IEEE CAD/Graphics’09
August 20, 2009

Volume Rendering using Grid Computing for Large-Scale Volume Data

Kunihiko Nishihashi, Toru Higaki, Kenji Okabe
Bisser Raytchev, Toru Tamaki, Kazufumi Kaneda

Hiroshima University, Japan

Background

Volume Rendering

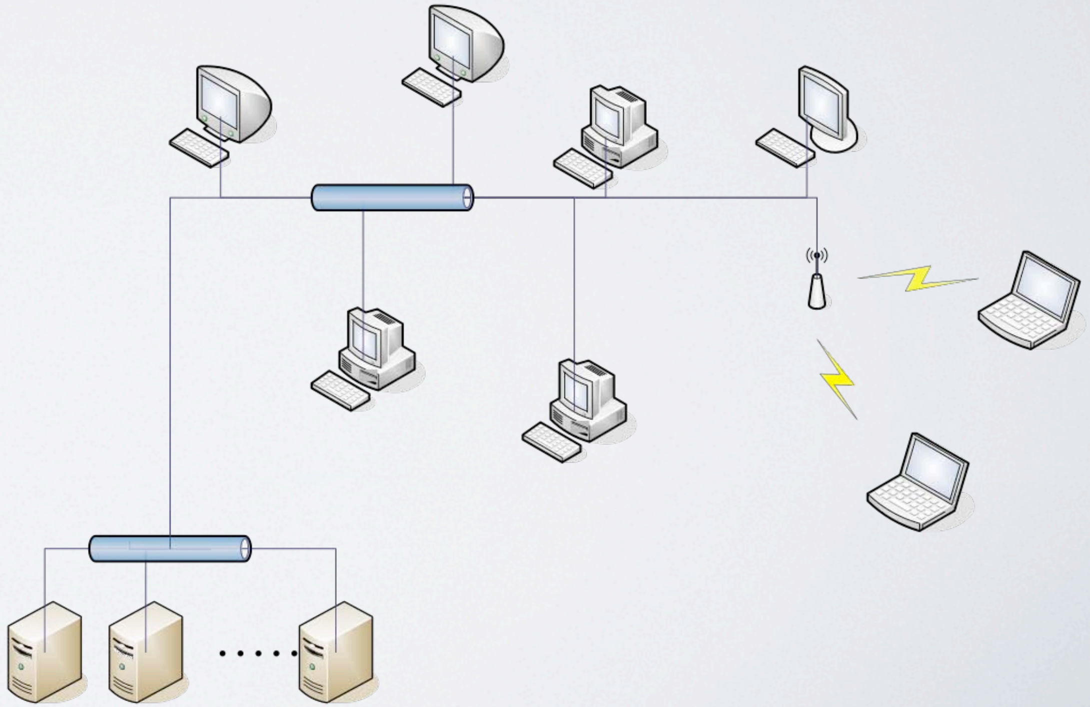
- Useful in the medical field
- Higher resolution of volume data
- Increase in computational costs

Fast Volume Rendering!!

Background

Grid Computing

- Effective use of a large number of idle computers
- Low cost
- Scalability of computational power



Goal

Fast volume rendering using grid computing!!

In a grid:

Computing resources often changing over time

➡ The return order of results rarely matches the sending order.

➡ Less efficient in sequential job-scheduling

⬇

Dynamic Job-scheduling

Contents

- Related Work
- Proposed Method
  - Disadvantages of Sequential Job-Scheduling
  - Obstacle-Flag
  - Dynamic Job-Scheduling
- Exception Handling
- Experimental Results
- Conclusions

Related Work

- **GPU**: [Callahan ’05], [Hofsetz ’08], [Keles ’06]
  - GPU based calculation
  - Visibility sorting, texture slab, z-occlusion culling
- **PC Clusters**: [Matsui ’04], [Lacroute ’96], [Stompel ’03]
  - Real time volume rendering of a 1024^3 volume data
  - Parallel image compositing algorithm
- **Grid Computing**: [Alfonso ’05], [Norton ’03], [Bethel ’03]
  - Visibility-driven compression schemes
  - Connectionless protocols

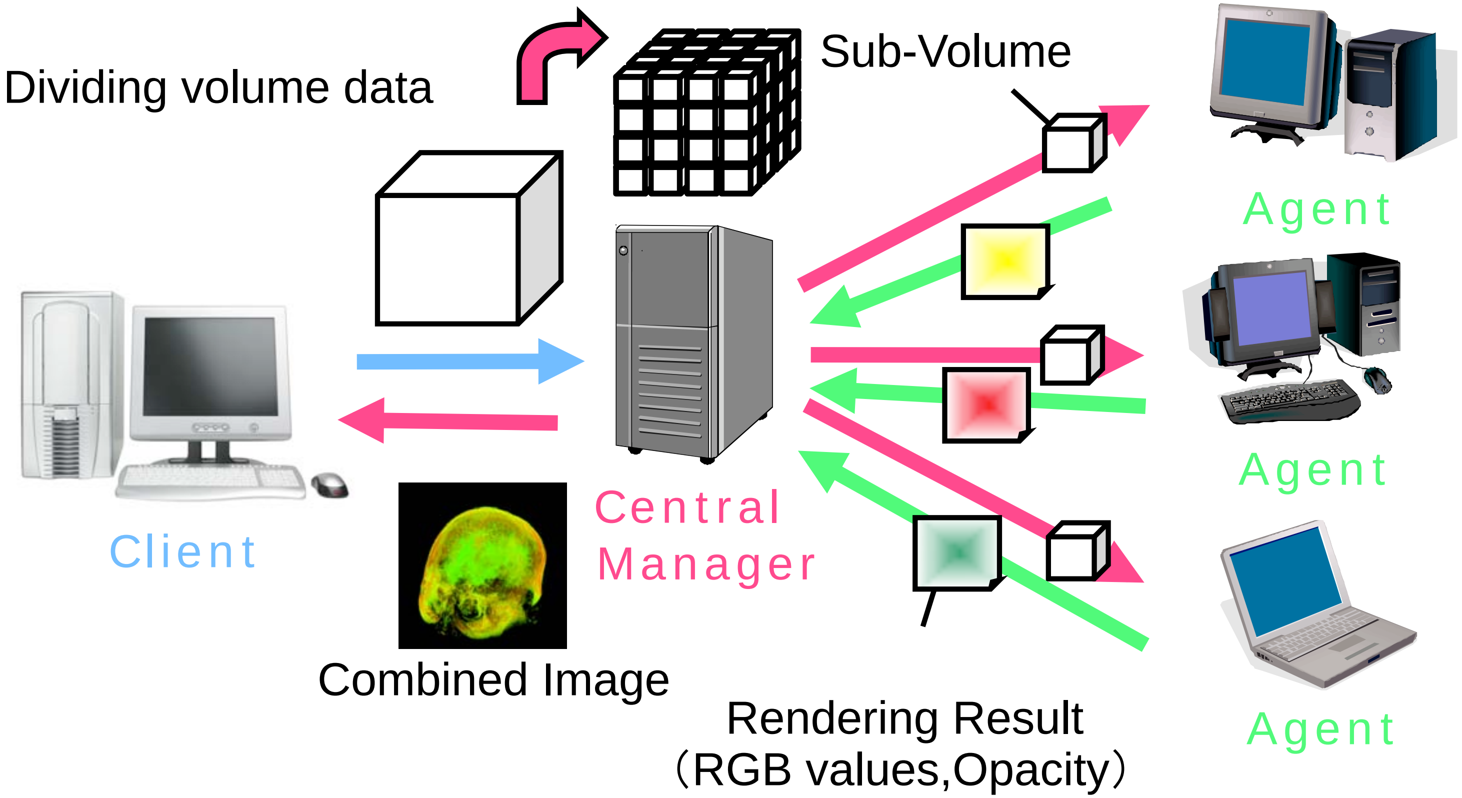
System

Client: Send a volume rendering request to a server.

Server: Sending jobs, combining of rendering results, etc...

Agent: Volume rendering

System Configuration



Proposed Method

Obstacle-Flag

- Management the occlusion state of the sub-volume

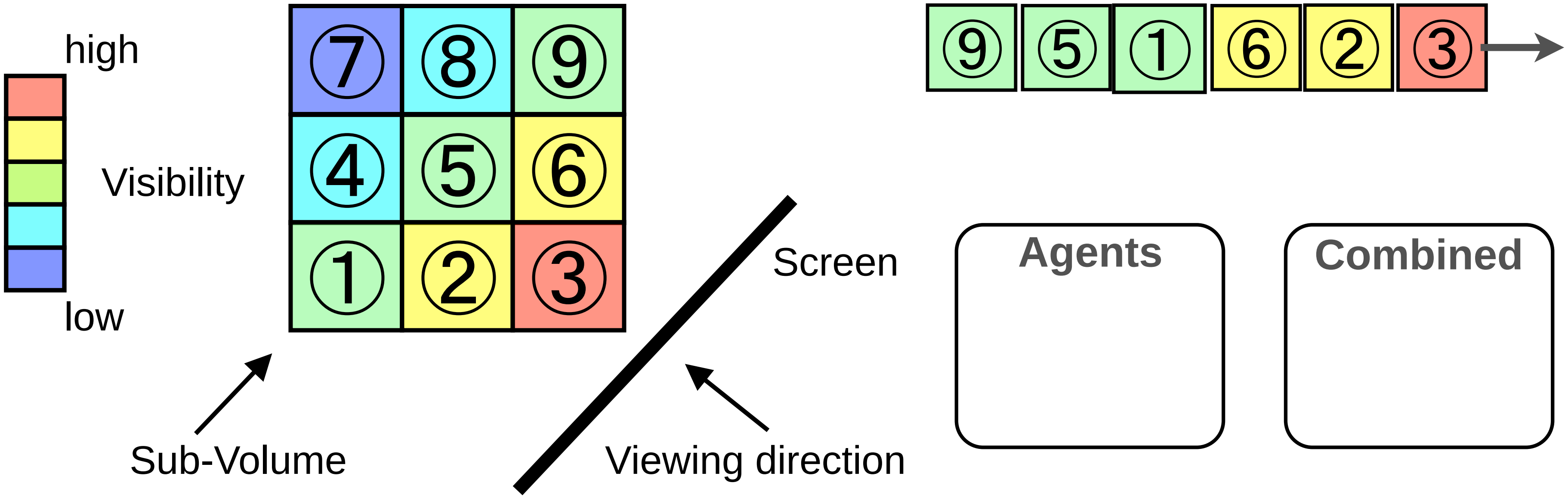
Dynamic Job-Scheduling

- Updating the obstacle-flags

  ➡ Determining dynamically sending order of sub-volumes

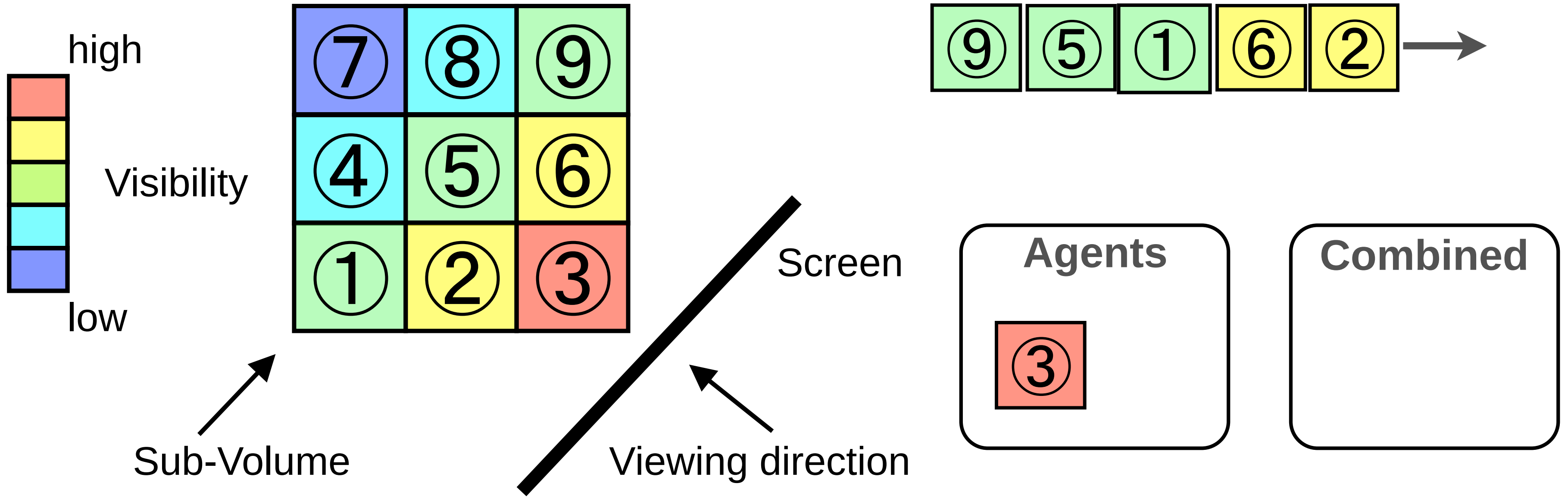
Disadvantages of Sequential Job-Scheduling

Sequential Job-Scheduling



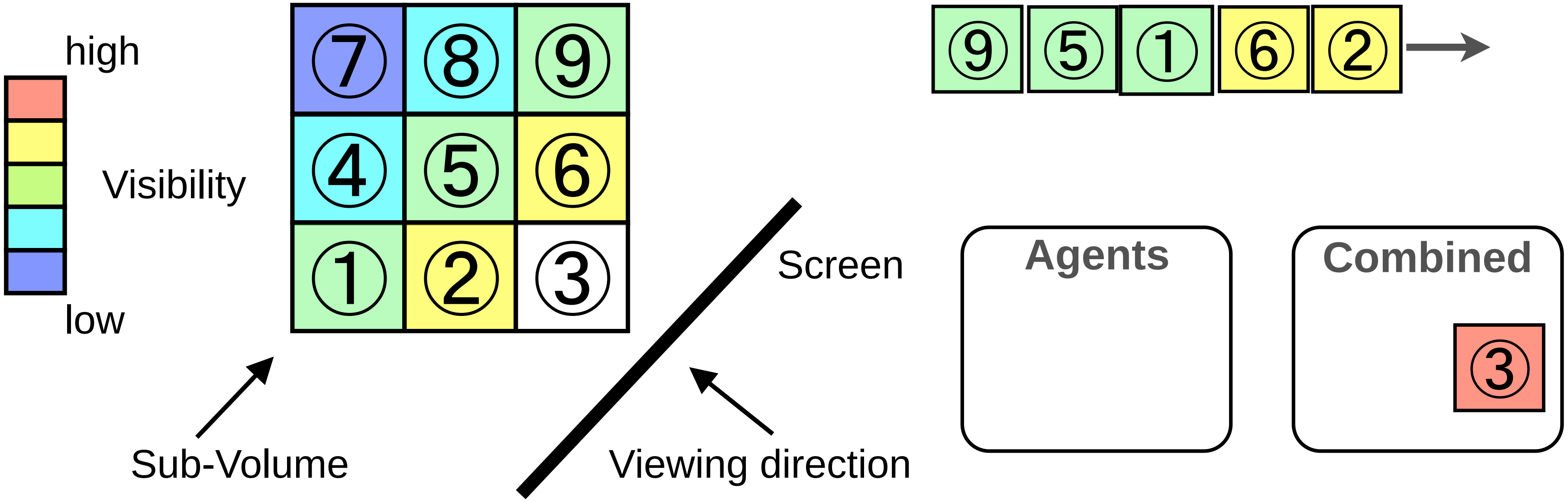
Disadvantages of Sequential Job-Scheduling

Sequential Job-Scheduling



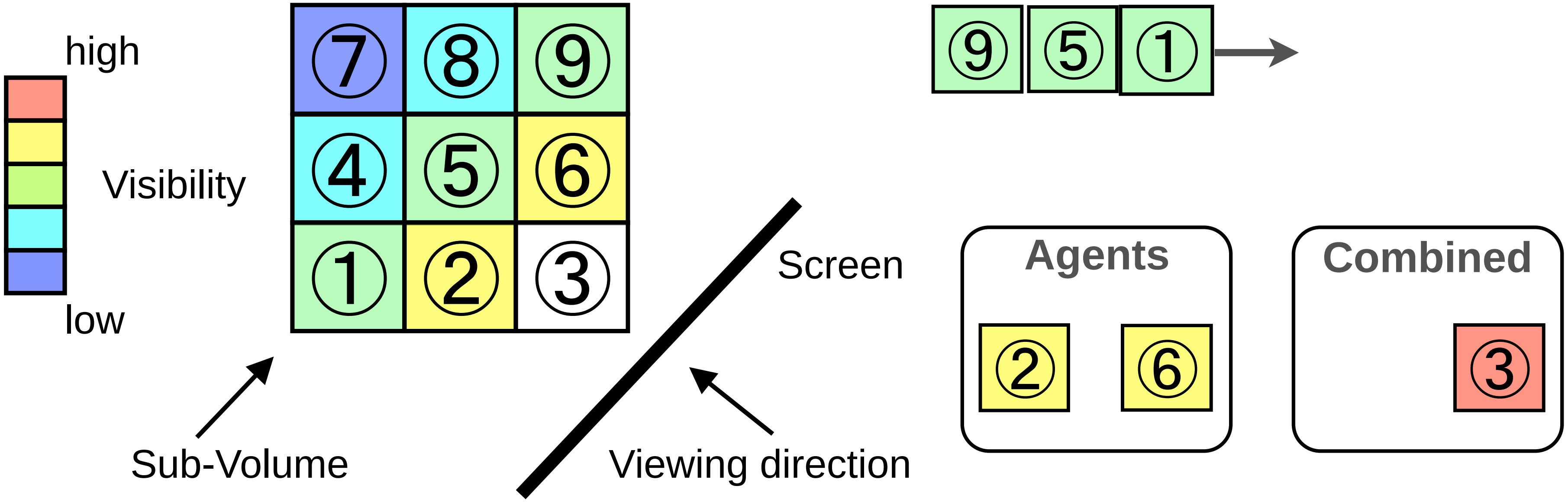
Disadvantages of Sequential Job-Scheduling

Sequential Job-Scheduling



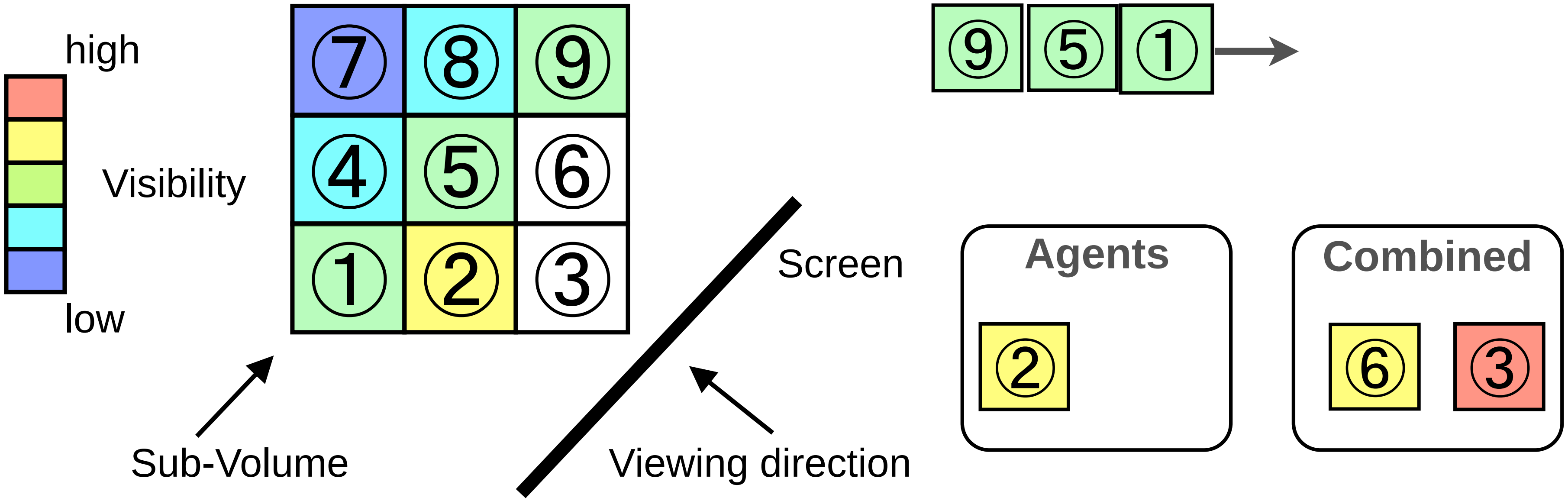
Disadvantages of Sequential Job-Scheduling

Sequential Job-Scheduling



Disadvantages of Sequential Job-Scheduling

Sequential Job-Scheduling



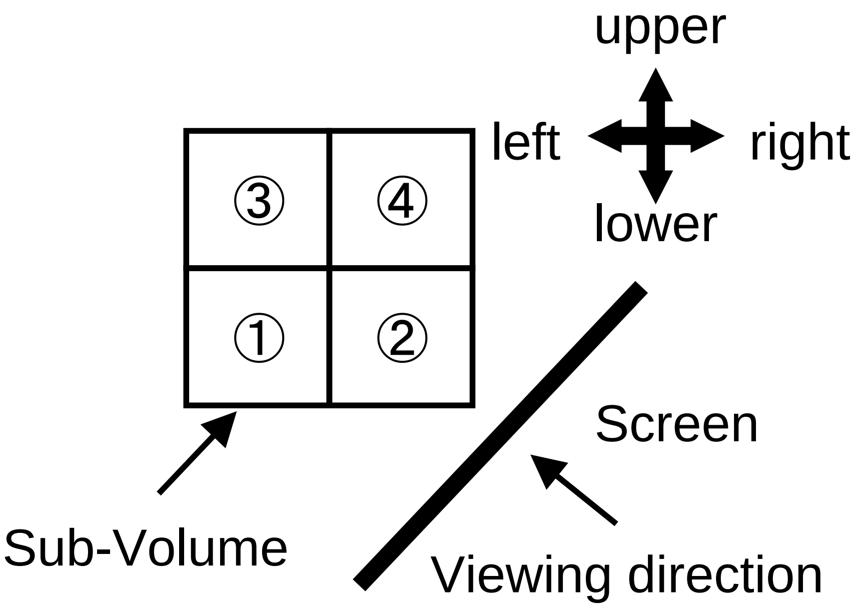
Can not be sent sub-volume ⑨!!

Obstacle-Flag

The obstacle-flags manage the relationships between sub-volumes.

- Need 4 bits
- Occluded: **1**, Not occluded: **0**
- All zero → No occluding sub-volumes

Obstacle-Flag

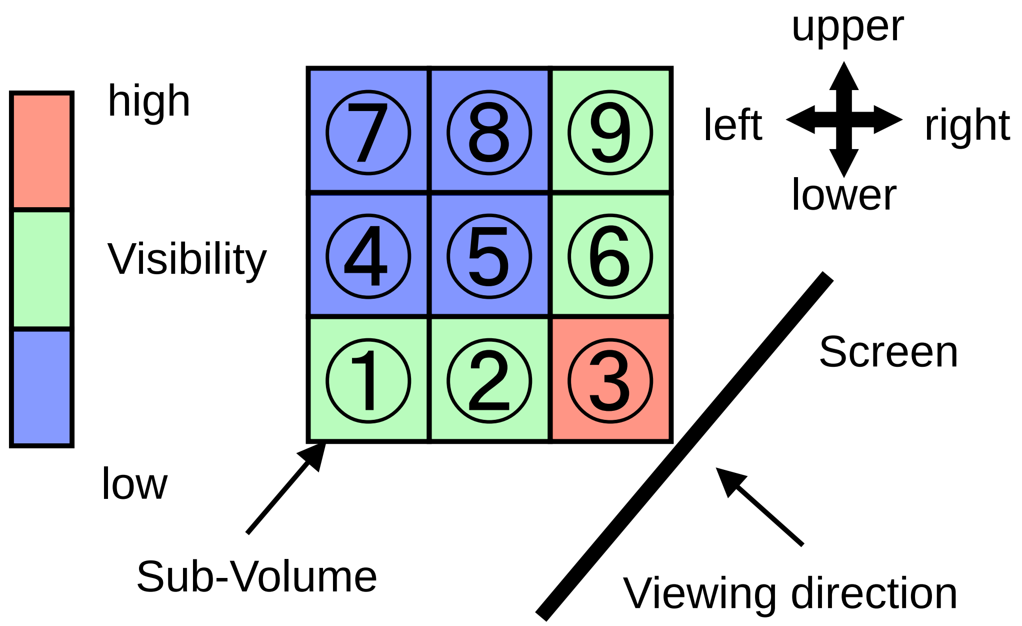


Obstacle-Flag

	①	②	③	④
upper	0	0	0	0
right	1	0	1	0
lower	0	0	1	1
left	0	0	0	0

Dynamic Job-Scheduling

Visibility of Sub-Volumes Based on Obstacle-Flags

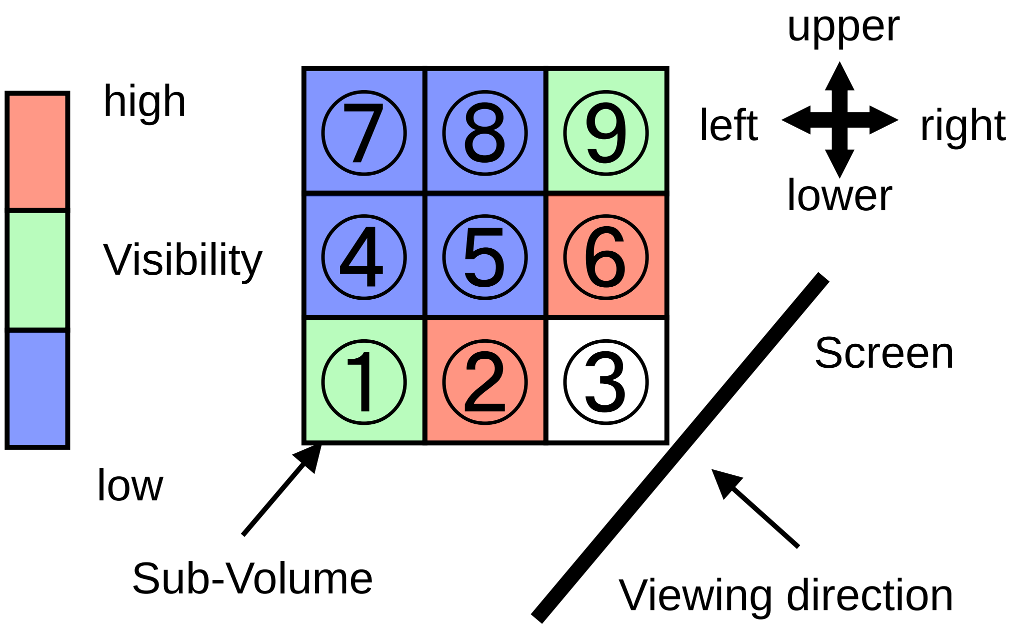


Obstacle-Flag

	②	③	⑤	⑥	⑨
upper	0	0	0	0	0
right	1	0	1	0	0
lower	0	0	1	1	1
left	0	0	0	0	0

Dynamic Job-Scheduling

Dynamic Job-Scheduling using Obstacle-Flags



Obstacle-Flag

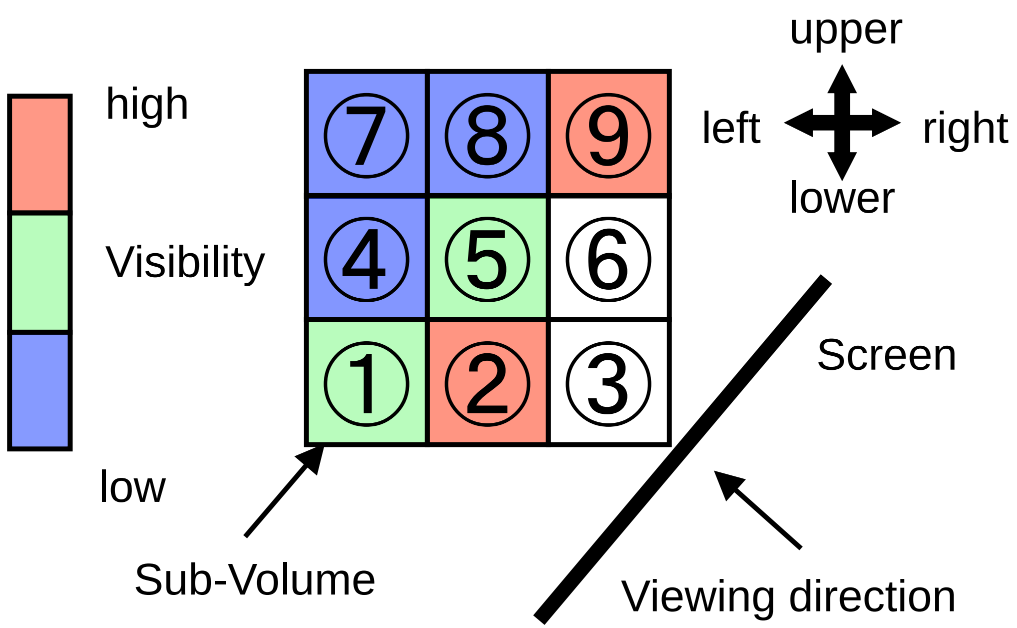
	②	③	⑤	⑥	⑨
upper	0	0	0	0	0
right	1→0	0	1	0	0
lower	0	0	1	1→0	1
left	0	0	0	0	0

Sub-Volume ③ has already been rendered and combined.

➡ Sub-volumes ② and ⑥ have their obstacle-flags updated.

Dynamic Job-Scheduling

Dynamic Job-Scheduling using Obstacle-Flags



Obstacle-Flag

	②	③	⑤	⑥	⑨
upper	0	0	0	0	0
right	0	0	1→0	0	0
lower	0	0	1	0	1→0
left	0	0	0	0	0

Sub-Volume ⑥ has already been rendered and combined.

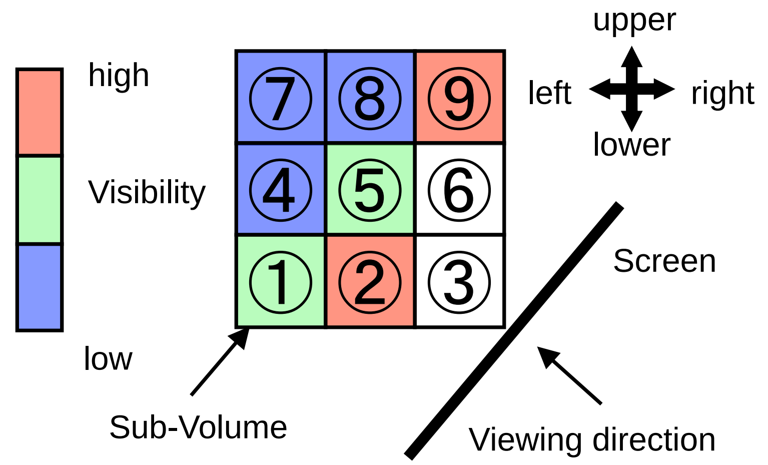
➡ Sub-volumes ⑤ and ⑨ have their obstacle-flags updated.

➡ Sub-volume ⑨ will be sendable.

Dynamic Job-Scheduling

○

Dynamic Job-Scheduling using Obstacle-Flags

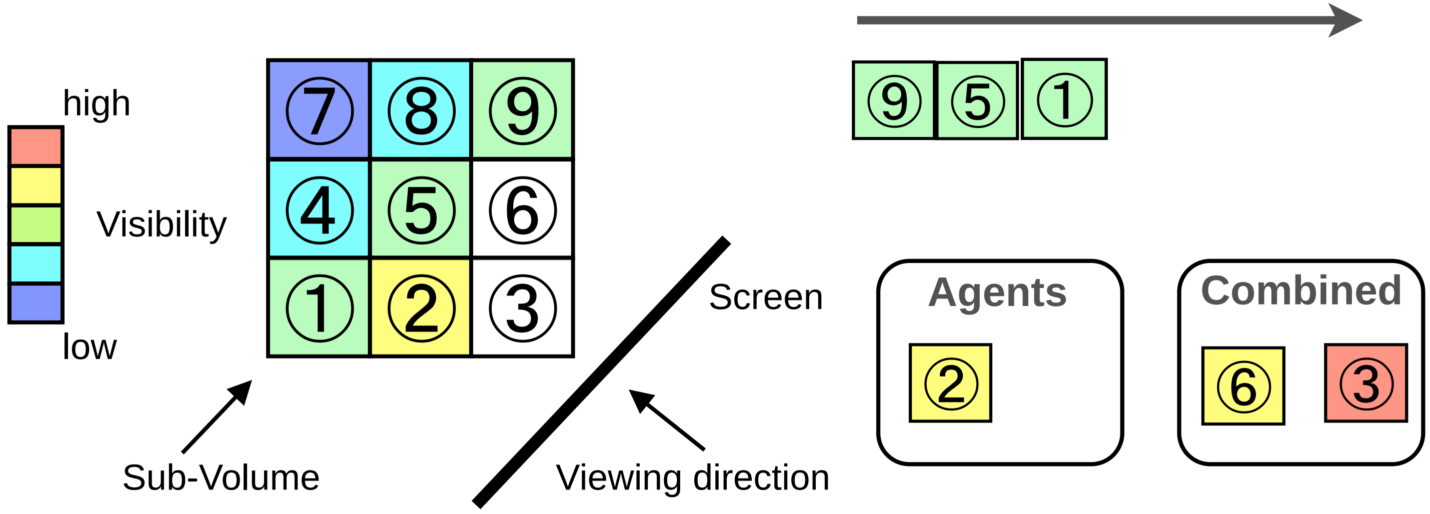


Obstacle-Flag

	②	③	⑤	⑥	⑨
upper	0	0	0	0	0
right	0	0	0	0	0
lower	0	0	1	0	0
left	0	0	0	0	0

✕

Sequential Job-Scheduling



Exception Handling

Sub-Volumes have three states:

- Not-occluded
- Partially-occluded
- Fully-occluded

If an idle agent is available...

Sending partially-occluded sub-volumes

↓

Minimize waiting time, while maximize agent utilization

Experiments

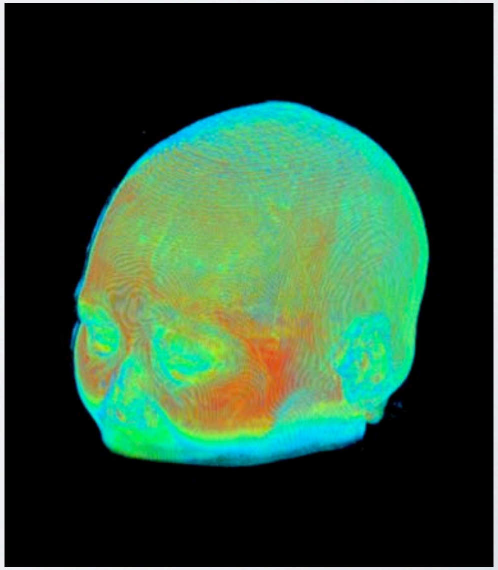
- We used our university’s campus grid.
- The computer grid’s managing software is Condor.

Test Data

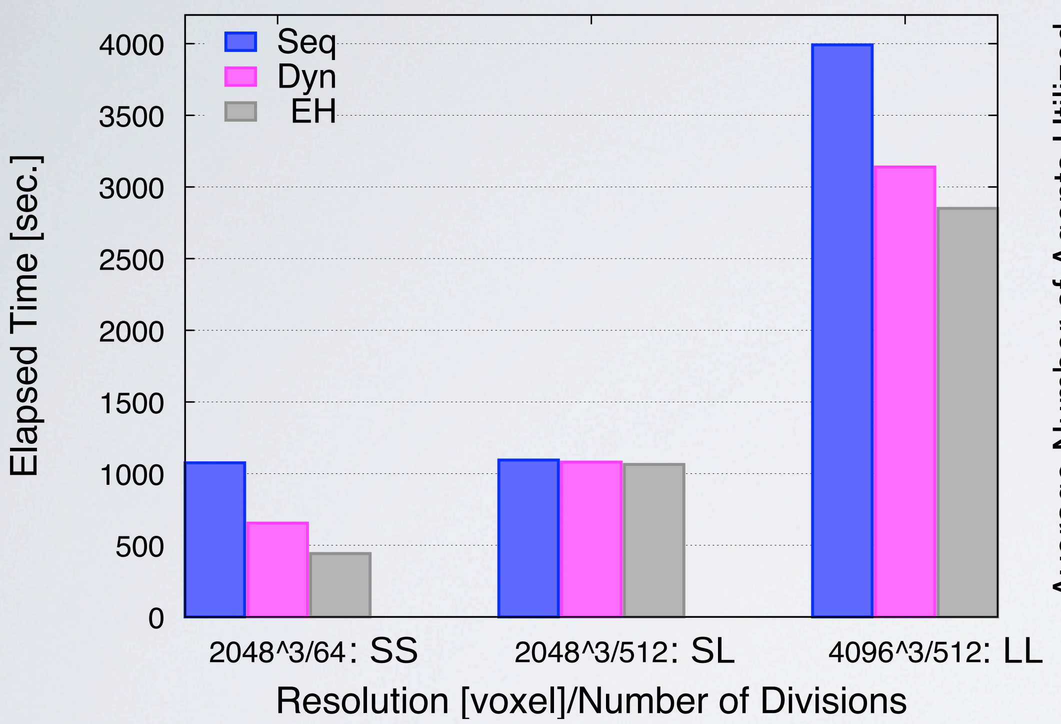
Case	Resolution [voxel]	VD size [GB]	Number of Divisions	SV size [MB]	Screen size [pixel]
SS	2048^3	16	64	256	3000x3600
SL			512	32	
LL	4096^3	128	512	256	5800x7200

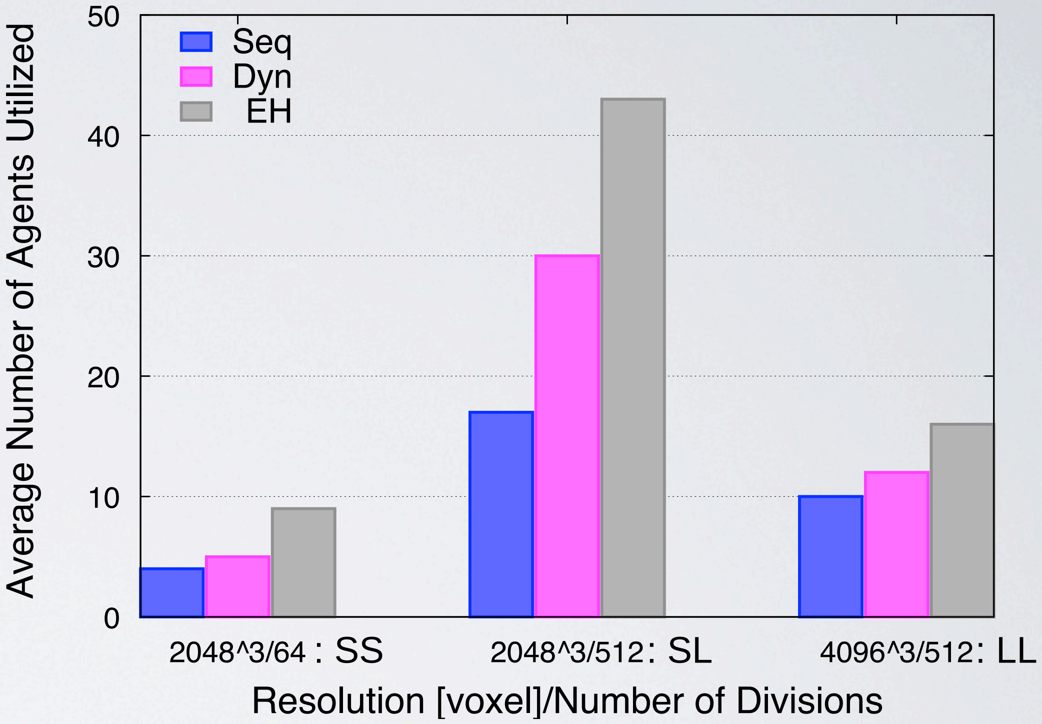
Experimentation Environment

Number of Agents	OS	CPU	Memory
34	Linux	Xeon 3.06GHz	2GB
469		Pentium4 3.06GHz	990MB



Experimental Results

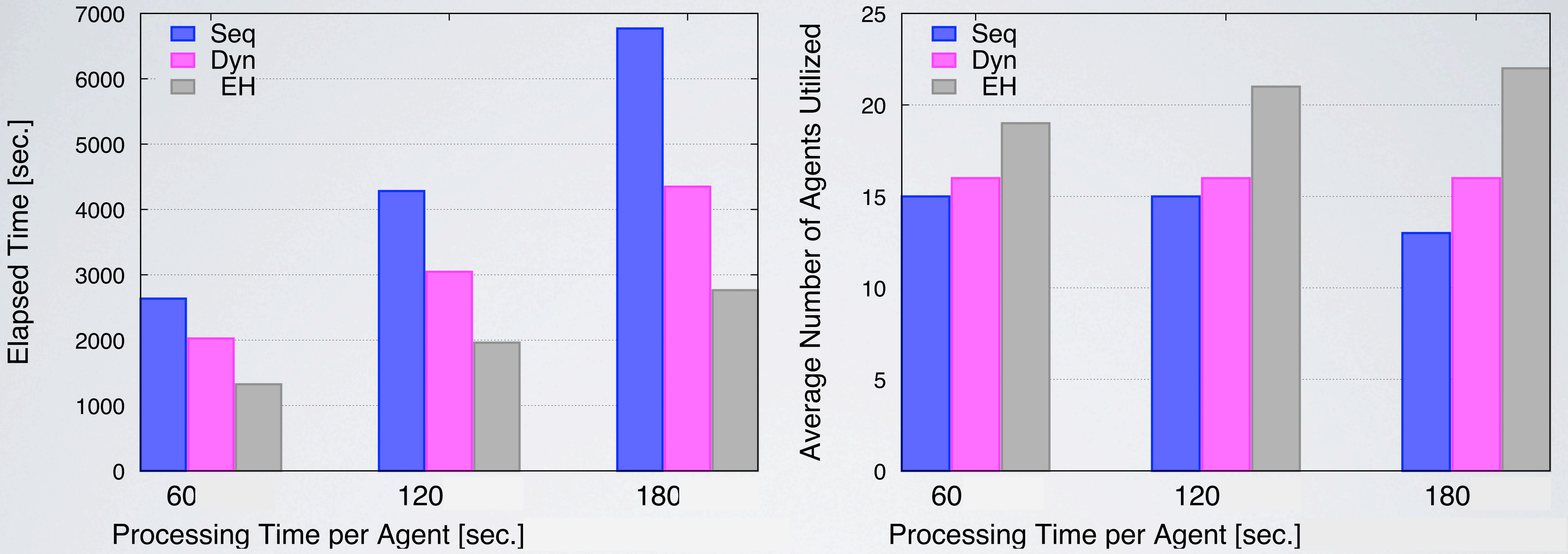




- Proposed methods used more agents.
- Elapsed time was reduced only in the SS and LL cases.

  ➡ Depending on agent processing time

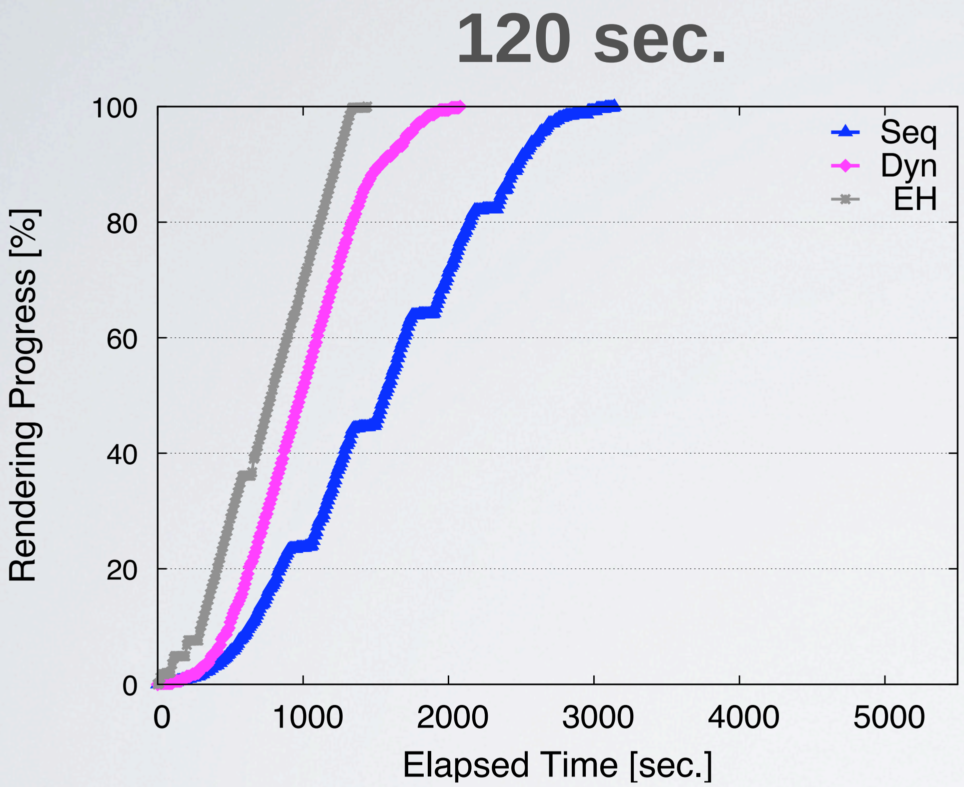
Results for Various Agent Processing Times

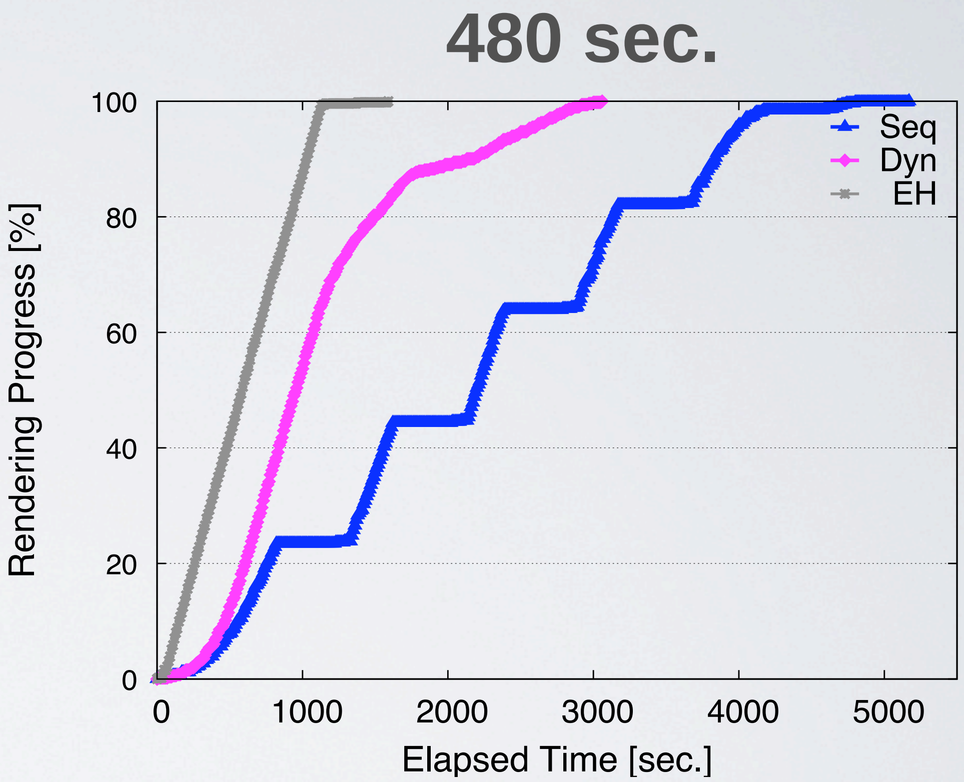


- The longer the agent processing time becomes, the better the performance

Results in the case of Interruptions

The interruption time:





Interruption Time [sec.]	Speed-Up Ratio	
	Dynamic Job-Scheduling	Exception Handling
0	1.30	1.93
120 (short)	1.65	2.20
480 (long)	1.71	3.02

Conclusions

- New method for large-scale volume data rendering in a grid computing system is proposed.
  - Dynamic Job-Scheduling using obstacle-flags
  - Performs better than the sequential job-scheduling as verified experimentally

Future work

- Experiments using larger volume data
  - Terabyte volume data
  - Increasing the number of divisions