Accelerating the Lagrangian particle tracking in hydrology to continental-scale

Chen Yang

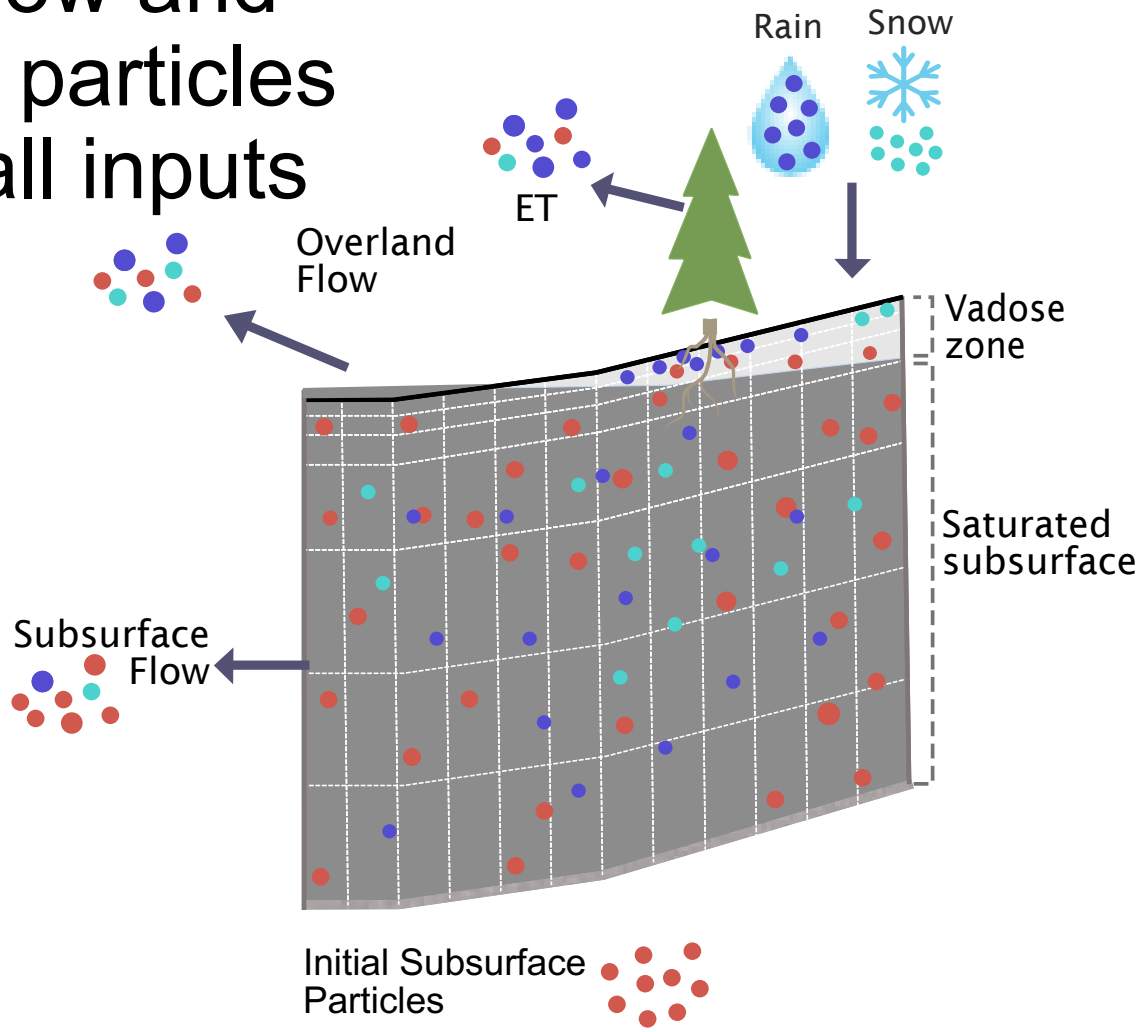
Princeton University

IDEAS Watersheds

LANL IDEAS Watersheds Project
LBNL Watershed Function Project

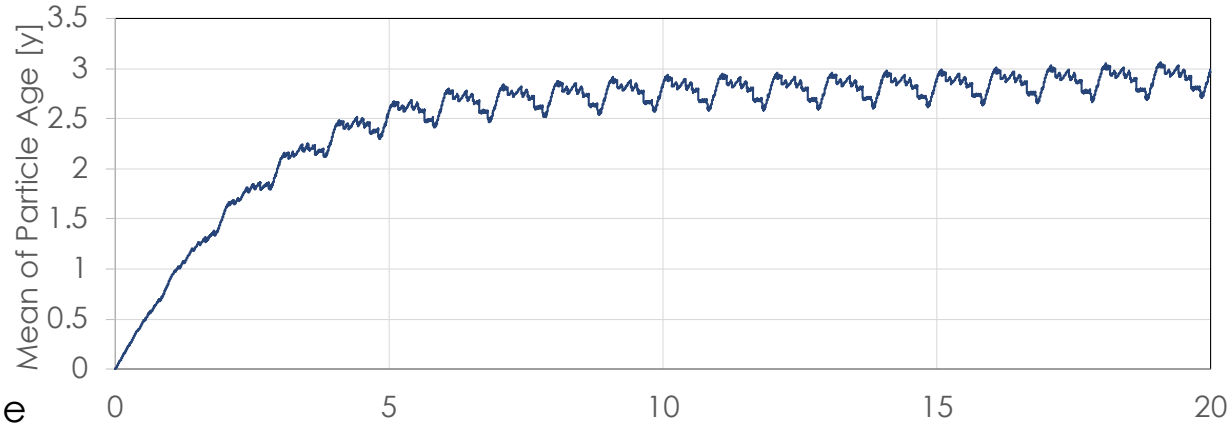
EcoSLIM reads ParFlow and CLM output and adds particles dynamically tracking all inputs

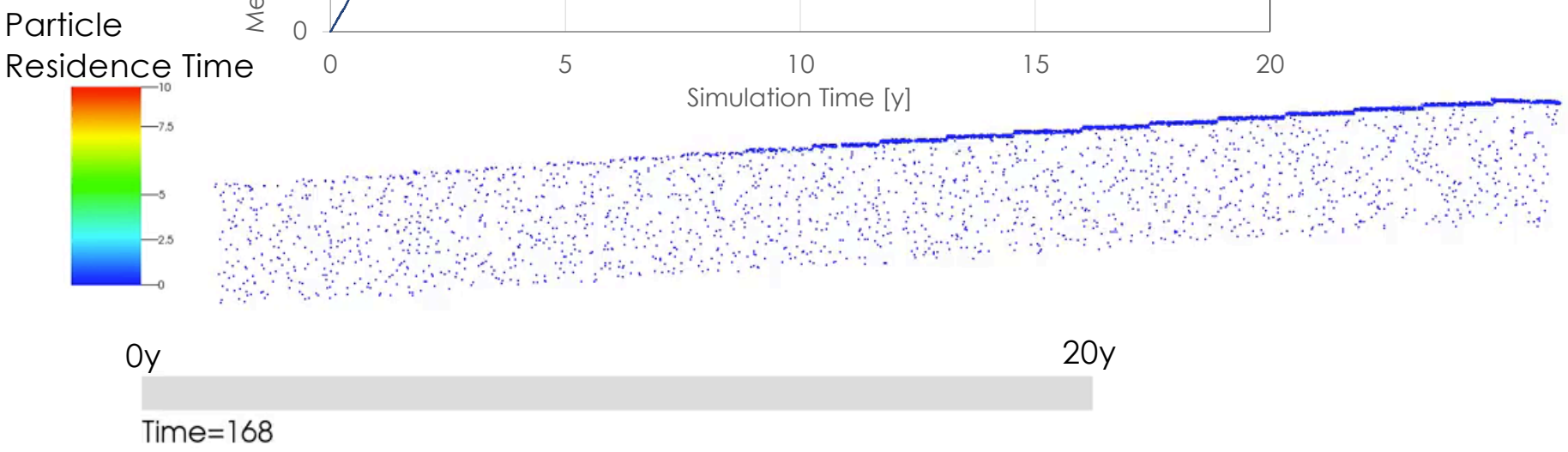
- Lagrangian particle tracking approach
- Tracking of rain, snow, groundwater throughout the model domain
- ET removes particles from the root zone in a formulation like partially-penetrating wells



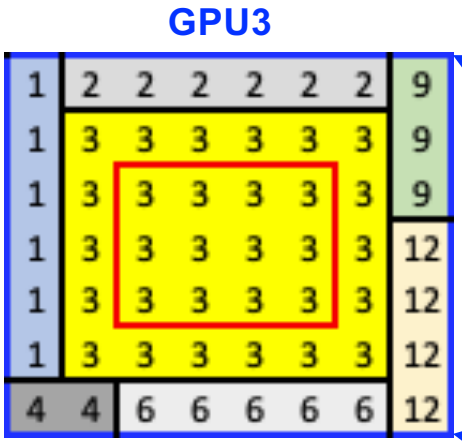
Maxwell et al Ecohydrology 2019

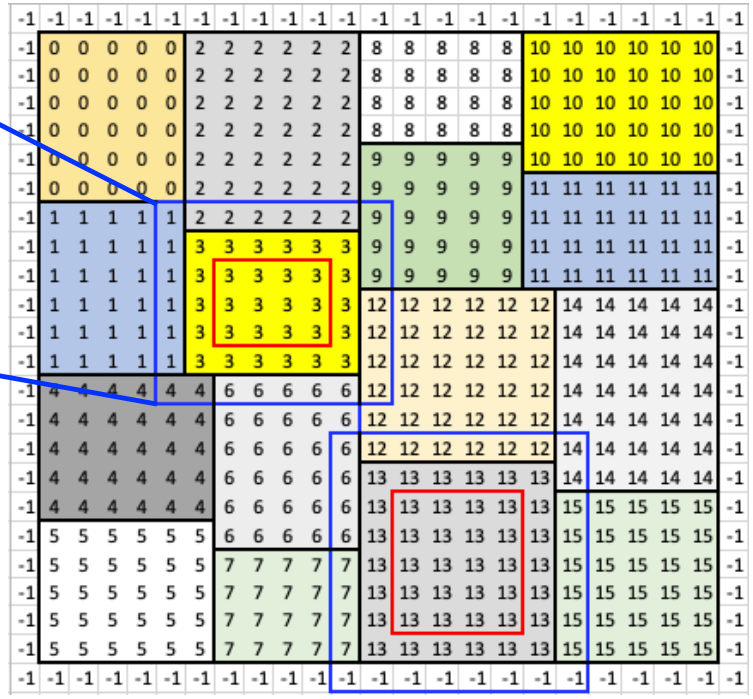
Ages evolve dynamically over the simulation and can be ‘spun up’





Domain decomposition





modeling domain

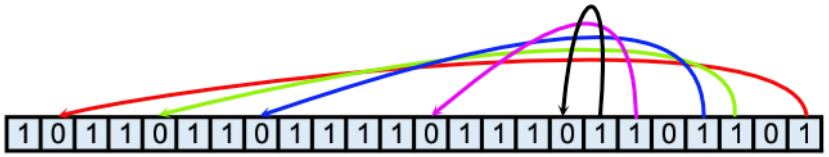
- Decomposition is only in horizontal direction
- Each GPU is responsible for one subdomain
- The yellow area is the real subdomain labeled by the GPU rank
- One more grid-cell around is halo cells labeled by the ranks of neighbor GPUs
- Particles moving into halo cells will be transferred to the labeled neighbors
- Transfer can be conducted every or multiple timestep(s)

Particle transfer

- **Packed transfer**
- **One-by-one transfer**

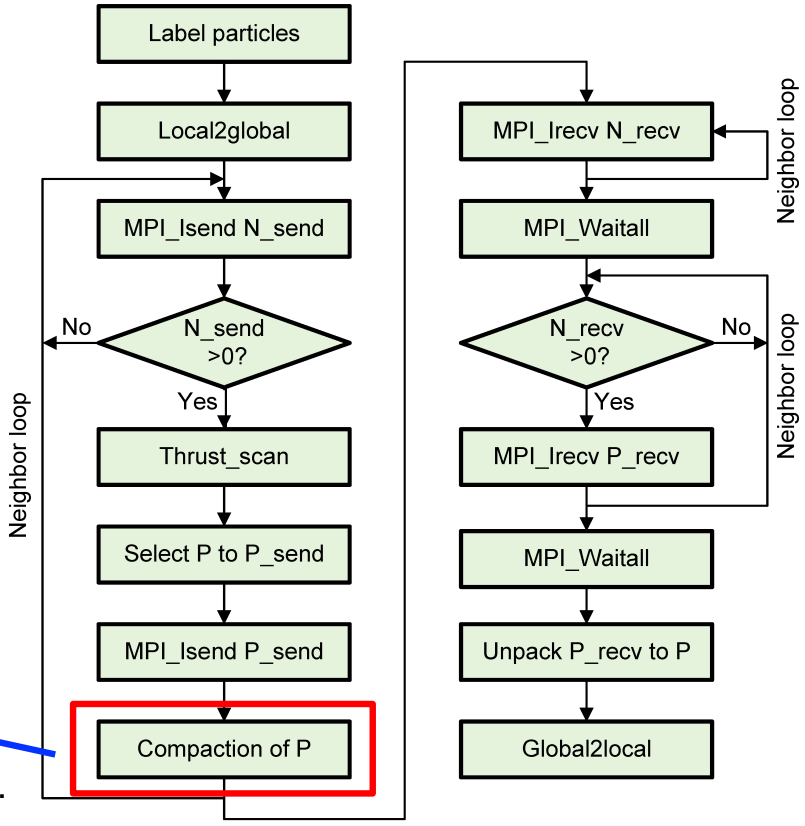
2D Total particle array

Each row represents a particle with several attributes.



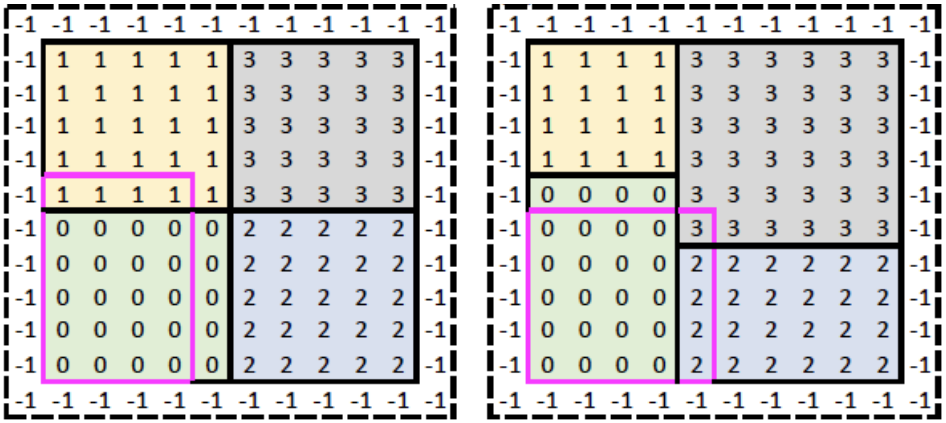
Total P array compaction, one cell represents one particle.

1 represent active particle and 0 represents inactive particle.

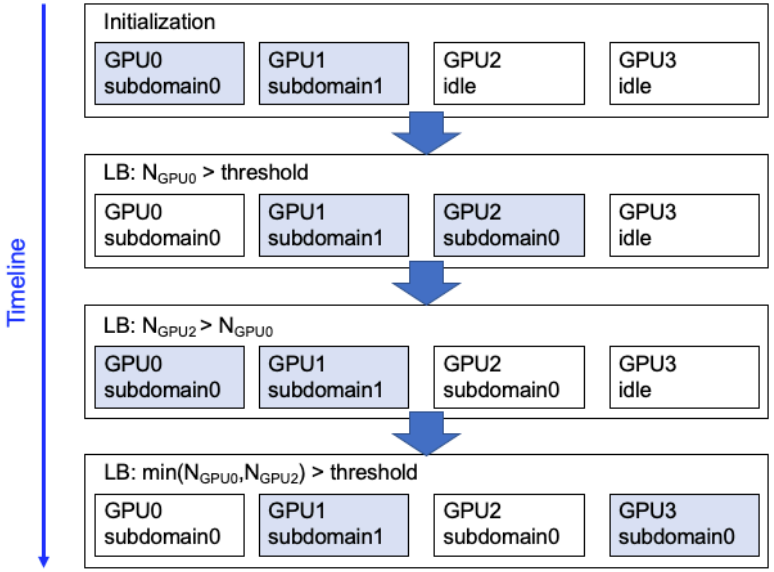


Packed transfer

Load balancing

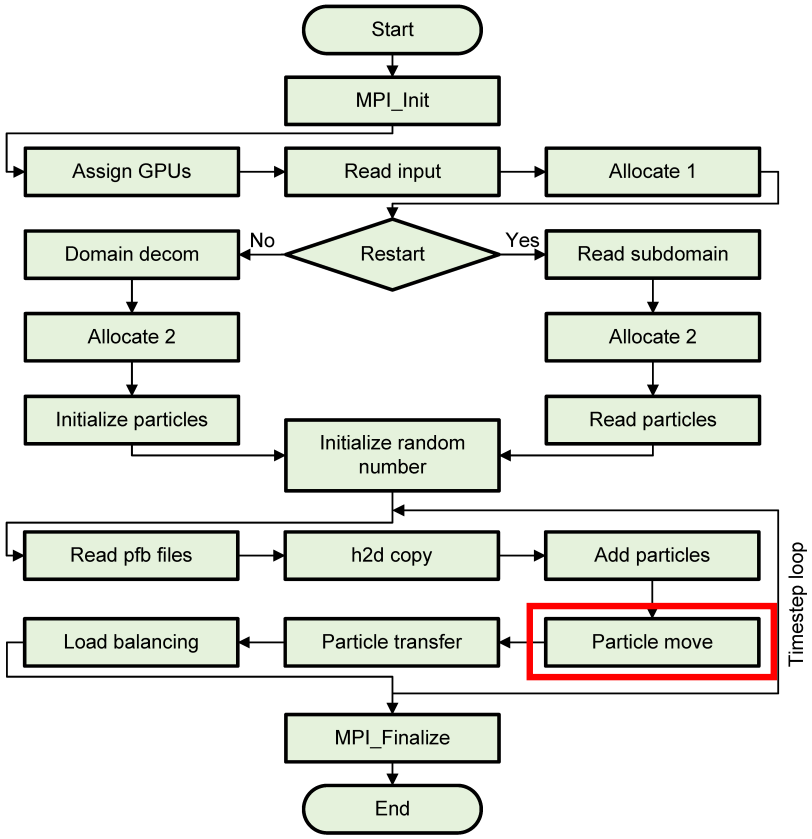


Dynamic domain decomposition



GPU help

Memory optimization



Flow chart of parallelized EcoSLIM

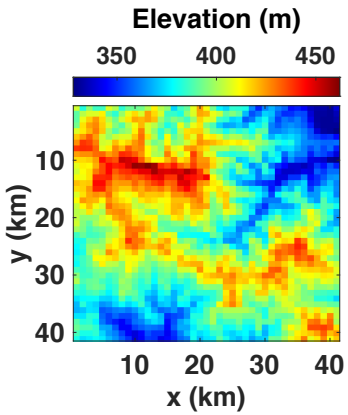
CUDA Fortran

- No third-party libraries for optimization purpose
- Avoid the bank conflict in the access of P array
- Store the gridded data (read only) in texture memory
- Store some most frequently used gridded data in shared memory
- Restrict the number of registers used, but for readability of the code, this way cannot be overused.
- Copy the state of pseudorandom number generator (derived type) from global to register

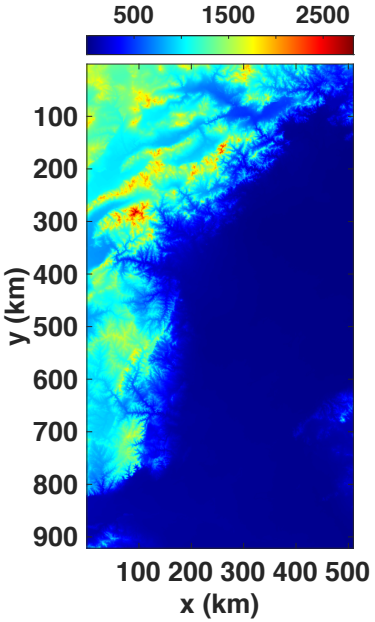
```
type(curandStateXORWOW):: hh   register

!-------------------------------------
ii = offset + (blockIdx%x - 1) * blockDim%x + threadIdx%x
ii_l = threadIdx%x

hh = handle(ii)  global
!-------------------------------------
```

Test domains

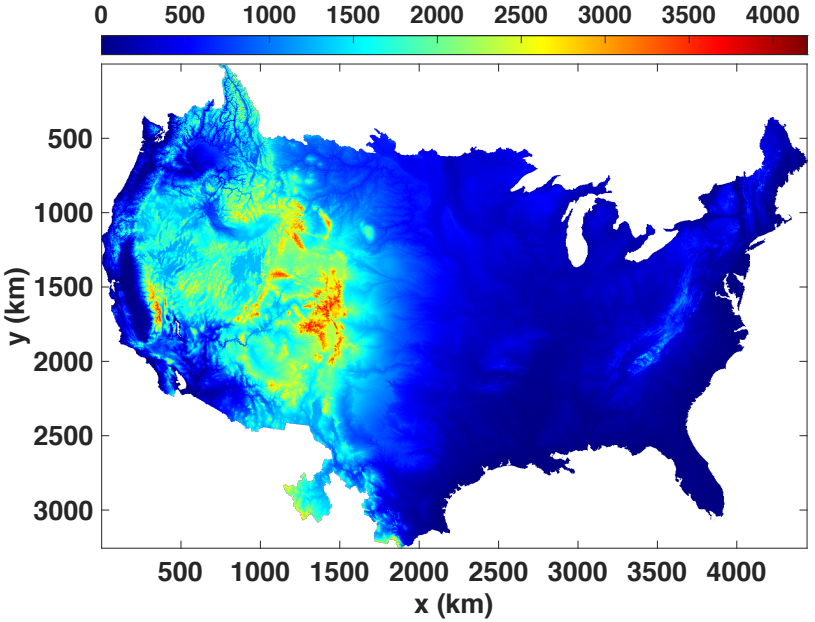
The Little Washita



The North China Plain



The Continental US

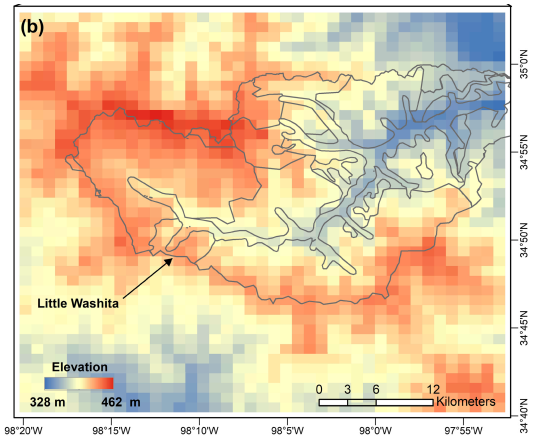


Tests across three spatial scales

Test results: Little Washita

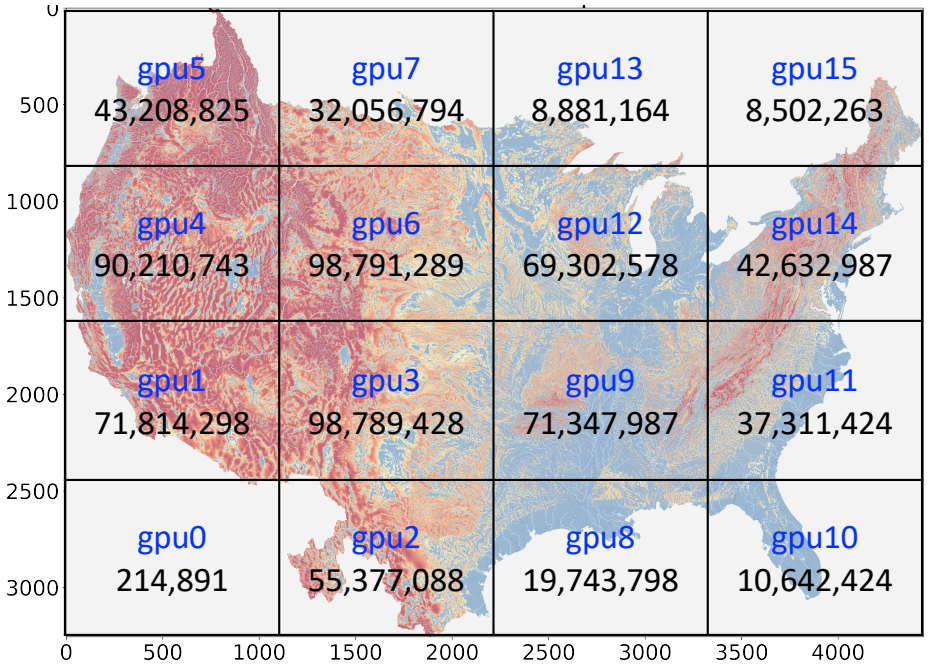
Della-GPU cluster at Princeton University:

- *Two NVIDIA A100 GPUs per node*
- *Two 2.60-GHz AMD EPYC 7H12 sockets per node*
- *Each socket has 64 cores without hyper-threading*
- *criterion: time-128threads/time-2GPU >= 4*
- *criterion: time-128threads/time-4GPU >= 8*
- *24000 hourly timestep, total time of particle loop+sort*

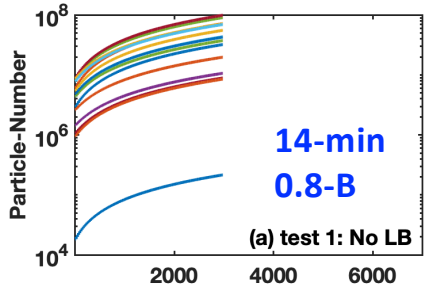


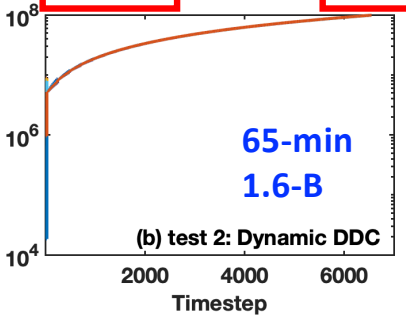
	Speedup	Particle Transfer	Load Balancing
test1(2GPU)	13.73 (>4)	Yes	Yes
test2(4GPU)	25.49 (>8)	Yes	Yes
test3(4GPU)	12.72	No	No

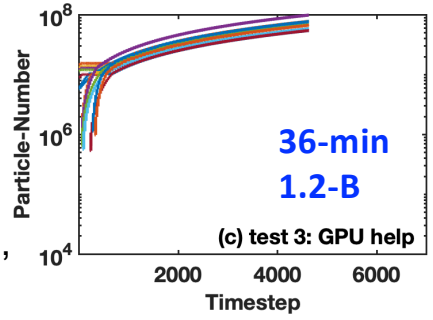
Test results: Continental US



Test results on the Continental US								
Tests	Started GPUs	Planned GPUs	p	q	Injected number	LB	LB frequency	Wall-clock
Test1	16	16	4	4	1 per day	/	/	14-min (0.8 B)
Test2	16	16	4	4	1 per day	LB1	240 hours	65-min (1.6 B)
Test3	8	16	4	2	1 per day	LB2	24 hours	36-min (1.2 B)







hourly timestep

Preliminary run using 16GPUs, severe load imbalance

The maximum number of particles on each GPU is 100 million, so the total capacity of 16 GPUs is 1.6 billion.

Contact me

email: cy15@princeton.edu

github: https://github.com/aureliayang/EcoSLIM_CONUS

Welcome comments and questions!

IDEAS Watersheds

LANL IDEAS Watersheds Project
LBNL Watershed Function Project