Multiapplication Intertile Synchronization on Ultra-High-Resolution Display Walls

Sungwon Nam
Electronic Visualization Laboratory
University of Illinois at Chicago

The Importance of Visualization

- Vision is the dominant sense for the acquisition of information from our everyday world.
- Nearly 1/3 of your brain is devoted to processing visual information.
- Visualization has 3 main roles in scientific / engineering computing:
  - As an instrument to view and understand complex phenomena (like a microscope)
  - To validate results (like a computational simulation)
  - To explain complex results to a lay audience (such as government / policy makers and the general public, and to inspire the next generation of scientists)

The 105 Mpixel Lens

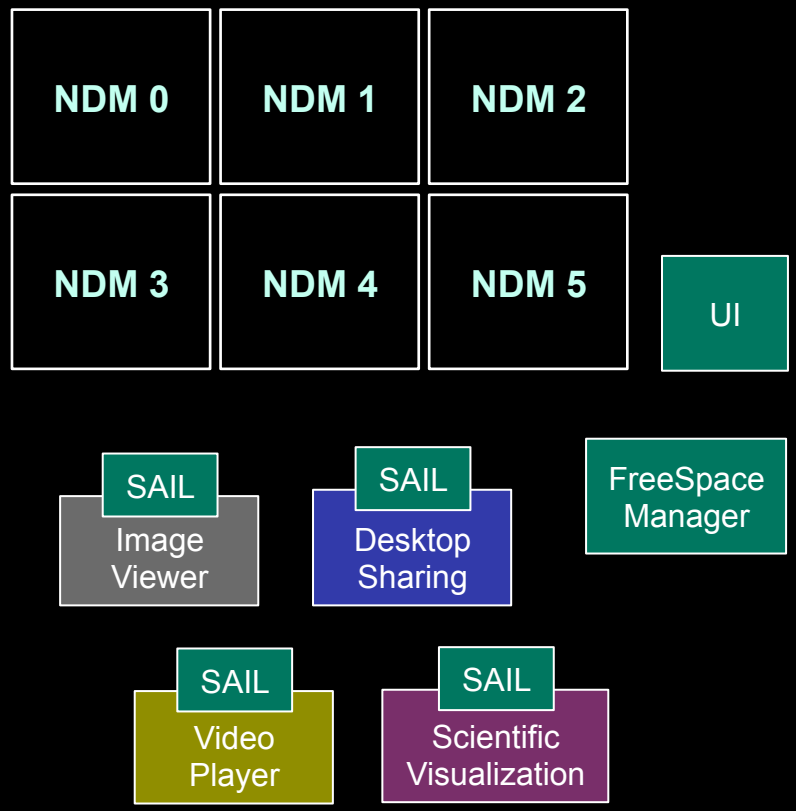
- 28 PCs with GPU
- 11x5 LCDs
- 105 Mpixel

The 20 Mpixel Lens

- 1 PC
- 3 GPUs
- 18 Screens

Scalable Adaptive Graphics Environment (SAGE)

- It consists of
  - FreeSpace Manager (FSM)
  - Node Display Manager (NDM)
  - Application Interface (SAIL)
  - Sync. Manager
  - UI
- Each NDM manages single logical screen(a tile) and is driven by a computer in a cluster.
- SAIL streams pixel to the display wall



Communication methods for a cluster-based display wall

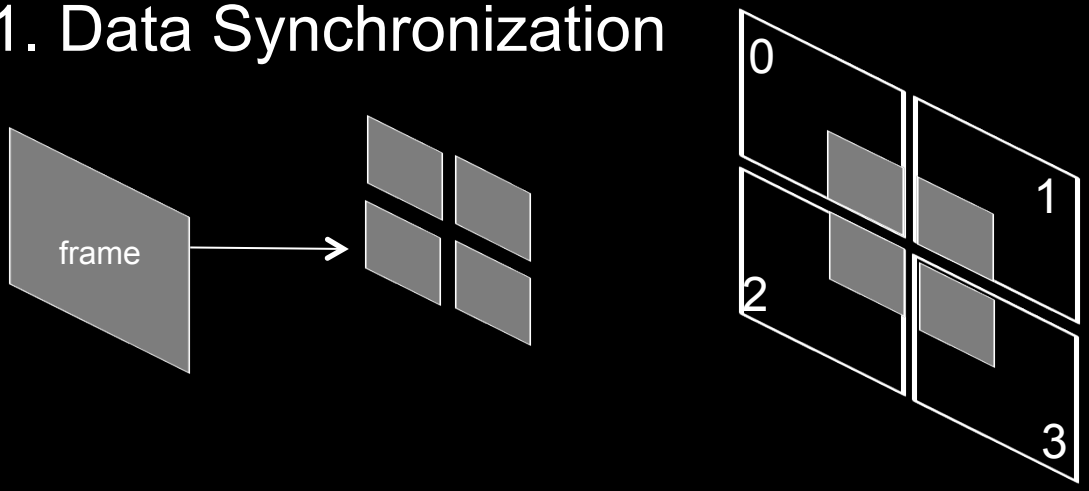
- Synchronized Execution
  - all render nodes have the same copy of the application instance
- Primitive Distribution
  - a client distributes grphics primitives to render servers
- Pixel Distribution
  - a client renders and transmits only pixels to display servers

What is the problem?

- You are watching motion pictures on the tiled display run by a cluster of computers.
- You want to synchronize frame transition on each tile to be seamless.
- It's a human factor.
- It is trivial if there is only one animation on the tiled display.
- What if you want to display multiple animations each has its own frame rate at the same time.

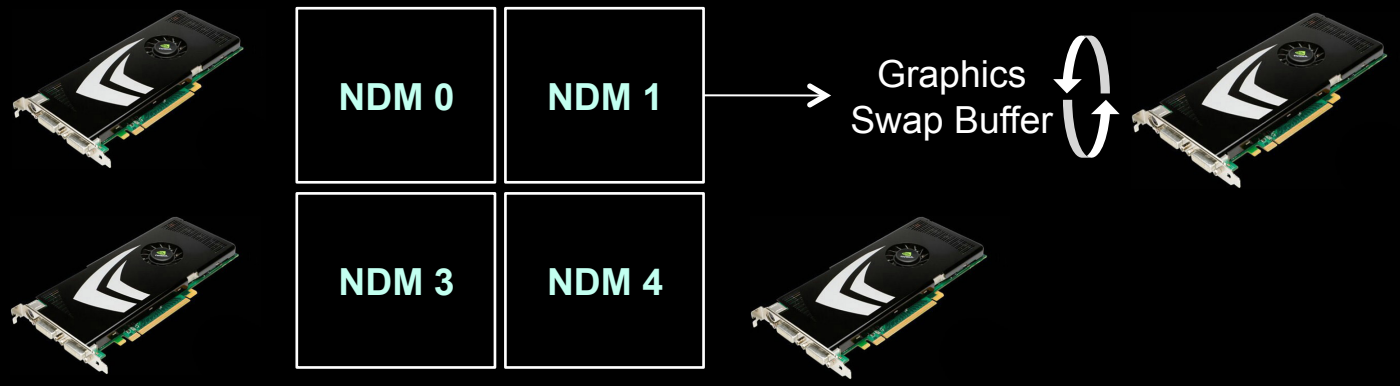
Intertile Synchronization Requirements

1. Data Synchronization

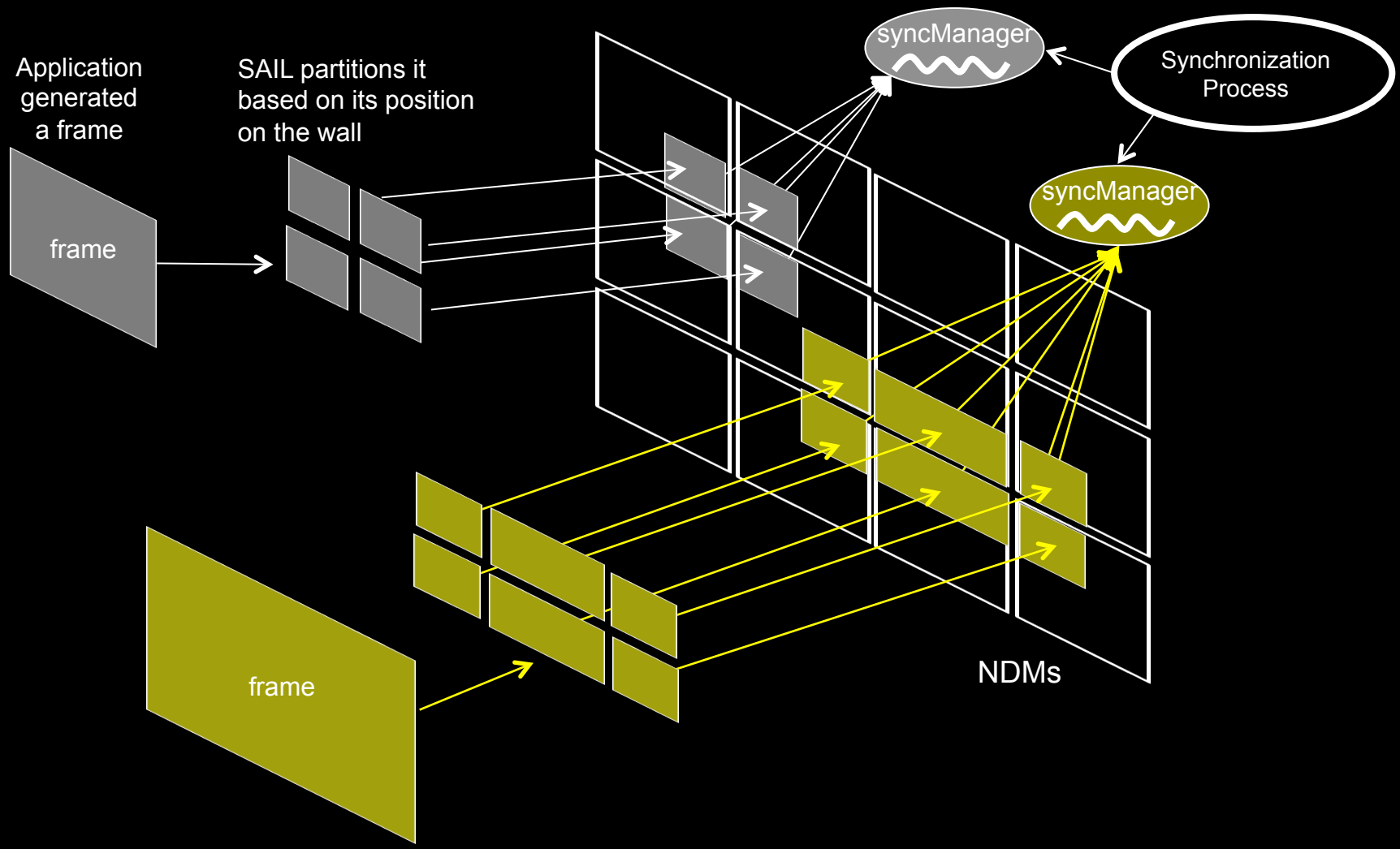


All the image fragmensts must be in the same frame number

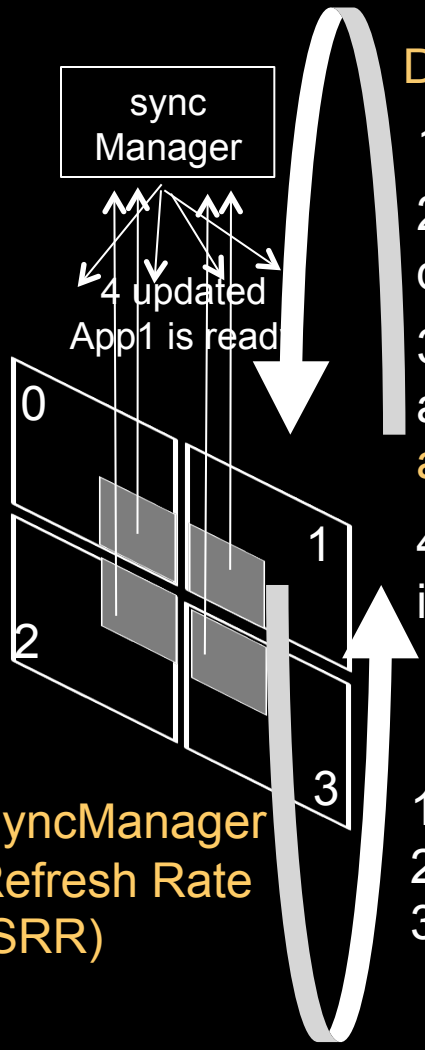
2. Graphics Swap Buffer Synchronization



SAGE Intertile Synchronization (old)



A Global Synchronization Manager (Two Phase Algorithm)



Data Synchronization

1. Wait for a msg from NDMs for certain period.
2. Determine which application screen is ready to be displayed.
3. Create a message based on step 2, and broadcast the message. A single message for all application on the wall.
4. Corresponding NDMs are ready to display their fragment, if the message contains a flag for the application

1st phase

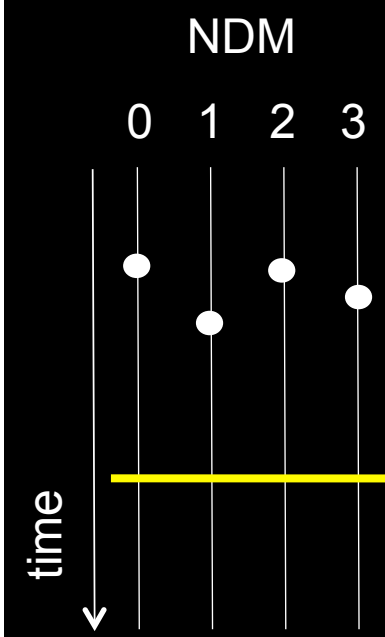
SyncManager Refresh Rate (SRR)

Graphics Swap Buffer Synchronization

1. wait for a msg from all NDMs
2. Broadcast a msg
3. NDMs execute graphics swap buffer

2nd phase

A Global Synchronization Manager (One Phase Algorithm)



Data Synchronization

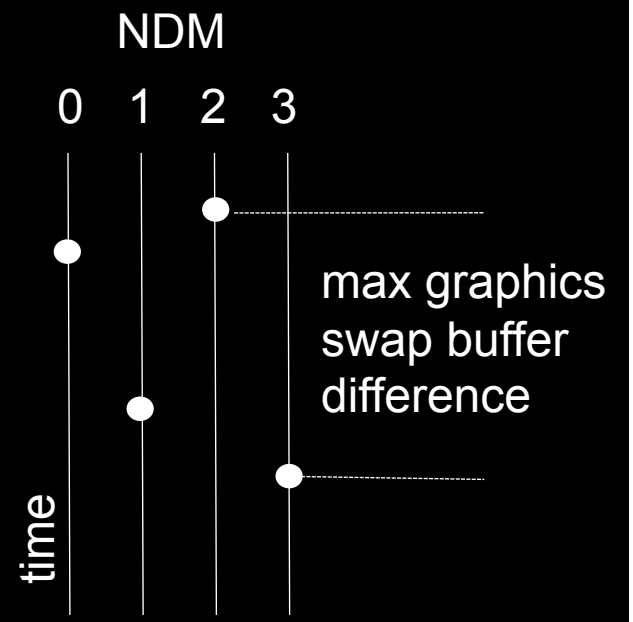
1. Wait for sync. msgs from NDMs for certain period.
2. Determine which application screen is ready to be displayed.
3. Create a message based on step 2, and broadcast the message.
   A single message for all application on the wall.
   A Presentation Time is included in the message.
4. Corresponding NDMs are ready to display their fragment if the message contains a flag for the application

1st phase

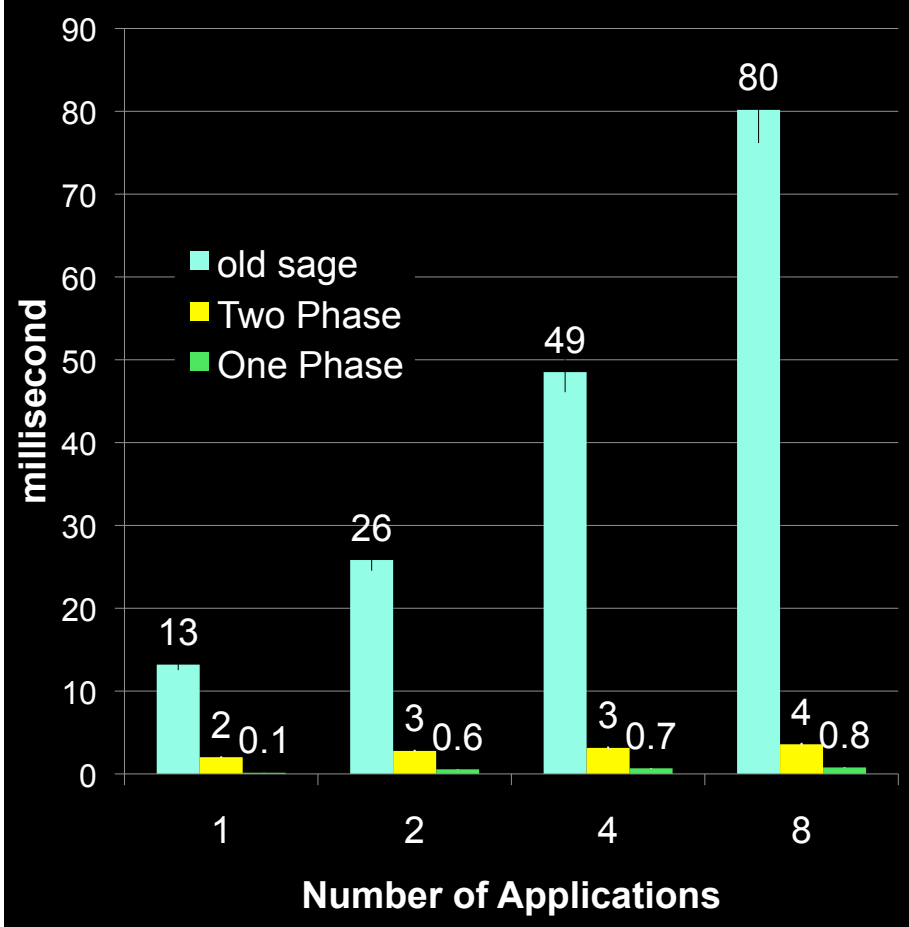
Graphics Swap Buffer Synchronization using Network Time Protocol (NTP)

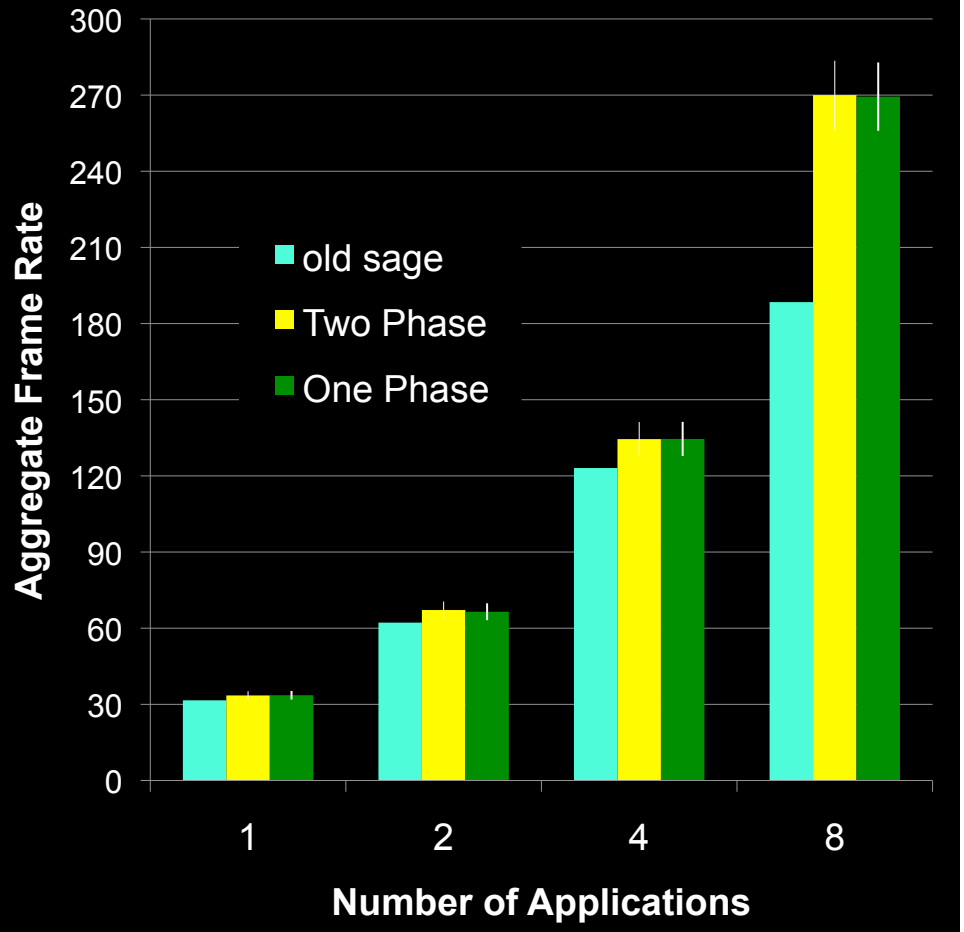
1. Each NDM waits until its own clock reaches the Presentation Time
2. A NDM executes graphics swap buffer

Sync Mismatch (Single Application)



Sync Mismatch and Frame Rate (Multiple Application)





Conclusion

- Presented two algorithms to acheive intertile synchronization.
  - Two Phase Algorithm
  - One Phase Algorithm
- Both focus on reducing network messages.
- We can enforce graphics swap buffer sync. with global sync. manager in multiapplication enbled tiled-display environment.

Acknowledgements

- The publication was based on work supported in part by Sharp Laboratories of America, the King Abdullah University of Science and Technology (KAUST) (Award US-2008-107/SAC0064), the National Science Foundation (Award OCI 0943559), and the Office of Advanced Scientific Computing Research, Office of Science U.S. Department of Energy, under Contract No. DE-AC02-06CH11357.
- We would like to thank Lance Long, Alan Verlo, Patrick Hallihan, Andrew Johnson, and Maxine Brown at the Electronic Visualization Laboratory, and Gail Pieper at the Argonne National Laboratory for reviewing the paper.

Thank you