P2P: Napster, Gnutela, Bit-Torrent

Yiwei Sun, Yishi Lu, Kar Ho Tan

I.P2P architecture

1. Definition

- What does Peer-to-Peer Architecture (P2P Architecture) mean?

Peer-to-Peer architecture is a decentralized communications model in which each party has the same capabilities and either party can initiate a communication session.

The peers are computer systems which are connected to each other via the Internet. Files can be shared directly between systems on the network without the need of a central server. In other words, each computer on a P2P network becomes a file server as well as a client.

- How P2P works?

The only requirements for a computer to join a peer-to-peer network are an Internet connection and P2P software. Common P2P software programs include Kazaa, Limewire, BearShare, Morpheus, and Acquisition. These programs connect to a P2P network, such as "Gnutella," which allows the computer to access thousands of other systems on the network.

Once connected to the network, P2P software allows you to search for files on other people's computers. Meanwhile, other users on the network can search for files on your computer, but typically only within a single folder that you have designated to share. While P2P networking makes file sharing easy and convenient, is also has led to a lot of software piracy and illegal music downloads. Therefore, it is best to be on the safe side and only download software and music from legitimate websites.

2. Several different Peer-to-Peer models

A. Unstructured networks

Advantages:

- Unstructured peer-to-peer networks do not impose a particular structure on the overlay network by design, but rather are formed by nodes that randomly form connections to each other.(ex: Gnutella).
- Unstructured networks are easy to build and allow for localized optimizations to different regions of the overlay. Also, because the role of all peers in the network is the same, unstructured networks are highly robust in the face of high rates of "churn"—that is, when large numbers of peers are frequently joining and leaving the network.

Disadvantages:

- Flooding causes a very high amount of signaling traffic in the network, uses more CPU/memory (by requiring every peer to process all search queries), and does not ensure that search queries will always be resolved.
- There is no guarantee that flooding will find a peer that has the desired data. Popular content is likely to be available at several peers and any peer searching for it is likely to find the same thing. But if a peer is looking for rare data shared by only a few other peers, then it is highly unlikely that search will be successful.

B. Structured networks

- In structured peer-to-peer networks the overlay is organized into a specific topology, and the protocol ensures that any node can efficiently search the network for a file/resource, even if the resource is extremely rare.
- The most common type of structured P2P networks implement a distributed hash table (DHT), in which a variant of consistent hashing is used to assign ownership of each file to a particular peer. This enables peers to search for resources on the network using a hash table: that is, (key, value) pairs are stored in the DHT, and any participating node can efficiently retrieve the value associated with a given key.
- However, in order to route traffic efficiently through the network, nodes in a structured overlay must maintain lists of neighbors that satisfy specific criteria. This makes them less robust in networks with a high rate of churn (i.e. with large numbers of nodes frequently joining and leaving the network).
- high cost of advertising/discovering resources and static and dynamic load imbalance.

II. Napster

Napster is a controversial application that allows people to share music over the Internet without having to purchase their own copy on CD. After downloading Napster, a user can get access to music recorded in the MP3 format from other users who are online at the same time. You can simply type in the name of an artist or song, receive a list of what's available, and then download the music from another user's hard drive. Users need to continually check the Napster directory since the music that is available depends on who is online at the time.

III. Gnutella

Gnutella is a system (a large peer-to-peer network) in which individuals can exchange files over the Internet directly without going through a Web site in an arrangement sometimes described as peer-to-peer(here meaning "person-to-person"). Like Napster and similar Websites, Gnutella is often used as a way to download music files from or share them with other Internet users and has been an object of great concern for the music publishing industry. Unlike Napster, Gnutella is not a Web site, but an arrangement in which you can see the files of a small number of other Gnutella users at a time, and they in turn can see the files of others, in a kind of daisy-chain effect. Gnutella also allows you to download any file type, whereas Napster is limited to MP3 music files.

IV. BitTorrent

BitTorrent is a communications protocol of peer-to-peer file sharing which is used to distribute data over the Internet. BitTorrent is one of the most common protocols for transferring large files, and peer-to-peer networks have been estimated to collectively account for approximately 43% to 70% of all Internet traffic (depending on location) as of February 2009. In November 2004, BitTorrent was responsible for 35% of all Internet traffic.As of February 2013, BitTorrent was responsible for 3.35% of all worldwide bandwidth, more than half of the 6% of total bandwidth dedicated to file sharing.

To send or receive files one uses a BitTorrent client; a computer program that implements the BitTorrent protocol. Such clients include µTorrent, Xunlei, Transmission, qBittorrent, Vuze, Deluge, and BitComet. BitTorrent trackers provide a list of files available for transfer, and allow the client to find peers known as seeds who may transfer the files.

What is Cloud Computing?

Louis, Shehar, Yakov

- Cloud computing is a type of computing that relies on *sharing computing resources* rather than having local servers or personal devices to handle applications.
- In cloud computing, the word cloud is used as a metaphor for "*the Internet*," so the phrase *cloud computing* means "a type of Internet-based computing," where different services , such as servers, storage and applications , are delivered to an organization's computers and devices through the Internet.

What Is It Made Of?

- We could say that the cloud is made of layers, usually know as a front end layer, and the back end layer. The network layer is then used to connect end users' devices.

On the front end of a cloud computing system there is:

- a client
- an application
- user interface: what the user "sees and interacts with"

On the back end of the system, there are:

- computers that run the applications
- servers (with a central server(s))
- data storage systems
- basically, what we call "the cloud"

Types of Cloud Computing

Public Cloud

- The whole computing infrastructure is made available to the general public by a service provider
- Don't need to purchase hardware, software or supporting infrastructure
- Clients do not have physical control over the infrastructure
- low-cost, pay-as-you-go model
- Rapid access to affordable computing resources
- A great level of efficiency in shared resources

Private Cloud

- Cloud infrastructure dedicate to one particular customer/organization
- Not shared with others, yet it is remotely located.

- Allow businesses to host applications in the cloud, while addressing concerns regarding data security and control
- On-Premise Private Cloud: hosted within an organization's own facility
- Externally Hosted Private Cloud: hosted by a third party specializing in cloud infrastructure

Hybrid Cloud

- Combine the advantages of both the public and private cloud models
- Increase the flexibility of computing
- Augmenting a traditional private cloud with the resources of a public cloud can be used to manage any unexpected surges in workload
- Example: You can use a public cloud to interact with the clients but keep their data secured within a private cloud.

The Disadvantages of Cloud Computing

Although cloud computing is a staple in data sensitive business models, it comes with a set of disadvantages that users will need to understand and research before committing to a service provider.

These disadvantages include:

- Downtime
- Security and Privacy
- Vulnerability to Attack
- Limited Control Flexibility
- Platform Dependencies
- Costs

What is JavaSpaces?

- Javaspaces is a powerful high level tool for building distributed applications
- Concept of shared network-based space which serves object storage.
- Collection of processes that cooperate through the flow of objects into and out of one or more spaces.
- It simplifies the design and implementation of distributed computing systems

JavaSpaces vs Databases

- In JavaSpaces technology a space is shared network accessible for objects
- Data you store in space is persistent and searchable

- In Databases understand the data is stored and manipulated directly through query languages such as SQL
- JavaSpaces stores entries that they understand only by the type and the serialized form of each field
- There are no queries in JavaSpaces application design and is looked as “exact match”
- In object databases it contains object oriented image of stored data which can be modified and used
- In JavaSpaces, they only work on copies of entries.

JavaSpaces Operations:

- Javaspaces can hold many entries which is a typed group of objects.
- Once entries are written into the space, it can be looked up by certain operations
- The two look-up operations are read() and take()
- The read function either returns an entry that matches or indicates if no match was found
- The take function acts like the read function but if a match is found it removes the entry from space

Primary operations of JavaSpaces:

- Write() : Writes objects into space (Store operation)
- Take() : Gets objects from a space (Fetch operation)
- Read() : Makes copy of objects in a space (Search operation)
- Notify() : Notifies a specified object when entries that match the given template are written into a space (if take function or read function doesn’t find the object, the process will wait until an object arrives)

Example of a distributed application using JavaSpaces

In a multiuser chat system all the messages that are being created are all written to a space which acts as a chat area. Users write message objects into the space while others wait for a new message object to appear, which then is read and its content is displayed. Users also can be kept in the space and updated whenever another user joins or exits the chat. Since the space exits, when a new user joins they can read the entire conversation in the chat system.

Advantages:

- Performance
- Scalability
- Resource Sharing

- Fault tolerance and availability

Disadvantages:

Latency

Synchronization

Partial failure

Advantages of Java Spaces Technologies:

- *It is simple.* The technology doesn't require learning a complex programming interface; it consists of a handful of simple operations.
- *It is expressive.* Using a small set of operations, we can build a large class of distributed applications without writing a lot of code.
- *It supports loosely coupled protocols.* By uncoupling senders and receivers, spaces support protocols that are simple, flexible, and reliable. Uncoupling facilitates the composition of large applications,
- enhances software reuse: we can replace any component with another, as long as they abide by the same protocol.

Message-Oriented Communication Outline

Arshpreet Chahal , Asad Nazir, Joseph Morana

Message-oriented communication is a way of communicating between processes. Message oriented communication can be in the form of synchronous or asynchronous communication, and transient or persistent communication.

Synchronous communication - Sender blocks until message is stored in a local buffer at the receiving host. The sender blocks waiting for the receiver.

Asynchronous communication – Sender continues immediately after it has submitted the message. No need for both the sender and the receiver to execute at the same time.

-The amount of time messages are stored determines whether the communication is transient or persistent.

Transient communication - stores the message as long as both sender and receiver are both executing. If either the sender or receiver is not available then the message is discarded.

Persistent communication - stores the message until the receiver receives it. Messages are stored by communication middleware for as long as needed to ensure delivery of message.

Message Passing Interface (MPI)
MPI is designed for parallel applications using *transient* communication.
MPI is a standardized and portable message-passing system designed by a group of researchers from academia and industry. It is used in many environments and is platform independent.

Message-Oriented Middleware (MOM)
MOM supports asynchronous persistent communication. Processes communicate by message queues. This is done by the sender inserting a message into the queue, then the receiver fetching it from the queue. The sender or receiver doesn't have to be online for the message to be transmitted.
Message Queuing System with Routers
The routers are aware of the network and the queue managers know the nearest router. The routers need to be updated when the queues are added or removed.

Message Broker
The message broker is in charge of making the applications agree on a message format (sender and receiver need to have the same message format). It will transform the incoming messages to the receiving format.

IBM's WebSphere
In IBM's WebSphere, queue managers manage all queues. Messages are put into, and removed from queues. Processes can put messages only in local queues.

Message Transfer:
Messages are transferred between queues. At each endpoint of channel is a message channel agent. Message channel agents (MCAs) are responsible for:

- Setting up channels using lower-level network communication

facilities (e.g., TCP/IP)
- (Un)wrapping messages from/in transport-level packets
- Sending/receiving packets
- Channels are inherently unidirectional
- Automatically start MCAs when messages arrive
- Any network of queue managers can be created
- Routes are set up manually (system administration)

Routing:
By using logical names, in combination with name resolution to local queues, it is possible to put a message in a remote queue. The entry in a routing table is in the form (*destQM, sendQ*). To improve management flexibility, the queue manager uses local alias names.

Advantages

Asynchronicity:
-Message queues give temporary storage when the receiving program is busy or not connected.
-Most asynchronous MOM systems offer persistent storage to backup the message queue. Sender and receiver do not need to be connected to the network at the same time. If the receiver application fails, senders can continue without being affected.

Routing:
Many message-oriented middleware implementations depend on a message queue system. Some implementations permit routing logic to be provided by the messaging layer itself, while others depend on client applications to provide routing information or allow for a mix of both paradigms.

Transformation:
The message sent to the destination does not have to be the same from the message received. The MOM system can transform the message to meet the requirements of the receiver.

Disadvantages

-require an extra component in the architecture (message broker), can lead to reduction in performance and reliability; more expensive
-many inter-application communications have an intrinsically synchronous aspect, with the sender wanting to wait for a reply to a message before continuing. Because message-based communication inherently functions asynchronously, it may not fit well in such situations.

Distributed Gaming Architecture

Larry, Michael, Joseph, Andrew

What is a distributed architecture?

- A distributed architecture is the implementation by which the components of a distributed system communicate with each other.
- Pros: reliability, accessibility, concurrency, scalability
- Cons: security threats, more complex, unpredictable

Types of distributed architecture:

- Client-server: A central server provides resources to many clients.
    - Thin client: The server does the processing.
    - Thick client: The server provides the data. The client does the processing.
- 3-tier: Consists of a client, server, and database (aka presentation tier, application tier, and data tier).
- Peer-to-peer: Each peer can be both a client and a server.

Centralized client-server:

- Many clients connect to a single server.
- Pros: Simple to administrate, maintain, and locate resources.
- Cons: Difficult to scale, high cost of ownership, less redundancy.

Peer-to-peer:

- Pros: Computation is split between different systems; lowers bandwidth requirements.
- Cons: Difficult to achieve game state consistency; hacking is easier.

What is distributed gaming?

- The users become resource providers.
- Provides availability, reliability, and scalability to online gaming.

Hybrid architecture:

- Many modern games combine different topologies to achieve their benefits while limiting drawbacks.

How it works:

- Resources are distributed among a network of peers, and multiple peers may have the same resources.
- Some resource providers may be considered "trustworthy" by specific consumers.
- Peers can locate each other and can switch between different resources.

Examples of distributed architectures:

- Colyseus, Mimaze, Improbable, Meesha Network
    - Colyseus partitions the game world state and the execution among participants. Colyseus also uses efficient object location and speculative prefetching.
    - Meesha Network Presents a simple API to create game-specific resources and the game network in which they exist.

Distributed gaming outline

Advantages and Disadvantages:

- Advantages of Distributed Gaming
  - Failures on the system don't affect others
  - Can help reduce costs by not having to run a server
  - Can be scaled easily
- Disadvantages
  - Can be complex to implement
  - More difficult to prevent cheating
  - Lowered security

Kerberos

Hou In Choi, Changlin Li, Yuqian Zhang

Definition of Kerberos

Kerberos is a computer network authentication protocol which works on the basis of 'tickets' to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner. Its designers aimed it primarily at a client–server model and it provides mutual authentication—both the user and the server verify each other's identity. Kerberos protocol messages are protected against eavesdropping and replay attacks.

Goal

1) Never transmit unencrypted passwords over the network, i.e. "in the clear".
2) Protect against the misuse of intercepted credentials (also called "replay attacks").
3) Do not require the user to repeatedly enter a password to access routine services.

Cryptographic algorithm

The cryptographic algorithm typically employed in Kerberos is Data Encryption Standard (DES), although the modular design of Kerberos allows alternative encryption libraries to be used. DES is a standard algorithm for "private-key" cryptography.

Important terminology

AS (authentication server): A server that issues tickets for a desired service which are in turn given to users for access to the service.

KDC (Key Distribution Center): A service that issues Kerberos tickets, and which usually run on the same host as the ticket-granting server (TGS).

TGT (Ticket-Granting-Ticket): A special ticket that allows the client to obtain additional tickets without applying for them from the KDC.

TGS (Ticket-Granting-Server): A server that issues tickets for a desired service which are in turn given to users for access to the service. The TGS usually runs on the same host as the KDC.

How it works

1) User Client-based Logon
2) Client Authentication
3) Client Service Authorization
4) Client Service Request

Advantage

1) Mutual Authentication: both the user and the server verify each other's identity.
2) Eliminate the transmission of unencrypted passwords across the network.
3) Eliminates the threat that packet sniffers would otherwise pose on a network.

Drawback & limitations

- Kerberos has strict time requirements, which means the clocks of the involved hosts must be synchronized within configured limits.
- Single point of failure: It requires continuous availability of a central server.
- The administration protocol is not standardized and differs between server implementations.
- Clock synchronized between server, client and Kerberos (timestamp used to prevent attack.

WRITTEN BY:
AMMAD KHAN
NICHOLAS LARIS
MARTINO NIKOLOVSKI

Contents

Introduction of cluster computing

The computing industry can be divided into 2 eras: **Sequential Computing Era** and **Parallel Computing Era**

High performance parallel computing systems including these most common ones: Massively Parallel Processors (MPP), Symmetric Multiprocessors (SMP), Cache-Coherent Non-Uniform Memory Access (CC-NUMA), **Distributed Systems** and **Cluster.**

Cluster Computer and its Architecture

Each cluster is consisted of a collection of interconnected computer working together as a single, integrated computing resource that can be single or multiprocessor system with memory, I/O, and an OS. Usually a cluster contains minimum 2 nodes that are connected in a single cabinet or they are physically separated and connected over local area network and appear as single system to users and applications.

Cluster Classifications

-Application Target
-Node Ownership
-Node Hardware
-Node Configuration
-Levels of Clustering
-Node Operating System

Advantages and Disadvantages of using Clusters

-Relatively cheap
-High Performance
-Use single system image
-High Availability
-Programmability
-Problem in finding fault on individual node
-Difficult to handle by a Layman

What is Apache Hadoop?

-Hadoop is a software library that utilizes clusters to improve performance
-Provides an interface for processing BIG data sets
The benefits that come from Hadoop come down to two main features:
- Hadoop Distributed File System (HDFS) - Used to store our large data across a distributed file system
- Hadoop MapReduce: System used for parallel processing of large data sets
HDFS -Hadoop's way of spreading files over computer cluster
MapReduce -Spreads workload closer to data, rather than serial calculations

Disadvantages to Hadoop

-Coding mapReduce requires a lot of code and mastery of difficult APIs
-Hard to create long complex processing jobs with mapReduce
-Can perform pretty slowly when given large programs (map > reduce >map > reduce…)

Apache Spark

-Successor to Hadoop
-Does not have a file system, so usually is used with HDFS. Only implements better mapping
How does it work?
-Uses a Resilient Distributed Dataset (RDD) as an interface to view data stored in HDFS
-Stores a subset of data, as a result of user-called functions

-Each RDD shows a lineage of where it originated, so we know exactly what was done to it, and so that if a node fails, we don't lose any data (fault tolerance)

Memory
-By default, it will use memory-only caching (on which it can perform 100x faster than Hadoop)
-If needed or told to, it can use the disk for storage
-You decide where and how everything is stored

Cluster managers
-Mesos, Hadoop YARN, Standalone - responsible for starting node, restarting nodes, distributing workload

Spark Big Data Analytics
-following are the types of processing that are performed in spark (batch, interactive processing, interactive analytics

Spark Stack
-Spark Stack consist of the following main components
Spark Core, ML Machine learning, GraphX Graph Processing

Types of Data
-The data given to the Spark can be of the following format. HDFS (Hadoop Distributed file format), CSV (comma separated values), tab-delimited, txt file, JSON file, Sequence File, JDBC

Persistence of Data in Spark
-Persistence of data means that if Spark need to perform the same operation over and over, its saves the in memory computation in RAM to save on computational power.

The Lineage Concept
-The lineage concepts explain that how Spark takes multiple HDFS files from disparate nodes, first convert it into a Hadoop RDD file and then output the file in a singular file regarded as Lineage.

Types of shared variables in Spark
-The types of shared variables used in Spark: Accumulator, Broadcast variables

Simple stats operation
-Simple Statistic Operation includes: Count, mean, sum, min, max, variance

Spark interfaces with languages
-Can work with following Language: Java, Scala, R, Python API's

Efficiency of spark
-Concepts of Lazy executions.

Other Spark API
-Spark SQL, Spark Streaming, Spark ML, GraphX

Spark Interaction Demonstration
-Creating an RDD
-Showing content of RDD
-Transformed RDD
-Union of RDD's
-Line contains the specific keyword words
-Making alphabet capitalized

-Key/value exercise
    -Word count
-Spark Web UI (User Interface)

References

1. Cluster Computing at a Glance
   http://academic.csuohio.edu/yuc/hpcoo/lect/chapter-B1.pdf

2. A Comparative Analysis: Grid, Cluster and Cloud Computing
   http://www.ijarcce.com/upload/2014/march/IJARCCE9B%20%20a%20%20anjan%20A%20Comparative%20Analysis%20Grid%20Cluster%20and%20Cloud%20Computing.pdf

3. Dean Wampler, Ph.D: https://github.com/deanwampler/spark-scala-tutorial
4. Hadoop Documentation: https://hadoop.apache.org/docs/r2.7.2/
5. Spark Documentation: http://spark.apache.org/docs/latest/
6. Spark instructional guide: http://www.liondatasystems.com/courses