Parallel and Distributed Systems [PDS_4717]

Game of Life with MPI

Stijn Jacobs, Simon Wilmots

De opleiding industrieel ingenieur is een gezamenlijke
opleiding van UHasselt en KU Leuven

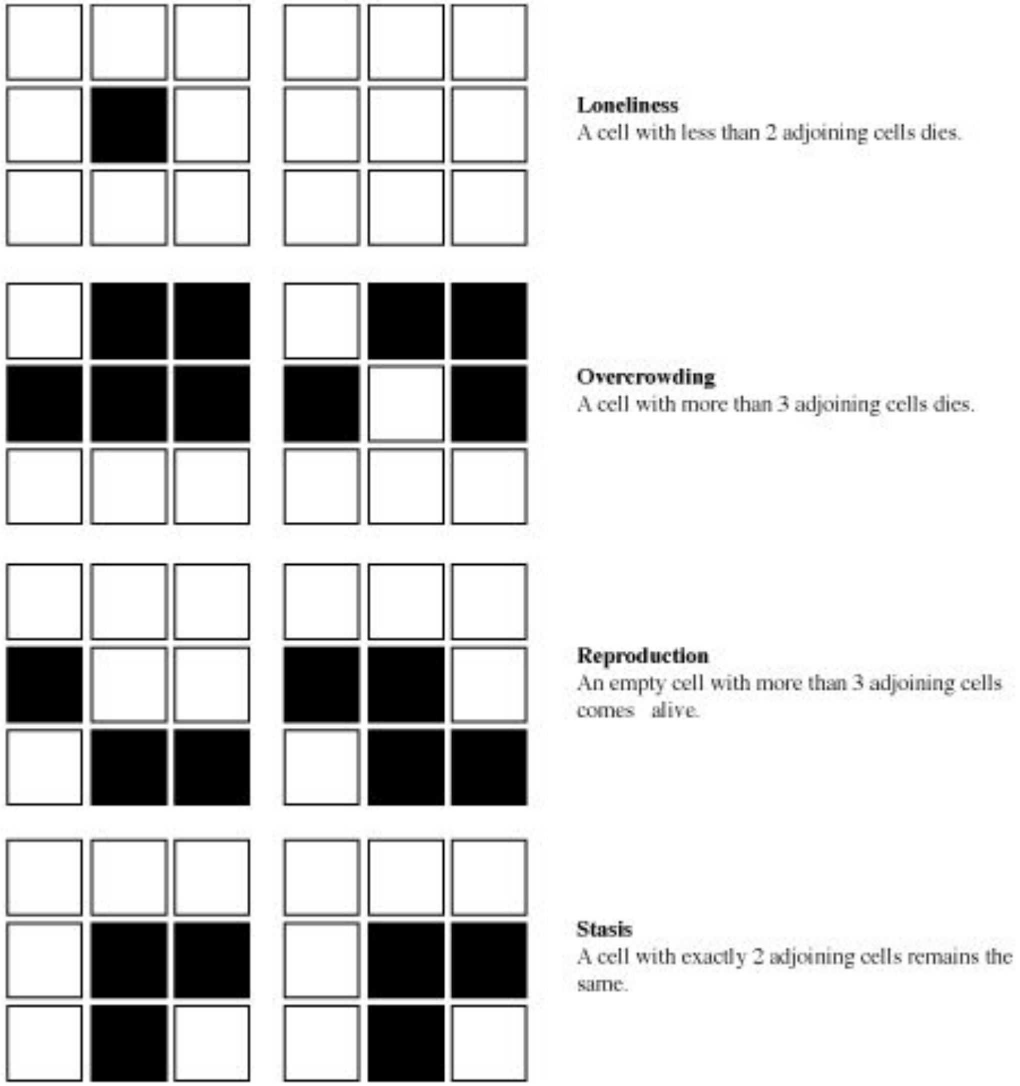
Game of Life

Rules

- Underpopulation
- Overcrowding
- Survival
- Reproduction

Core design choices

- Load file with start layout
- Grid as 1D vector with helper function
    - 2D vector not dynamic with nr. of processes
- Wrap around with boundary conditions
- Exploration of parallelization and scalability



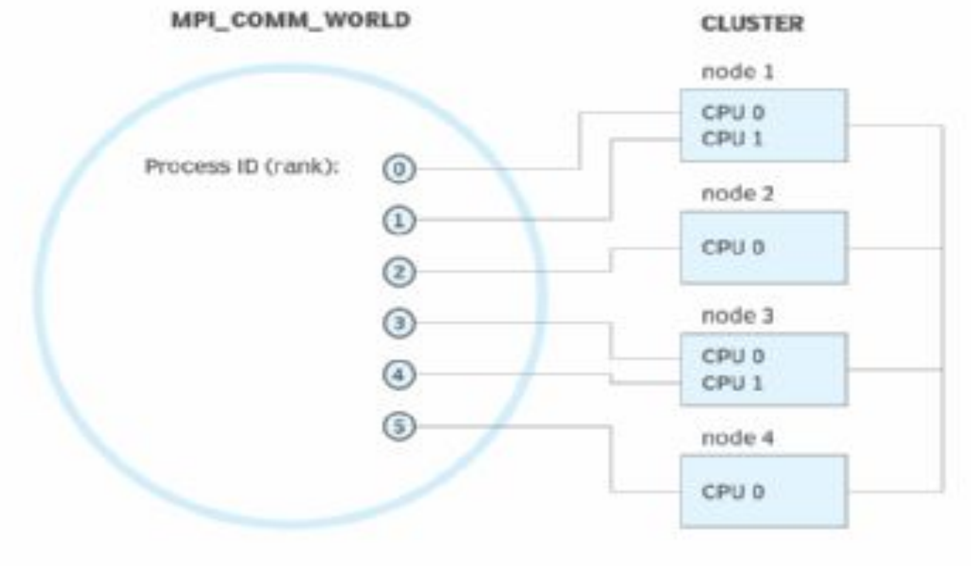
Message Passing Interface (MPI)

Why should you use MPI

- Explicit control over communication & nr. of processes
- Platform-agnostic (i.e. CUDA = NVIDIA)

Why not MPI

- Explicit message passing is complex (error-prone)
- More complexity/overhead as OpenMP



Implementation

Serial

- Load begin state
- iterate over all fields per generation
- Main keeps all data

Parallel

<u>Regular Game of Life</u>

- Load begin state
- Row decomposition to processes
- 2 **read-only buffer-rows**
- Helpers clear **one** generation & send data back
- Main re-collects & re-distributes

<u>Buffered Game of Life</u>

- Load begin state
- Row decomposition to processes with **extra processing buffer**
- Helper clear **multiple** generations, shrink gradually & send data back
- Main re-collects & re-distributes

Running on VSC & Limitations

- Seperate SLURM-file for each node & task-count
- Max 4 nodes
- Max 74 processes per node
- Task distribution: not always evenly, but shouldn’t matter

```
SLURM_JOB_ID: 63278551        You, 17 ho
SLURM_JOB_USER: vsc36706
SLURM_JOB_ACCOUNT: lp_h_pds_iiw
SLURM_JOB_NAME: 3_node_05_tasks.slurm
SLURM_CLUSTER_NAME: wice
SLURM_JOB_PARTITION: batch
SLURM_NNODES: 3
SLURM_NODELIST: m33c22n[2,4],m33c25n2
SLURM_JOB_CPUS_PER_NODE: 3,1(x2)
Date: Mon Dec  9 21:44:36 CET 2024
Walltime: 00-01:00:00
```

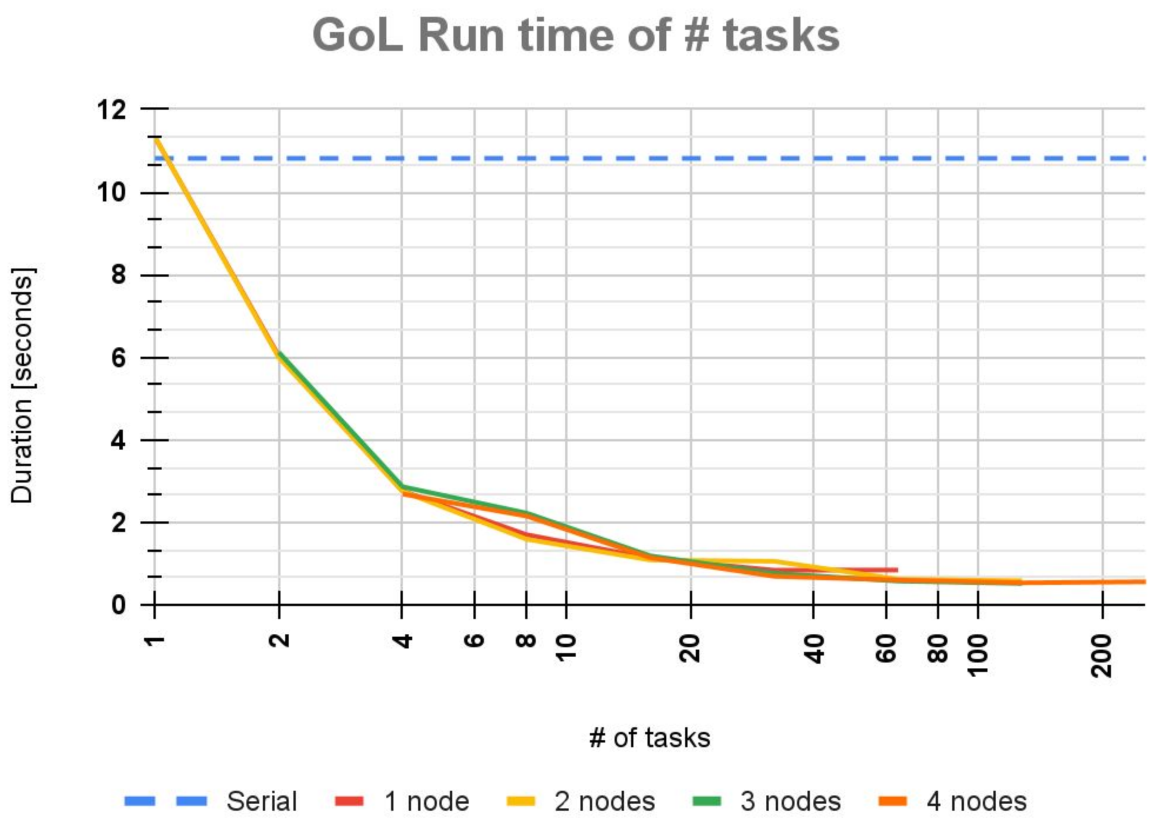
VSC Parameters

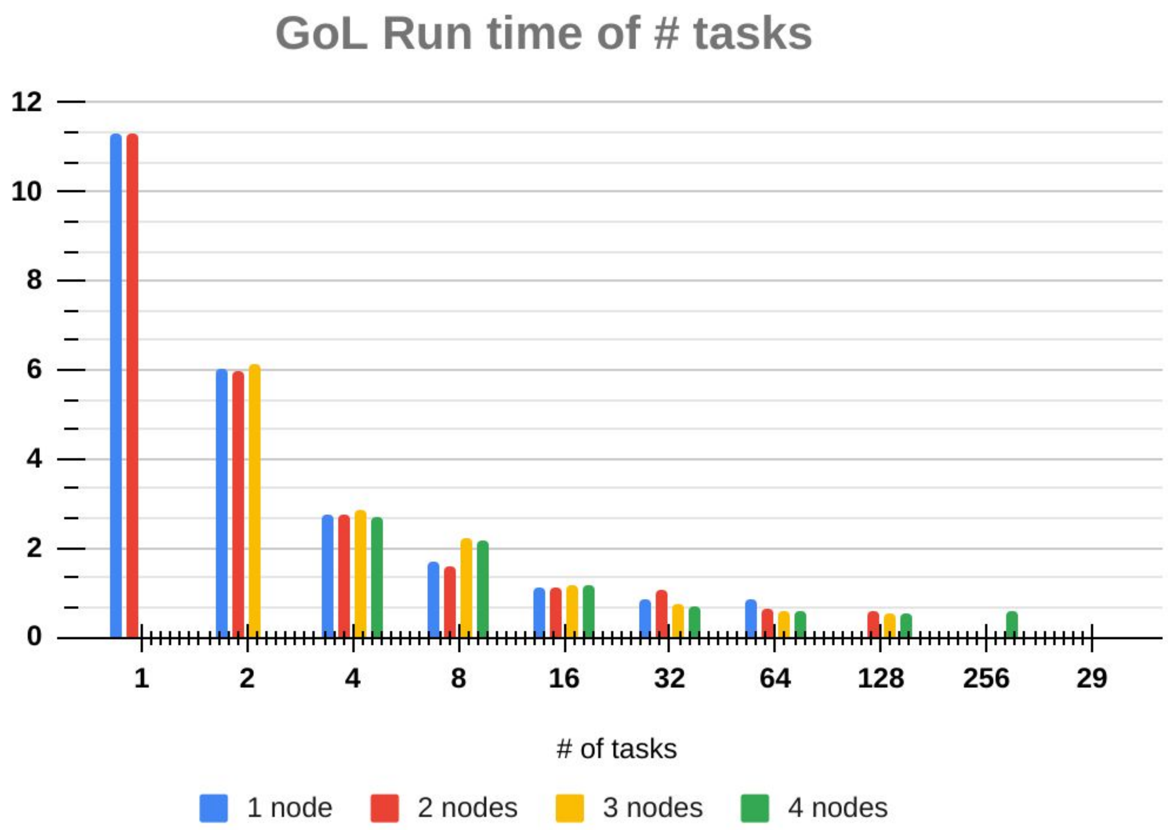
- --nodes = x {1, 2, 3, 4}
- --ntasks = $2^x + 1$
- --cpus_per_tasks = 1

```
✓ [Dec/10 10:17] vsc36726@r23i27n01 /vsc-hard-mounts/leuven-user/367/vsc36726/PDS_practicum/GameOfShenanigans $ ./slurmidilidocious.sh
Submitted batch job 63279103 on cluster wice
Submitted batch job 63279104 on cluster wice
Submitted batch job 63279105 on cluster wice
Submitted batch job 63279106 on cluster wice
Submitted batch job 63279107 on cluster wice
Submitted batch job 63279108 on cluster wice
Submitted batch job 63279109 on cluster wice
Submitted batch job 63279110 on cluster wice
Submitted batch job 63279111 on cluster wice
Submitted batch job 63279112 on cluster wice
Submitted batch job 63279113 on cluster wice
Submitted batch job 63279114 on cluster wice
Submitted batch job 63279115 on cluster wice
Submitted batch job 63279116 on cluster wice
Submitted batch job 63279117 on cluster wice
Submitted batch job 63279118 on cluster wice
Submitted batch job 63279119 on cluster wice
Submitted batch job 63279120 on cluster wice
Submitted batch job 63279121 on cluster wice
Submitted batch job 63279122 on cluster wice
Submitted batch job 63279123 on cluster wice
Submitted batch job 63279124 on cluster wice
✓ [Dec/10 10:17] vsc36726@r23i27n01 /vsc-hard-mounts/leuven-user/367/vsc36726/PDS_practicum/GameOfShenanigans $
```

RESULTS

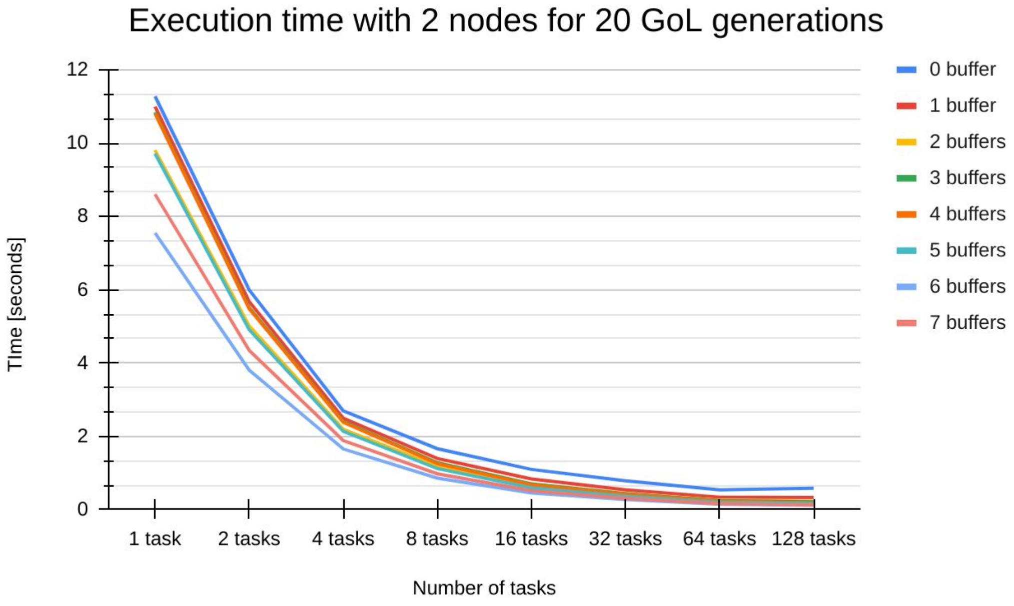
Results – Parallel (1 - Regular GoL)





Results – Parallel (2 - Buffered GoL)

Execution time with 2 nodes for 20 GoL generations [seconds]								
Nr. of buffer rows in function of nr. of tasks	1 task	2 tasks	4 tasks	8 tasks	16 tasks	32 tasks	64 tasks	128 tasks
0	11.2762	5.99837	2.69679	1.66354	1.09974	0.78666	0.541712	0.584371
1	10.9976	5.67831	2.49766	1.39567	0.838987	0.539332	0.342648	0.335333
2	9.80647	5.01307	2.19323	1.18628	0.671896	0.426691	0.253332	0.23031
3	10.8373	5.51604	2.40498	1.2807	0.704008	0.442415	0.25098	0.214147
4	10.8039	5.48139	2.38632	1.25563	0.68144	0.423323	0.234835	0.190403
5	9.71294	4.91305	2.13709	1.12028	0.599018	0.369784	0.198715	0.163828
6	7.5485	3.80696	1.65616	0.862754	0.459138	0.280525	0.149048	0.119304
7	8.60382	4.34511	1.88655	0.980076	0.517766	0.313977	0.16542	0.129935



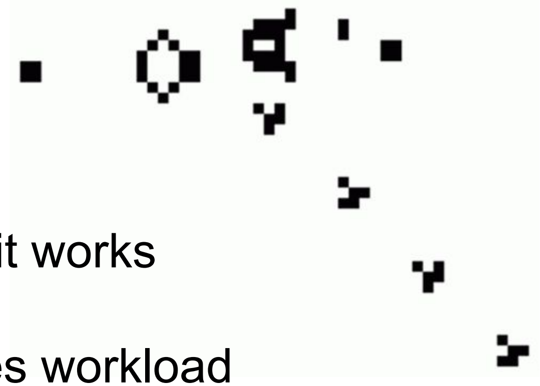
Relative execution time per MPI task, corrected for inaccurate generation count (3 nodes)								
Nr. of buffer rows in function of nr. of tasks	Calculated generations	2 tasks	4 tasks	8 tasks	16 tasks	32 tasks	64 tasks	128 tasks
0	20	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
1	20	94.78%	91.36%	87.03%	81.67%	69.14%	62.32%	56.99%
2	18	93.09%	88.84%	82.71%	75.46%	59.62%	50.76%	42.94%
3	20	92.26%	87.55%	80.84%	72.99%	54.54%	44.98%	36.45%
4	20	91.76%	86.91%	79.76%	71.24%	52.98%	41.64%	32.87%
5	18	91.41%	86.41%	79.07%	70.05%	51.61%	39.90%	30.22%
6	14	91.21%	85.40%	78.59%	69.02%	51.07%	38.51%	29.67%
7	16	91.05%	85.17%	78.27%	68.49%	50.44%	37.19%	27.83%

Relative execution time per buffer size, corrected for inaccurate generation count (3 nodes)							
Nr. of buffer rows in function of nr. of tasks	2 tasks	4 tasks	8 tasks	16 tasks	32 tasks	64 tasks	128 tasks
0	100.00%	45.95%	25.81%	21.53%	12.31%	9.39%	10.01%
1	100.00%	44.29%	23.70%	18.55%	8.98%	6.18%	6.02%
2	100.00%	43.85%	22.93%	17.45%	7.89%	5.12%	4.62%
3	100.00%	43.60%	22.61%	17.03%	7.28%	4.58%	3.95%
4	100.00%	43.52%	22.44%	16.71%	7.11%	4.26%	3.58%
5	100.00%	43.43%	22.33%	16.50%	6.95%	4.10%	3.31%
6	100.00%	43.02%	22.24%	16.29%	6.89%	3.97%	3.26%
7	100.00%	42.98%	22.19%	16.19%	6.82%	3.84%	3.06%

Shortcoming & future work

- Personal Coding Skills
- Enforcing cores on specific nodes
- Non-linebased data distribution
- Irregular process count support
- Separate timing of communication time & processing time

Conclusion



General

- Both process count & buffersize have a meaningful impact
- Efficiency boost of process count has diminishing returns.

MPI

- Hard, but neat when it works
- Efficiently redistributes workload
- Communication can be more expensive than computation
- Choose OpenMP for DevX
- Think before you share

Parallel and Distributed Systems [PDS_4717]

Game of Life with MPI

Stijn Jacobs, Simon Wilmots

De opleiding industrieel ingenieur is een gezamenlijke
opleiding van UHasselt en KU Leuven