

Network Gaming

(Including MMORPG)

1

Introduction

- MUD
 - Multi-user Dungeons
- Match makings
 - Real-time strategy
 - FPS
- MMORPG
 - MMOG
 - Massive Multiplayer Online Role-playing Games

2

MUD

- A large scale game with fantasy-type
- Create your own character
 - Strength/Intelligence/Endurance/Wisdom
- Most of the MUDs are text-based
 - MUD commands
- Players play them simultaneously
- Persistent worlds
 - The amount of time invested
 - Never ending
- Competition
 - Skill & Attributes
 - Experience points
- Ownership
 - Acquire better items
- Society

3

Match Makings

- Arcade
 - 休閒遊戲
- RTS
 - Real-time strategy games
 - Internet
- FPS
 - First-person-view shooting games
- AI
- Game lobby (遊戲大廳)
 - Match making
 - Ranking
 - Chatting
 - A special case
 - Xbox Live

4

Xbox Live

- Xbox games with Live features
 - Xbox is capable of networking in default.
 - An official lobby server created and maintained by Microsoft
 - Unique gamer tag in the whole world for each registered player
 - Every player in any Xbox Live game will know whether his friends are on or off in any Xbox Live game.
 - Then he/she can send the invitation to his friends for them to join the same game.
 - All Xbox Live development project must be followed and certificated by Microsoft Xbox Live guide.
 - World-wide ranking on the Xbox Live lobbies
 - Game points controlled by Microsoft
 - Voice chatting
- Xbox Live Arcade

5

MMORPGs

- Graphical MUD
 - With much more functionality
- Persistent
- Real-time online (即時在線)
- Pay-for-play
- New game gtyles :
 - RPG
 - Adventure
 - Action
 - Society
 - Messaging
 - Avatar

6

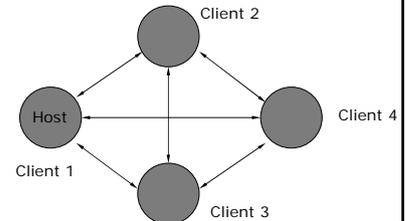
Network Gaming Architecture

- Peer-to-peer
- Client server

7

Peer-to-peer

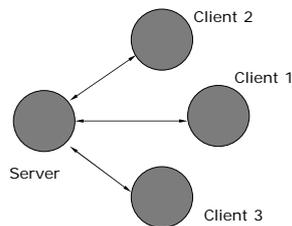
- The client broadcasts messages to each other clients.
- Simple pattern
- Easy to code
- But need more bandwidth for each client
- Limited players
- Fault tolerance
- High security risk



8

Client Server

- One server serves every clients
- More complex in coding
- Many players
- The server needs more bandwidth
 - This means "Expensive" for hosting the servers.
- Improved security
- No fault tolerance
- Dedicated server solution



9

Internet Consideration

- Limit network bandwidth
- Unstable network environment
- Update frame rate
 - For match making : 5-8 fps
 - Position updates
 - Inventory updates
 - Kill or killed
 - ...
 - For MMOG : 1-2 fps
 - Position updates
 - Inventory updates
 - Kill or killed
 - Friend list updates
 - Chatting
 - Messaging
 - ...

10

Poor Networking Problems

- Limited bandwidth
 - Data packet size is important
 - Solutions
 - Data compression
 - Lower update frame rate
- Low update frame rate
 - Synchronization is the key
 - Solutions
 - Extrapolation
 - Prediction
 - Chasing
- Network LOD
- Network visibility

11

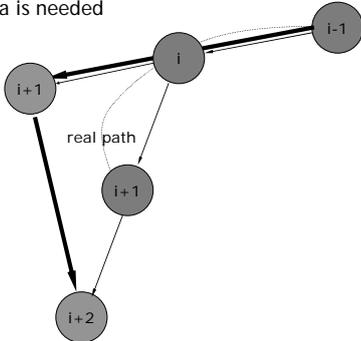
Data Packet

- Data compression
- Encode / Decode
- CPU time vs data size
- Socket
- TCP/IP & UDP

12

Position Extrapolation

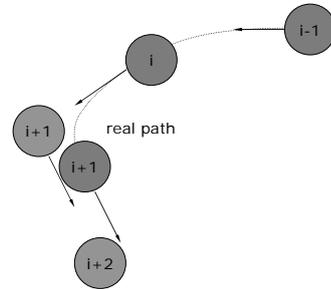
- Use position(i-1) & position(i) to get the next possible position (i+1)
- No velocity data is needed



13

Position Prediction

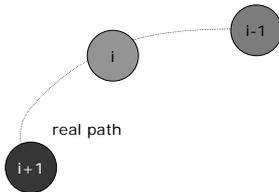
- Use current facing direction to predict next position
- Update : (position, facing direction)



14

Position Chasing

- Always being late one step to the master client
- Not good for fast moving objects



15

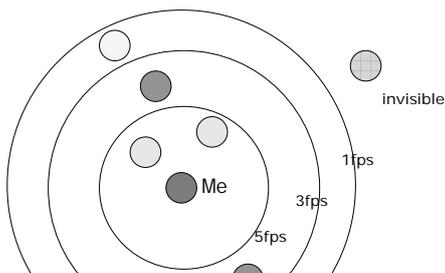
Network Visibility

- Server will not update the position of the clients that are out of a specific range of one client.
- Only when the outside clients move into the range of the client.
- That means the client can not "see" the clients outside its range on the net.

16

Network Level-of-details

- Besides the network visibility, apply the network LOD to the client between clients.
- For each client, the server should adaptively adjust the update frame rate for the other clients to the target client according to their physical distance.



17

TCP/IP or UDP

- Some game data can be lost.
 - Position for the object in each frame
 - UDP is fast but not reliable
- Some must be not
 - Kill or not kill, killed or not killed
 - TCP/IP
- Hybrid data transmission ways :
 - Data-can-not-lost going through TCP
 - Data-can-lost going through UDP
 - Or implement you reliable UDP solution

18

Network Security

- Anti-hacking
 - System administration issues
 - Very important in MMORPG
- Cheat prevention
 - Technical example:
 - A "fake" client instead of the game client to send cheating data packets.
 - Game-playing example:
 - Using the game bugs to get the improper fortune
 - Be-killed -> Dead -> Get free money -> Be-killed ->...

19

MMORPGs Features

- Avatar
- Levels
- RPG game play
- Mission
- Chatting
- Society & community
- Friends
- Combat
- NPCs / monsters
- Experience points
- Extended game contents
- Online customer services (GM)

20

MMORPGs Technical Elements

- Client-server architecture
- Servers
- Network bandwidth
- Network data packet
- Network security
- Graphics
- Database

21

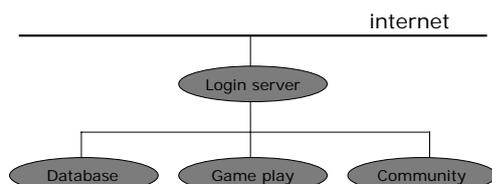
MMORPG Servers

- Stand alone servers
- Distributed system

22

Standalone Server

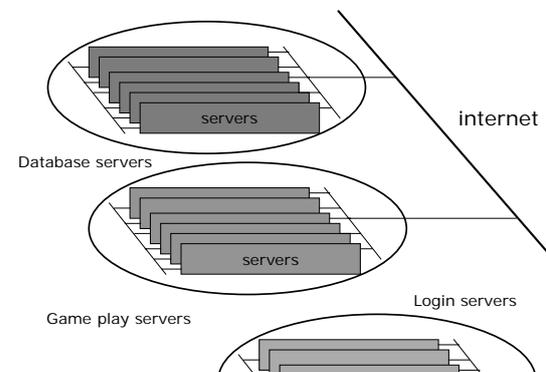
- Using a set of large servers as the game servers
- Each server plays as a single function



23

Distributed System - Concept

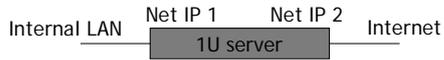
- Distributed PC clusters



24

Distributed PC Clusters

- 1-U Web Server Based on PC Architecture
- Two CPUs
- Two Network IPs



25

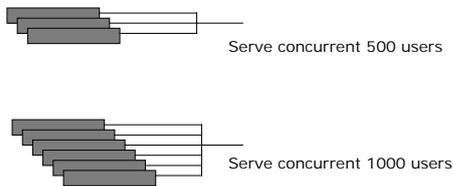
Distributed System - Features

- Two IPs
- Scalability
- Fault tolerance
- Load balance
- Dynamic zoning
- "Blade Servers"

26

Distributed System - Scalability

- Regional operation consideration
- Cost effective
- Flexible capacity

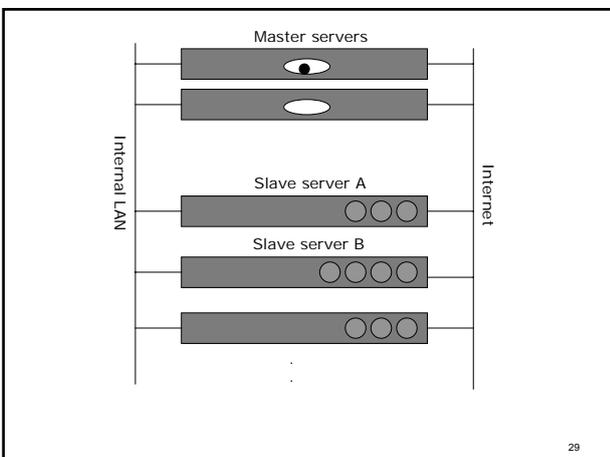


27

Distributed System - Fault Tolerance

- All servers can not down in anytime.
- If some one is going to failed,
 - For distributed servers, the jobs must be transferred to the other servers.
 - For standalone server, use redundant policy.

28



29

Distributed System - Load Balance

- Distributed MMOGs always do the load balance by In-house approach.
- Load ?
 - CPU bound
 - Memory bound
 - Network bandwidth

1 server = 500 concurrent players

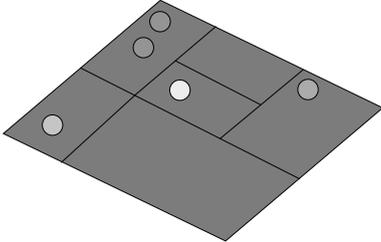
10 servers ✗ 5000 concurrent players

! It's very important to move the jobs on a crowded server to another ones

30

Zone Concept

- A "zone" is logically a region of game happening on servers.
- We always map the "zone" to a physical 3D scene.
- A zone is not really a physical hardware server but a process usually.
- A region communicates to the players directly (in the same memory block)



31

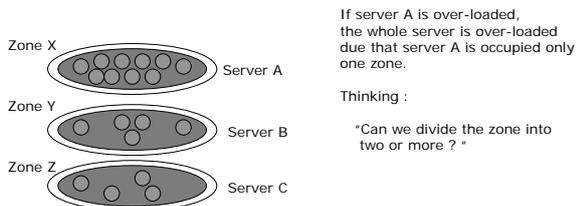
Interaction within/between Zones (Game Play)

- Within the zone
 - Combating
 - Chatting
 - Transaction
- Between zones
 - Messaging
 - Transaction (remote)
 - Banking

32

Load Balance Using Zone Moving

- Under the concept of "zone moving", we can move the zones between hardware servers to achieve the load balance.
- But is this a total solution to solve the load balance ?
- Condition :



33

Dynamic Zoning

- Dynamically adjust the zones to meet the loading requirement.
- Moving zones between hardware
- Divide the zone into small ones according to the load and move the smaller ones to different hardware.
- For the players in the same game zone but running on different process or machine, the data synchronization is becoming a challenge job.

34

Zone = A 3D Scene

- A 3D scene
 - 3D models
 - Moving objects
 - NPCs
 - ...
- Zones are physically neighboring together
- Using portals to connect the relationship among zones.
- Player travels between zones.
 - Logout current scene and login to another neighboring zone when stepping on the zone portal
 - Player in client will feel a little hanging during the moving

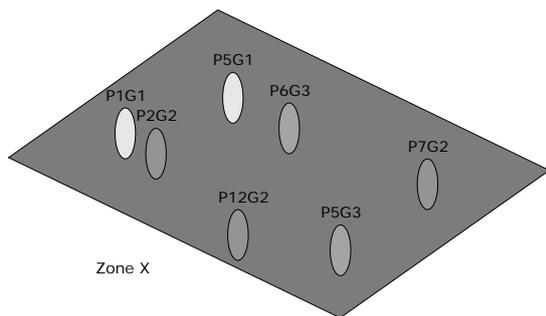
35

Population Group

- A data structure to manage the players.
- When a player logs into one zone, he should be assigned into one available population group and got the ID.
- Players should not be moved between population groups when he is staying in this zone.
- A zone can be divided into several groups.
- Groups can be created/deleted/moved by the servers for load balance management.

36

Players in a Zone



37

The Challenge of Dynamic Zoning (1/2)

- Physical terrain and models are not easy to be divided dynamically in-game.
 - If your zones are coupling with the physical scenes, to divide the geometric data in runtime needs some specific algorithms for 3D scenes
 - From my suggestion, don't do it
 - Find a mechanism that makes the zone dividable
 - And that is the "Population Group"
- One server can have multiple zones running.
- One zone can run on several servers.
 - Hard to code
 - Duplicated memory on servers
 - Synchronization between servers
 - Data communication overhead
- Game Play Sensitive

38

The Challenge of Dynamic Zoning (2/2)

- Running players in the same zone by multiple processes.
 - Duplicated memory on the same server
 - Synchronization between processes
 - Data communication between processes
- Players' attribute data should be classified as :
 - Frequently used between players
 - Must be duplicated between processes
 - Seldom used between players
 - Send between players when necessary
 - Locally used by player himself
- Cross-platform server API design

39