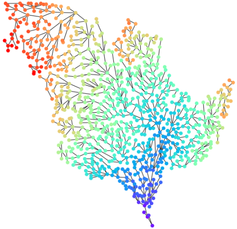


$$[c_{0,0}, c_{0,1}, \dots, c_{0,R-1}, c_{0,1}, c_{1,1}, \dots, c_{1,R-1}, \dots, c_{N-1,0}, c_{N-1,1}, \dots, c_{N-1,R-1}]$$

- Chose to replace the same site for all* replica.

Implementation

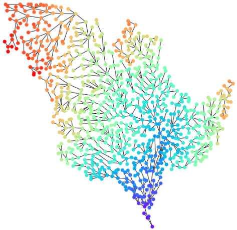
- C++ + alpaka (https://github.com/alpaka-group/alpaka)
  - template-based offloading abstraction layer
  - backends: sequential CPU, OpenMP, CUDA, HIP, ...
  - Application logic decoupled from target execution model by abstraction.



[a] http://doi.acm.org/10.1145/2063384.2063405

Implementation

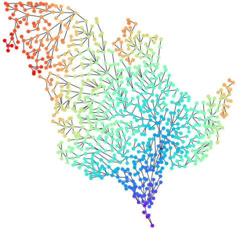
- C++ + alpaka (https://github.com/alpaka-group/alpaka)
  - template-based offloading abstraction layer
  - backends: sequential CPU, OpenMP, CUDA, HIP, ...
  - Application logic decoupled from target execution model by abstraction.
- Sequential + replica stacking
- Parallel + replica stacking
  Conflict resolution:
  1. Generate and store one replacement per node.
  2. Apply all stored replacements.
- Computation of observables on GPU.



[a] http://doi.acm.org/10.1145/2063384.2063405

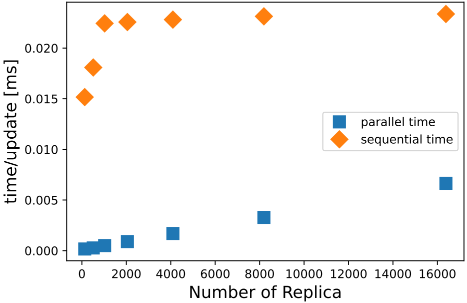
Implementation

- C++ + alpaka (https://github.com/alpaka-group/alpaka)
  - template-based offloading abstraction layer
  - backends: sequential CPU, OpenMP, CUDA, HIP, ...
  - Application logic decoupled from target execution model by abstraction.
- Sequential + replica stacking
- Parallel + replica stacking
  Conflict resolution:
  1. Generate and store one replacement per node.
  2. Apply all stored replacements.
- Computation of observables on GPU.
- Counter-based random number generator
  - for parallel independent/identical random number generation and
  - reproducibility across platforms.
  - Philox[a] implementation available in alpaka.



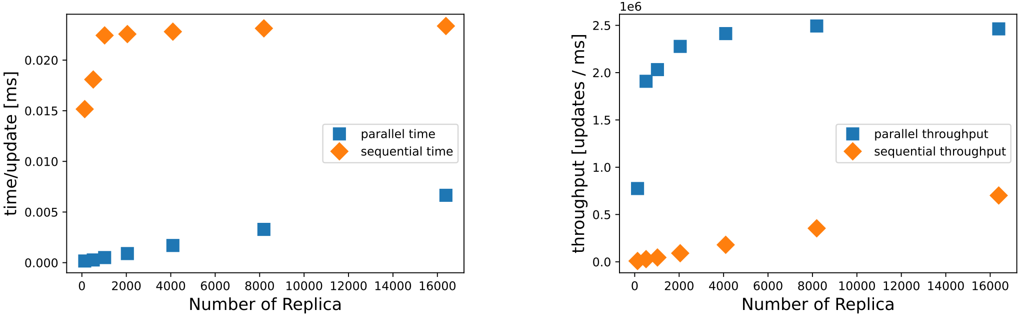
[a] http://doi.acm.org/10.1145/2063384.2063405

Performance



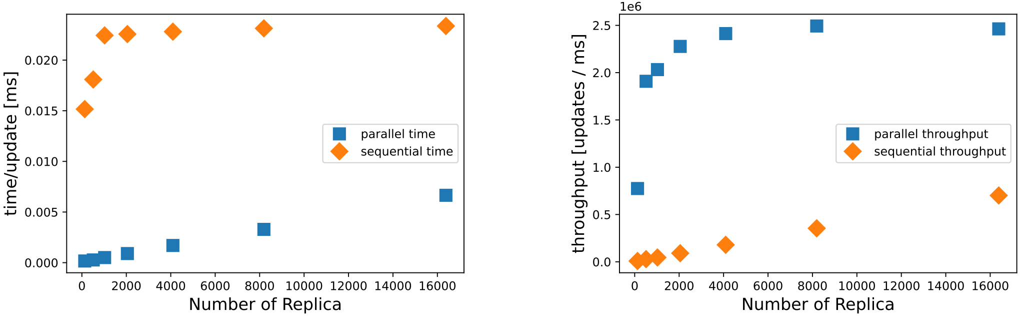
$$N_{\text{graph nodes}} = 313, \ \text{Tesla V100-SXM2}$$

Performance



$N_{\text{graph nodes}} = 313,\ \text{Tesla V100-SXM2}$

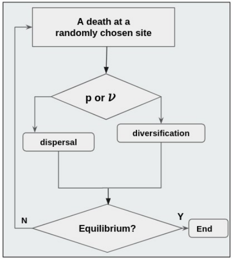
Performance

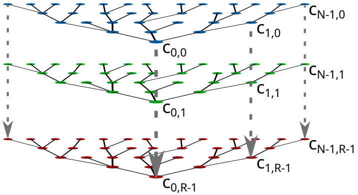


$N_{\text{graph nodes}} = 313, \text{ Tesla V100-SXM2}$

- Code is not finely optimized, no profiling done ... fast enough for now.

Conclusions

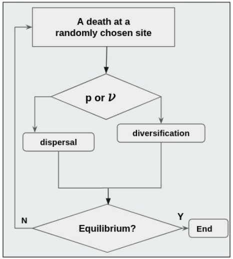


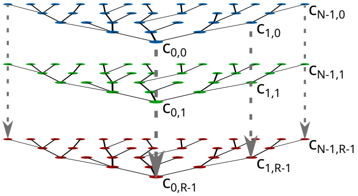


⇒ large speedup for both core simualtion loop and analysis

⇒ streamlined workflow for averages

Conclusions





⇒ large speedup for both core simualtion loop and analysis

⇒ streamlined workflow for averages

- Even when the stated goal is performance, first evaluate what type of performance improvement can help and is feasible in the short-term.
- When changing methods, build a baseline first to show that the change is safe.

Acknowledgments

- CASUS is jointly funded by the Federal Ministry of Education and Research (BMBF) and the Saxon Ministry for Science and Culture.

Thank You.

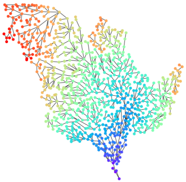
Random-event Generation

How to Select the Origin-Node for migration?

Given

ν probability of diversification event,

$p_{i,j}$ probability for dispersal from node j to node i.

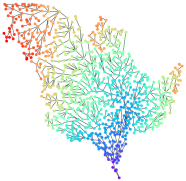


chose random node j → generate random number → accept with $p_{i,j}$ $\circlearrowleft$

– or –

Random-event Generation

How to Select the Origin-Node for migration?



Given

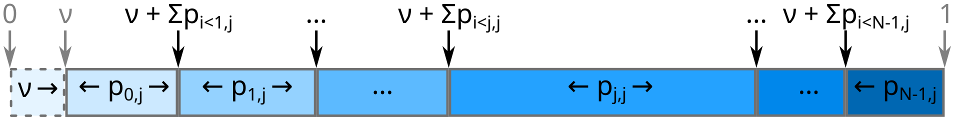
ν probability of diversification event,

$p_{i,j}$ probability for dispersal from node j to node i.

chose random node j → generate random number → accept with $p_{i,j}$ ↺

– or –

1. Prepare prefix sum of event probabilities.
2. Generate rantom number and search for corresponding interval.



0 $\quad \nu \quad \nu + \Sigma p_{i<1,j} \quad \ldots \quad \nu + \Sigma p_{i<j,j} \quad \ldots \quad \nu + \Sigma p_{i<N-1,j} \quad 1$

$\nu \rightarrow$ | $\leftarrow p_{0,j} \rightarrow$ | $\leftarrow p_{1,j} \rightarrow$ | … | $\leftarrow p_{j,j} \rightarrow$ | … | $\leftarrow p_{N-1,j}$