2019/10/13

The 1st assignment of the distributed system course

罗咏佳 软国1701 201794069

- **What are the distributed components?Please give some examples of components in some common distributed systems.**
  - zookeeper
  - ETCD
  - cassandra
  - HDFS
  - MapReduce
  - Yarn
- **List at least 5 common software architecture patterns and their usage, advantages and disadvantages.**
  - **Layered pattern**
    - This pattern can be used to structure programs that can be decomposed into groups of subtasks, each of which is at a particular level of abstraction. Each layer provides services to the next higher layer.
    - Usage
      - General desktop applications.
      - E commerce web applications.
    - Advantages
      - A lower layer can be used by different layers.
      - Layers make standardization easier because we can clearly define levels.
      - Changes can be made within the layer without affecting other layers.
    - Disadvantages
      - Not universally applicable. In some cases, some layers may be skipped.
  - **Client-server pattern**
    - This pattern consists of two parties; a server and multiple clients. The server component will provide services to multiple client components. Clients

request services from the server and the server provides relevant services to those clients. Furthermore, the server continues to listen to client requests.

- Usage
  - Online applications such as email, document sharing and banking.
- Advantages
  - A good set of services is set up and users can request their services.
- Disadvantages
  - Requests are usually handled in a separate thread on the server. Interprocess communication can cause additional overhead because different clients have different representations.

- **Master-slave pattern**
  - This pattern consists of two parties; master and slaves. The master component distributes the work among identical slave components, and computes a final result from the results which the slaves return.
  - Usage
    - In database replication, the master database is regarded as the authoritative source, and the slave databases are synchronized to it.
    - Peripherals connected to a bus in a computer system (master and slave drives).
  - Advantages
    - Accuracy - delegate the execution of the service to different slave devices with different implementations
  - Disadvantages
    - The slave device is isolated: there is no shared state
    - The delay in master-slave communication can be a problem,
- **Pipe-filter pattern**
  - This pattern can be used to structure systems which produce and process a stream of data. Each processing step is enclosed within a filter component. Data to be processed is passed through pipes. These pipes can be used for buffering or for synchronization purposes.
  - Usage

        - Compilers. The consecutive filters perform lexical analysis, parsing, semantic analysis, and code generation.
        - Workflows in bioinformatics.
    - Advantages
        - Show concurrent processing.
        - Easily add filters and the system can be easily extended. The filter can be reused. You can build different pipes by regrouping a given set of filters.
    - Disadvantages
        - Efficiency is limited by the slowest filtration process. Data conversion overhead when moving from one filter to another.
- **Broker pattern**
    - This pattern is used to structure distributed systems with decoupled components. These components can interact with each other by remote service invocations. A broker component is responsible for the coordination of communication among components.

      Servers publish their capabilities (services and characteristics) to a broker. Clients request a service from the broker, and the broker then redirects the client to a suitable service from its registry.
    - Usage
        - Message broker software such as Apache ActiveMQ, Apache Kafka, RabbitMQ and JBoss Messaging.
    - Advantages
        - Allows dynamic changes, additions, deletions, and relocations of objects, which makes developers' publications transparent.
    - Disadvantages
        - The service description is required to be standardized.
- **Peer-to-peer pattern**
    - In this pattern, individual components are known as peers. Peers may function both as a client, requesting services from other peers, and as a server, providing services to other peers. A peer may act as a client or as a server or as both, and it can change its role dynamically with time.
    - Usage

    - File-sharing networks such as Gnutella and G2)
    - Multimedia protocols such as P2PTV and PDTP.
  - Advantages
    - Support for decentralized calculations. It is robust to the fault handling of any given node.
    - Highly scalable in terms of resources and computing power.
  - Disadvantages
    - The quality of service is not guaranteed because the nodes are voluntary. Security is hard to guarantee. Performance depends on the number of nodes.
- **Event-bus pattern**
  - This pattern primarily deals with events and has 4 major components; event source, event listener, channel and event bus. Sources publish messages to particular channels on an event bus. Listeners subscribe to particular channels. Listeners are notified of messages that are published to a channel to which they have subscribed before.
  - Usage
    - Android development
    - Notification services
  - Advantages
    - New publishers, subscribers, and connections can be easily added. Effective for highly distributed applications.
  - Disadvantages
    - Scalability can be a problem because all messages are made through the same event bus.
- **Model-view-controller pattern**
  - This pattern, also known as MVC pattern, divides an interactive application in to 3 parts as:
    - model — contains the core functionality and data
    - view — displays the information to the user (more than one view may be defined)

  - controller — handles the input from the user

  This is done to separate internal representations of information from the ways information is presented to, and accepted from, the user. It decouples components and allows efficient code reuse.

- Usage
  - Architecture for World Wide Web applications in major programming languages.
  - Web frameworks such as Django and Rails.
- Advantages
  - It's easy to have multiple views of the same model that can be connected and disconnected at runtime.
- Disadvantages
  - Increase complexity. This can result in many unnecessary user action updates.

- **Blackboard pattern**
  - This pattern is useful for problems for which no deterministic solution strategies are known. The blackboard pattern consists of 3 main components.
    - blackboard — a structured global memory containing objects from the solution space
    - knowledge source — specialized modules with their own representation
    - control component — selects, configures and executes modules.

    All the components have access to the blackboard. Components may produce new data objects that are added to the blackboard. Components look for particular kinds of data on the blackboard, and may find these by pattern matching with the existing knowledge source.
  - Usage
    - Speech recognition
    - Vehicle identification and tracking
    - Protein structure identification
    - Sonar signals interpretation.

- Advantages
    - It's easy to add new apps. The structure of the extended data space is simple.
- Disadvantages
    - Modifying the structure of the data space is very difficult because all applications are affected. Synchronization and access control may be required.

- **Interpreter pattern**
    - This pattern is used for designing a component that interprets programs written in a dedicated language. It mainly specifies how to evaluate lines of programs, known as sentences or expressions written in a particular language. The basic idea is to have a class for each symbol of the language.
    - Usage
        - Database query languages such as SQL.
        - Languages used to describe communication protocols.
    - Advantages
        - Highly dynamic behavior is feasible. Provides end user programmability benefits.
    - Disadvantages
        - Performance may be an issue because the interpreted language is usually slower than the compiled language.