Distributed Systems *in* "Real Life Applications"

CONTENTS

APPLICATIONS OF DISTRIBUTED SYSTEMS

- Telecommunication networks:
  - Telephone networks and cellular networks
  - Computer networks such as the Internet
  - Wireless sensor networks
  - Routing algorithms
- Network applications:
  - World wide web and peer-to-peer networks
  - Massively multiplayer online games and virtual reality communities
  - Distributed databases and distributed database management systems
  - Network file systems
- Real-time process control:
  - Aircraft control systems
  - Industrial control systems
- Parallel computation:
  - Scientific computing, including cluster computing and grid computing
  - Distributed rendering in computer graphics

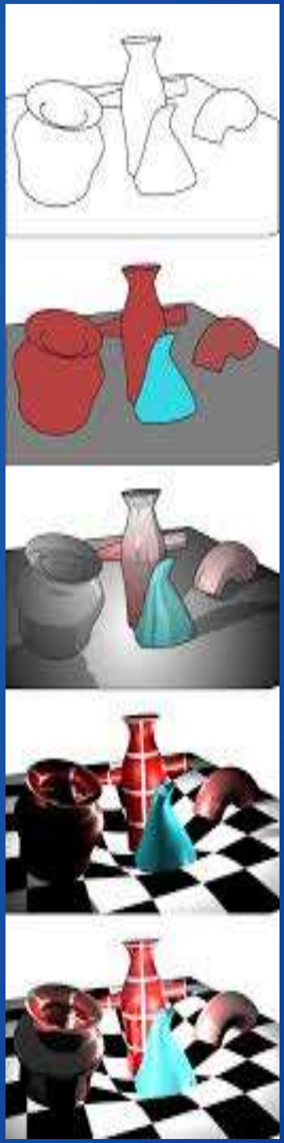
Part I

DISTRIBUTED RENDERING IN COMPUTER GRAPHICS

DISTRIBUTED RENDERING

- Basic concepts of parallel rendering in computer graphics
- The importance of parallel rendering
- Different types of rendering parallelism
- Algorithms in parallel rendering

INTRODUCTION



- What is Rendering and Parallel / distributed rendering?
- For complex objects or high quality images, the rendering process requires massive computational resources. For example, the rendering of medical visualization and some CAD applications may need millions or billions of floating-point and integer operations for each image.
- To obtain the required computing power, the only practical solution is to exploit multiple processing units to speed up the rendering process. This technology is known as parallel rendering

RENDERING FUNCTIONS

- The rendering process can be divided into a number of distinct functions:
- 1) Geometry processing: modeling transformation.
- 2) Rasterization: scan-conversion, shading,

TYPES OF PARALLELISM

1. Functional
2. Data
3. Temporal

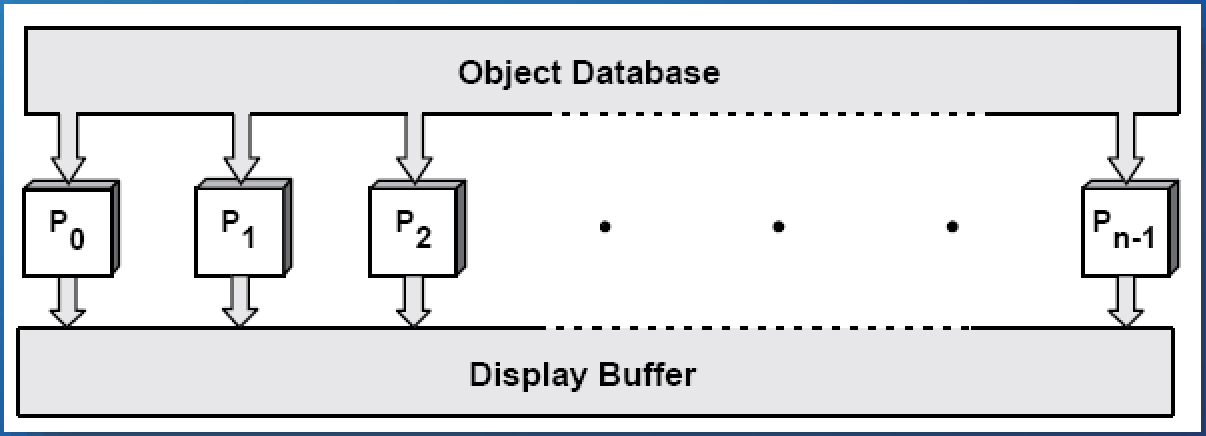
FUNCTIONAL PARALLELISM

- Functions are applied in series to individual data items.
- Functions are assigned to different processing units; once a processing unit completes its work on a data item, it sends the data item to the next unit and receives a new item from its upstream neighbor.
- The functional parallelism is suitable for polygon and surface rendering applications. We feed the 3D geometric objects into the beginning of the pipeline, and retrieve the final pixel values from the end of pipeline.

DATA PARALLELISM

- Data parallelism focuses on distributing the data across different processing units

- The graphics data are divided into a number of streams; multiple data items can then be processed simultaneously on different processing units, and the results are merged to create the final image



TEMPORAL PARALLELISM

- Temporal parallelism usually adopted by animation applications. In this kind of applications, a large number of images are generated for subsequent playback; parallelism is obtained by decomposing the overall rendering task in the time domain; and processors are assigned a number of frames to render, along with the data needed to produce those frames.

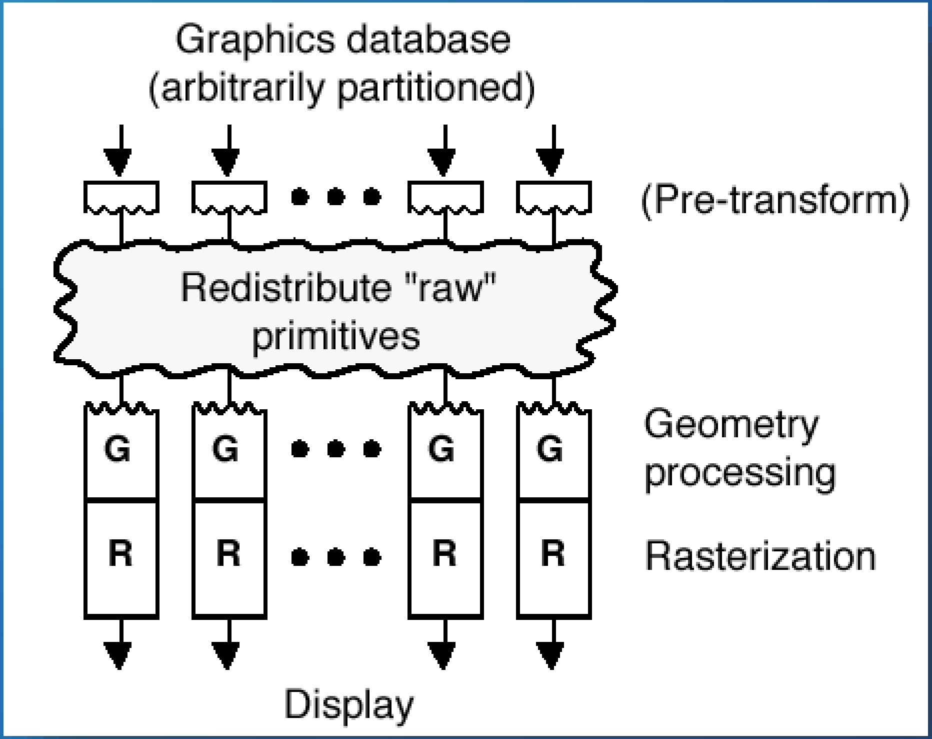
PARALLEL RENDERING – A SORTING PROBLEM

- As stated In the stage of rasterization, each processor is assigned a portion of the pixel calculations. The rendering can be considered as determining how each object affects each pixel. Since the modeling and viewing transformations are arbitrary, an object can fall anywhere on the screen. Therefore we can view rendering as a sorting problem, i.e. sorting objects to the screen

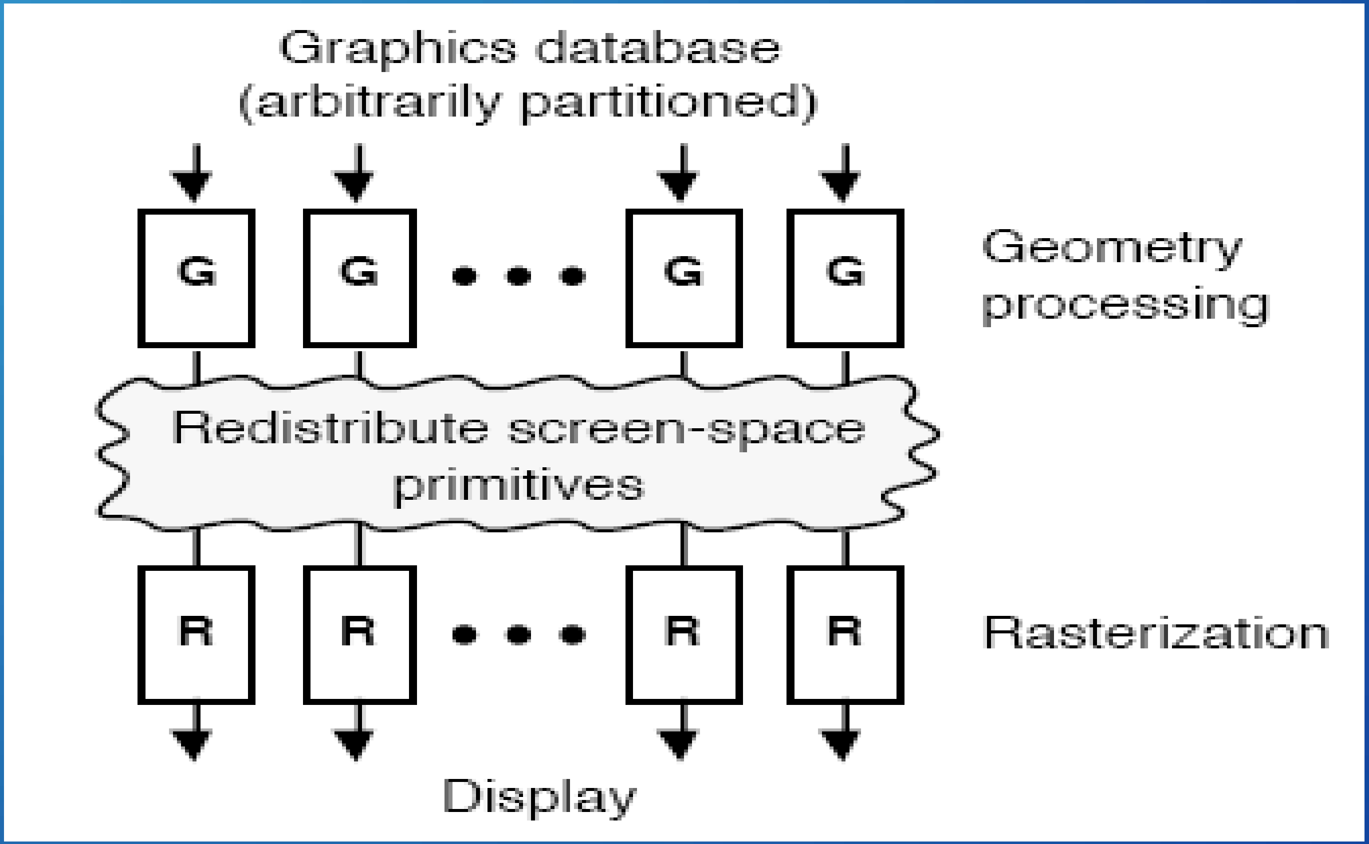
THREE GROUPS OF ALGORITHM

1. Sort-First
2. Sort-Middle
3. Sort-Last

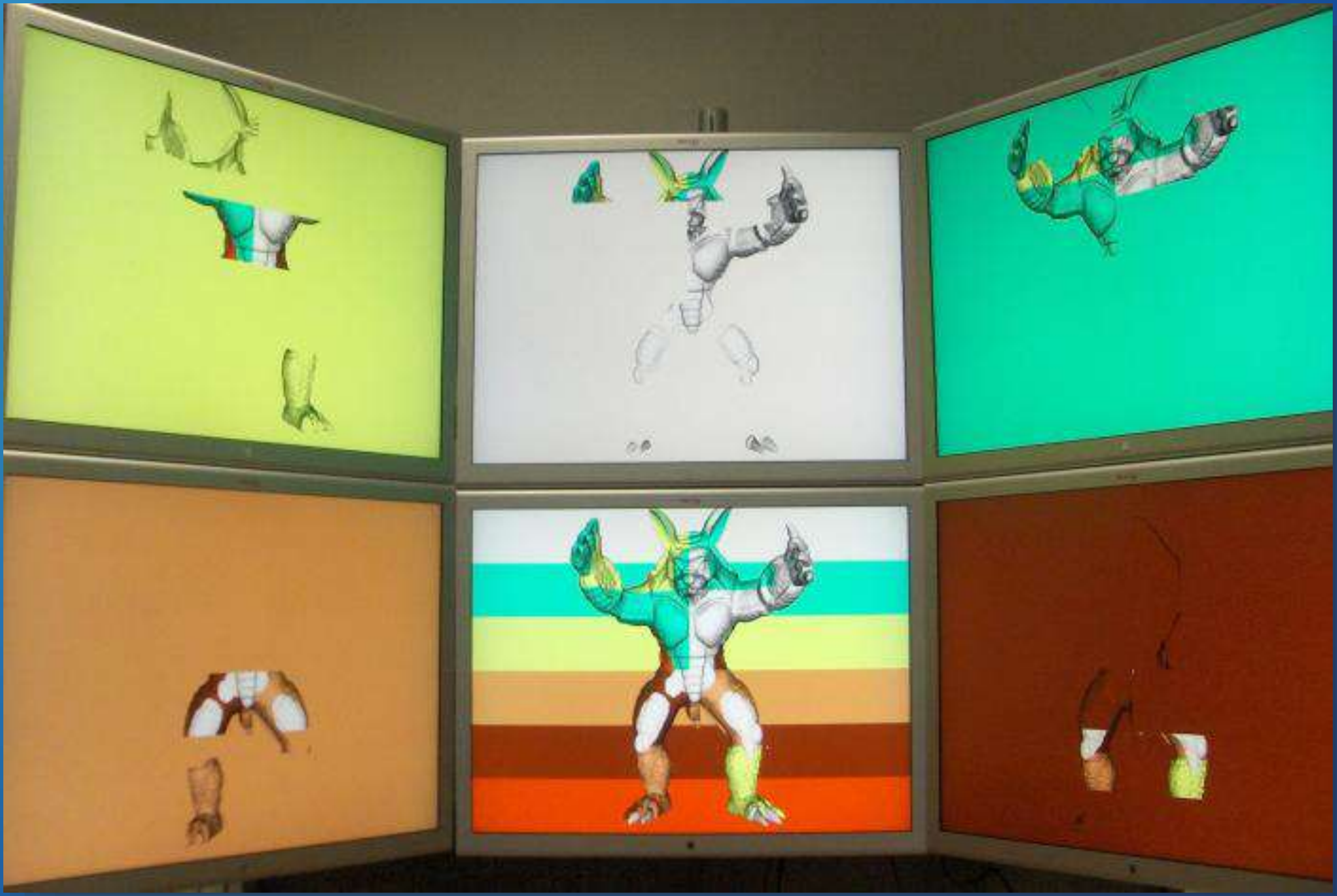
SORT-FIRST


Graphics database
(arbitrarily partitioned)
(Pre-transform)
Redistribute "raw" primitives
G
Geometry processing
R
Rasterization
Display

SORT-MIDDLE


Graphics database
(arbitrarily partitioned)
G
G
• • •
G
G
Geometry processing
Redistribute screen-space primitives
R
R
• • •
R
R
Rasterization
Display

SORT-LAST

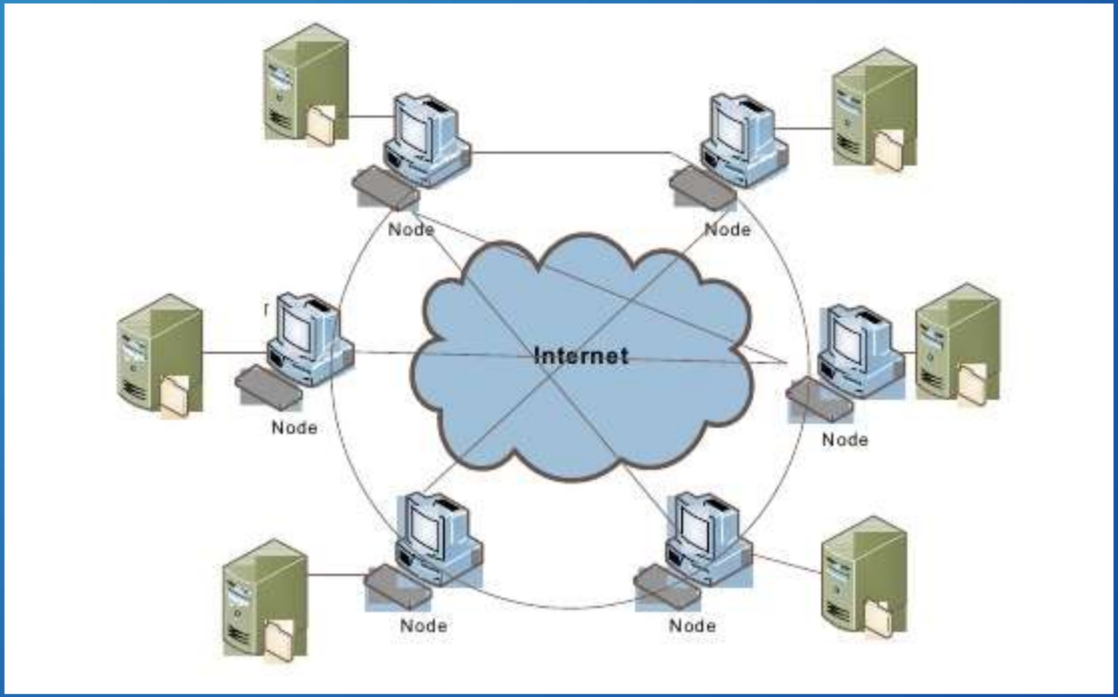


CONCLUSION

- Although parallel rendering techniques have been successfully employed in many applications; challenges still remain to improve the scalability, load balancing, communication, and image composition of these algorithms and produce higher performance.

Part II

PEER-TO-PEER NETWORKS



INTRODUCTION

- A peer-to-peer (P2P) network is a type of decentralized and distributed network architecture in which individual nodes in the network (called "peers") act as both suppliers and consumers of resources, in contrast to the centralized client–server model where client nodes request access to resources provided by central servers.

CURRENT APPLICATIONS OF P2P

- File-sharing networks
- Communications
- Content delivery
- Streaming media

PEER-TO-PEER FILE SHARING

- Peer-to-peer file sharing is the distribution and sharing of digital documents and computer files using the technology of peer-to-peer (P2P) networking.

- P2P file sharing allows users to access media files such as books, music, movies, and games using a specialized P2P software program that searches for other connected computers on a P2P network and locates the desired content. The nodes (peers) of such networks are end-user computer systems that are interconnected via the Internet.

THE BITTORRENT PROTOCOL

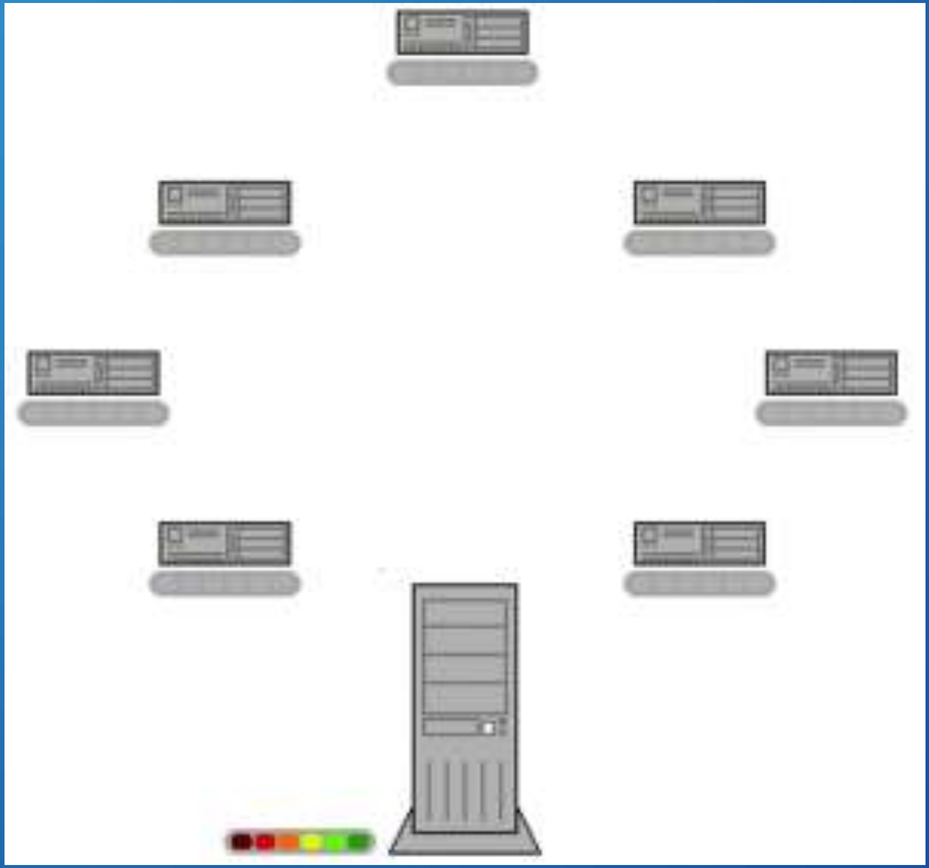
- BitTorrent is a protocol that enables fast downloading of large files using minimum Internet Bandwidth. It costs nothing to use and includes no spyware or pop-up advertising.

- Unlike other download methods, BitTorrent maximizes transfer speed by gathering pieces of the file you want and downloading these pieces simultaneously from people who already have them. This process makes popular and very large files, such as videos and television programs, download much faster than is possible with other protocols.

TRADITIONAL CLIENT-SERVER DOWNLOADING

- You open a Web page and click a link to download a file to your computer.
- The Web browser software on your computer (the client) tells the server (a central computer that holds the Web page and the file you want to download) to transfer a copy of the file to your computer.
- The transfer is handled by a protocol (a set of rules), such as FTP (File Transfer Protocol) or HTTP (HyperText Transfer Protocol).

ADVANCED P2P DOWNLOADING



ADVANTAGES

- Fault tolerance
- Performance
- Cost efficiency

FUTURE SCOPE

- Academic Study Material in Schools and Colleges
- E-papers
- Free-to-Air Radio channel services
- Large software updates and other general data files
- Global availability of private data

Part III

MASSIVELY MULTIPLAYER ONLINE GAMING

INTRODUCTION

- Previously, Client/server communication architecture was used for gaming.

- This architecture has the advantage that a single authority orders events, resolves conflicts in the simulation, acts as a central repository for data, and is easy to secure.

- First, it introduces delay because messages between players are always forwarded through the server. Second, traffic at the server increases with the number of players, creating localized congestion1. Third, the size of the virtual world and the quantity the data it contains is limited by the storage capacity of the server.

A distributed architecture for MMOGs has the potential to overcome these problems.

- Players can send messages directly to each other, thereby reducing delay and eliminating localized congestion. The storage capacity created by combining the storage resources of all players can easily exceed that of a single server. Furthermore, the ability to execute thousands of remote processes in parallel on user machines provides unparalleled computational power for MMOGs

Distributed architecture has several fundamental and challenging problems that must be overcome:

- Authenticating players and granting access rights to the game.
- Maintaining consistency, ordering events, and propagating events to intended recipients
- Providing tamper-resistant storage of characters and game state
- Scheduling computations across the players
- Providing responsiveness to interactive elements of the game
- Ensuring the architecture is cheat-proof by preventing players from taking advantage of the architecture itself in order to cheat.

PROTOCOLS

1st protocol – MiMaze Protocol

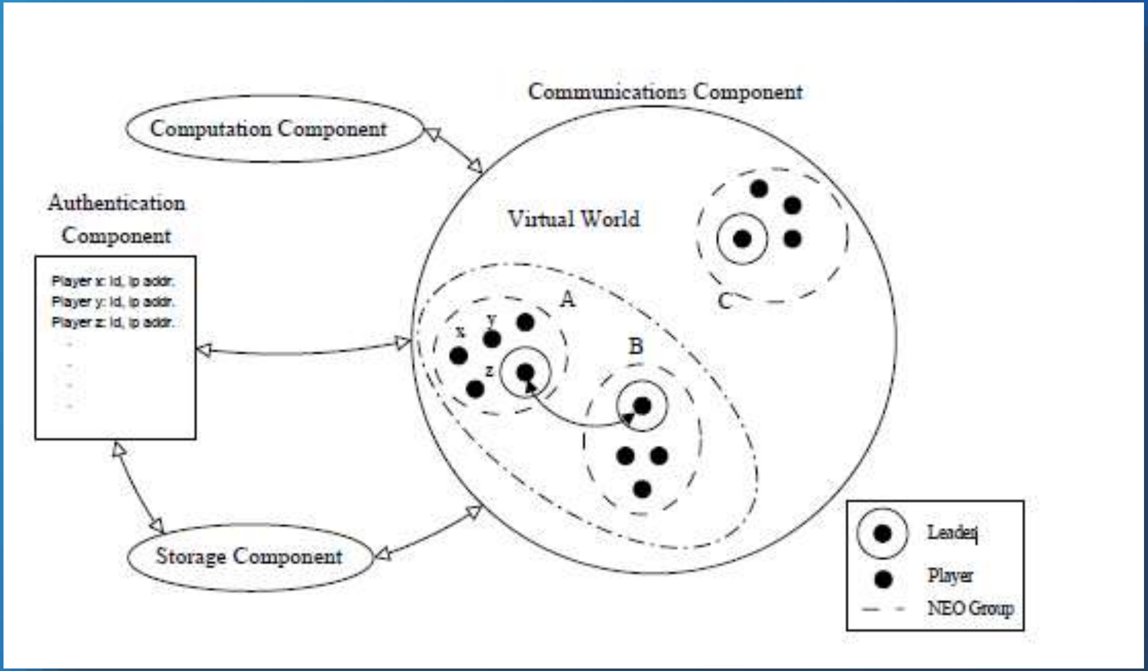
Uses multicast , problem of cheat

2nd protocol - Lockstep protocol

Lockstep protocol

- The **lockstep protocol** is a *partial* solution to the look-ahead cheating problem in peer-to-peer architecture multiplayer games, in which a cheating client delays his own actions to await the messages of other players.
- To avoid this method of cheating, the lockstep protocol requires each player to first announce a "commitment"
- Drawback:
- As all players must wait for all commitments to arrive before sending their actions, the game progresses as slowly as the player with the highest latency.
- Overcome: asynchronous lockstep protocol
- Lockstep is a major advance in communication protocols for distributed games because it is provably secure against several cheats. Unfortunately, time progresses in the lockstep protocol at the speed of the slowest player.
- Unfortunately , cannot meet the bounded time requirements necessary for games.

KNUTSSON



- The virtual world is divided into regions, and players in each region form a group. Each region maps to a multicast group through Scribe so that updates from the players are multicast to the group. Consistency is achieved through the use of coordinators. Every object in the game is assigned to a coordinator; therefore, any updates to an object must be sent to the coordinator who resolves any consistency problems. Fault tolerance is achieved through replication.

PEER TO PEER GAME ARCHITECTURE

We divide the architecture into four main components:
Authentication, Communications, Storage, and Computation.

First, the authentication component is responsible for controlling access to the game. The communication component determines how players send messages to each other and the storage component provides long-term storage of world state.

Last, the computation component schedules computations
across the player base.

AUTHENTICATION

The challenging problems with distributed authentication are security and scalability

We use hash table : provides unique ID WITH NAME AND expiration date

To help with scalability, once the pairs are generated, they are stored on a DHT by the player or server.

The advantage of using a DHT is that as the number of players increase, the storage capacity increases.

COMMUNICATION COMPONENT

A distributed communication architecture has the possibility of supporting millions of players.

client/server architectures, all communications must be sent to the server before being sent to players, introducing additional delay. Distributed communication allows players to send messages directly to each other, creating a more responsive game.

Consistency is required if players are from different regions if error arises they wont be able to play together.

Dividing communication between zones. As the players inc. we divide the zones into virtual world to peer to peer groups.
 To organize the peer-to-peer networks into a hierarchy of groups, we use distributed election protocols to elect one or more leaders to act as nodes in the hierarchy. The nodes assist in event propagation and group location. To help build the hierarchy of groups, we propose using a DHT to map unique zone IDs to the list of nodes in a given zone.
TO provide consistency and event ordering using a new protocol called NEO . NEO ensures low latency and also prevents cheating

STORAGE

- TWO TYPES:

1. *Ephemeral Data*

- Data which is ephemeral is simply stored on the DHT, indexed by its geographical area in the virtual world. Players that enter an area, query the DHT for all ephemeral data.

1. .

2. *Permanent Data*

- This allows the player to make sure the data is always available when they are online, but provides a backup in case their local data becomes corrupted.

COMPUTATION COMPONENT

- The server in a client/server architecture typically does not have enough power to simulate complex interactions between players and the artificial intelligence (AI) behind monsters and other non-player characters.

FUTURE SCOPE

- Currently available for expert users only. Can be extended to novice users.
- Scope for cost reduction in terms of hardware and resources.
- Maximum no. of players is currently limited.

Thanks.