中国云体系产业创新战略联盟
China Cloud System Pioneer Strategic Alliance

Cloud computing strategic alliance technical standards

HB/P-2020-0003

The Technology Standards of Dynamic Scheduling Framework and Workflow for Online Tasks in Information Service Platform of Large Scale Network

Drafted by: Tongji University，Donghua University

Release date: September 1, 2020

Preface

"Large-scale Network Information Service Platform Online Task Dynamic Scheduling Framework and Process Technical Standards" consists of the following 5 parts:

——Part 1: Scope;

——Part 2: Normative references;

——Part 3: Terms and definitions;

——Part 4: Framework of dynamic online task scheduling system;

——Part 5: Work flow of dynamic online task scheduling system

This standard was drafted in accordance with the rules given in GB/T 1.1-2009.

This standard was proposed by Tongji University.

This standard is under the jurisdiction of the Information Technology Standardization Technical Committee (SAC/TC180).

The organization responsible for drafting this standard: Tongji University

Participated in the drafting of this standard: Donghua University

The main drafters of this standard: Jiang Changjun, Zhang Zhaohui, Ding Zhijun, Yu Jian, Yan Chungang, Zhang Yaying

中国云体系产业创新战略联盟

Introduction

Large-scale network information service platforms, such as Taobao platform and 12306 ticket booking platform, are characterized by large scale tasks and short calculation time. At the same time, this type of platform has many computing platform features of distributed computing nodes. Therefore, large-scale network information service platforms have high throughput performance requirements for task scheduling systems.

Task scheduling is to allocate tasks to resources reasonably to achieve load balancing and high throughput. In order to use resources reasonably and fully, tasks must be optimally scheduled. At present, the more well-known task schedulers include Portable Batch Processing System (PBS), Load Sharer (LSF), Load Leveler, Load Balancer, and Batch Scheduler. These task scheduling systems are mainly used in "high-performance computing" cluster systems, suitable for large-scale task computing with low throughput. In large-scale network information service systems, for large-scale small tasks, there is a lack of a suitable online task dynamic scheduling system and its unified standard specifications. To this end, a large-scale network information service platform online task dynamic scheduling framework and process technical standards are specially formulated.

Technology Standards of Dynamic Scheduling Framework and Workflow for Online Tasks in Information Service Platform of Large Scale Network

1 Scope

This standard regulates the framework and work flow of the dynamic scheduling of high throughput and small tasks for large-scale network information service platforms, provides unified name specifications and definitions for them, and provides for the compilation of other standards for large-scale network information service platforms Reference.

This standard applies to all large-scale network information service platform-related organizations and their products and systems designed, developed, distributed, managed, and maintained, and provides reference specifications for the industry's information service platform.

2 Normative references

The following documents are indispensable for the application of this document. For dated reference documents, only the dated version applies to this document. For undated references, the latest version (including all amendments) applies to this document.

GB/T 1.1-2009 Standardization Guidelines

ISO/IEC 23006-4-2013 Information Technology Multimedia Service Platform Technology Part 4: Basic Services

ISO/IEC 23006-1-2013 Information Technology Multimedia Service Platform Technology Part 1: Architecture

GA/T 739.1-2007 Application Specification of Public Security Request Service Platform Part 1: Application Service Description

ISO/IEC 23006-5-2013 Information technology Multimedia service platform technology Part 5: Service aggregation

GB/T 25470-2010 The functional specification of common technology resource service platform for manufacturing informatization

GA/T 739.2-2007 Public Security Request Service Platform Application Specification Part 2: Request Service Application Interface

GA/T 1038.3-2012 Technical Specification for Fire Protection Public Service Platform Part 3: Information Exchange Interface

GB/T 30290.1-2013 Satellite Positioning Vehicle Information Service System Part 1: Function Description

GB/T 29746-2013 Real-time traffic information service data structure

AS 3965-1991 Information Technology, Open System and Interconnection, Public Management Information Service Definition

GB/T 29841.4-2013 Satellite Positioning Personal Location Information Service System Part 4: Terminal General Specification

3 Terms and definitions

Online task dynamic scheduling: After receiving the task, it will be scheduled immediately according to the current resource status. Task scheduling is to allocate tasks to resources reasonably to achieve load balancing and high throughput. Task scheduling follows the following two distribution principles: (1) Load balancing principle. Tasks should be allocated to processors with an earlier completion time, so that each processor has the same running time. (2) The principle of high throughput rate. Tasks should be allocated to processors with less execution time.

4 The framework of dynamic online task scheduling system

4.1 Overview

In a large-scale network information service platform, the task scheduling system must be able to schedule large-scale small tasks while also meeting the scheduling performance of the system with high throughput. Its characteristics are as follows: (1) On the basic platform of large-scale network information services (such as cloud computing platform), the computing tasks are all described using Resource Specification Language (RSL). The task scheduling system first parses the RSL of the basic platform, and then schedules. (2) The scheduling system is combined with the resource allocation management (Resource Allocation Management, RAM) and Metacomputing Directory Service (MDS) of the basic platform to make full use of the task management functions and platform resource information provided by the basic platform. (3) The scheduling system is oriented to small tasks (calculation time is about a few seconds or tens of seconds), suitable for high task arrival frequency.

4.2 Architecture of task scheduling system

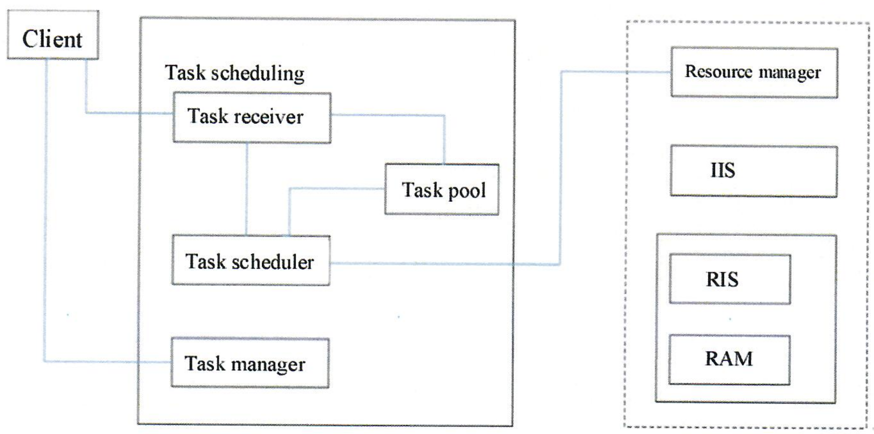

Figure 1 Architecture diagram of task scheduling system

The architecture of the task scheduling system is shown in Figure 1.

The task scheduling is composed of task pool, task receiver, task scheduler, and task manager. The task receiver puts the received task into the task pool. Resource managers collect resource information for use by task schedulers. The task scheduler is responsible for scheduling tasks in the task pool using scheduling strategies and creating a task manager for each task. Task managers are equivalent to customer agents, managing tasks according to customer needs. Resource Allocation Management (RAM) module, Resource Information Service (RIS) module, Information Index Service (IIS) module are the modules provided by the basic platform itself, used to collect the resources of the basic platform information.

(1) Task pool

As shown in Figure 2, the task pool is a piece of memory shared by task receivers and task schedulers for temporary storage of tasks. When the task submission speed is higher than the task scheduling speed, the tasks are queued in the pool and waiting for scheduling. Shared memory uses system call application. The task pool is an array of structures, and each structure stores a task. To facilitate the use of task pools, two linked list management are used to manage them. The two linked lists are the task list (job_list) and the idle list (idle_list). The structure in the task list is for storing jobs; the structure in the free list is empty. In the basic platform, tasks are described using RSL. The RSL specifies the resources required for the task, the full path name and parameters of the executable program, the full path name of the input file and the output file, etc. A client's job consists of three parts: client IP, port, and RSL. The IP and port are used to determine the client program so that the results can be sent back. The RSL clarifies the calculations required by the customer. As a critical resource, the task pool is protected by semaphores in order to avoid race conditions. The task receiver and task scheduler must access the task pool in the critical area. Entering or exiting the critical area requires P (request resource) or V (release resource) operations respectively. Use the system call to apply for the semaphore.

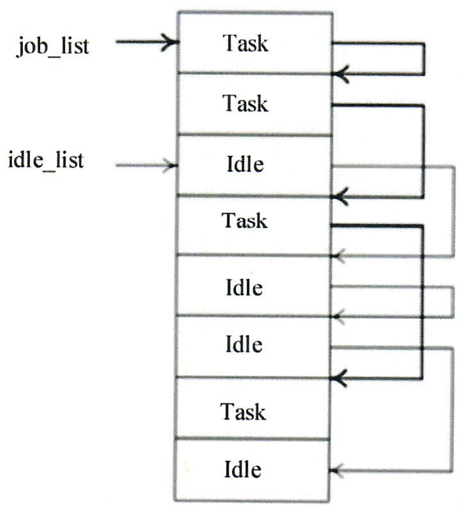

Figure 2 Task pool structure of the scheduler

(2) Task receiver

The task receiver is the background port monitoring process, responsible for receiving tasks submitted by customers. Clients use sockets to communicate with task receivers. After receiving a task on the port, the receiver enters the critical area, picks an idle node from the idle_list of the task pool, fills the task into the node, links the node to the job_list, exits the critical area, and continues to monitor the port. In the specific implementation, the task receiver needs to do different processing according to the different situations of the task pool. There are two special situations that need to be noted: (1) The task pool is full, and the task receiver cannot obtain free nodes. The task receiver uses a system call to enter the suspended state. (2) The task pool is empty. After the task receiver stores the task just received in the task pool, it uses a system call to send a signal to the task scheduler to wake it up.

Task submission to remote computing nodes: Before submitting the task, the task scheduler needs to consider the resources requested by the task, which can be divided into the following three situations: (1) If the task requests too many resources, any computing node cannot meet its needs , The scheduler will delete it and notify its submitter. (2) If the requested resource is more than the resource of the computing node, the scheduler returns the task to the task pool, and the scheduler selects the next task. (3) If the resources can meet the demand for resources, matching can be achieved. In the specific implementation, there are two special situations that need to be noted: (1) After all tasks in the task pool are scheduled, the task pool is empty. At this time, the scheduler uses the system call to enter the suspended state, waiting for the task receiver to signal to wake it up. (2) The original task pool is full. After the task scheduler finishes scheduling a task, there is an idle node in the task pool. At this time, the task scheduler must use a system call to send a signal to the task receiver to wake it up.

(3) Task scheduler

In view of the high response rate and small task characteristics of large-scale information service platforms, the scheduling system should adopt dynamic online scheduling. When there are tasks to be scheduled in the task pool: (1) The task scheduler takes out a task according to a certain priority and analyzes its RSL; (2) Obtains resources through the resource manager; (3) Submits the task and creates a corresponding task Task manager.

(3) Task manager

The application program interface (API) provided by the basic platform can: (1) obtain the status of the task; (2) cancel the task being executed, kill the process, and reclaim the resources occupied by the process. In addition, the scheduling system should implement the following 3 APIs for the return of result data: (1) Read data from the result file; (2) Use Socket to send data; (3) Open the Socket port and receive data.

(4) Tasks of the information service platform

The task consists of 3 parts: client IP address, client port number, and RSL. The client IP address and client port number are used to locate the client program. RSL is used to describe the task requested by the customer, specifying the path, parameters, input and output files, etc. of the task.

5 Work flow of dynamic online task scheduling system

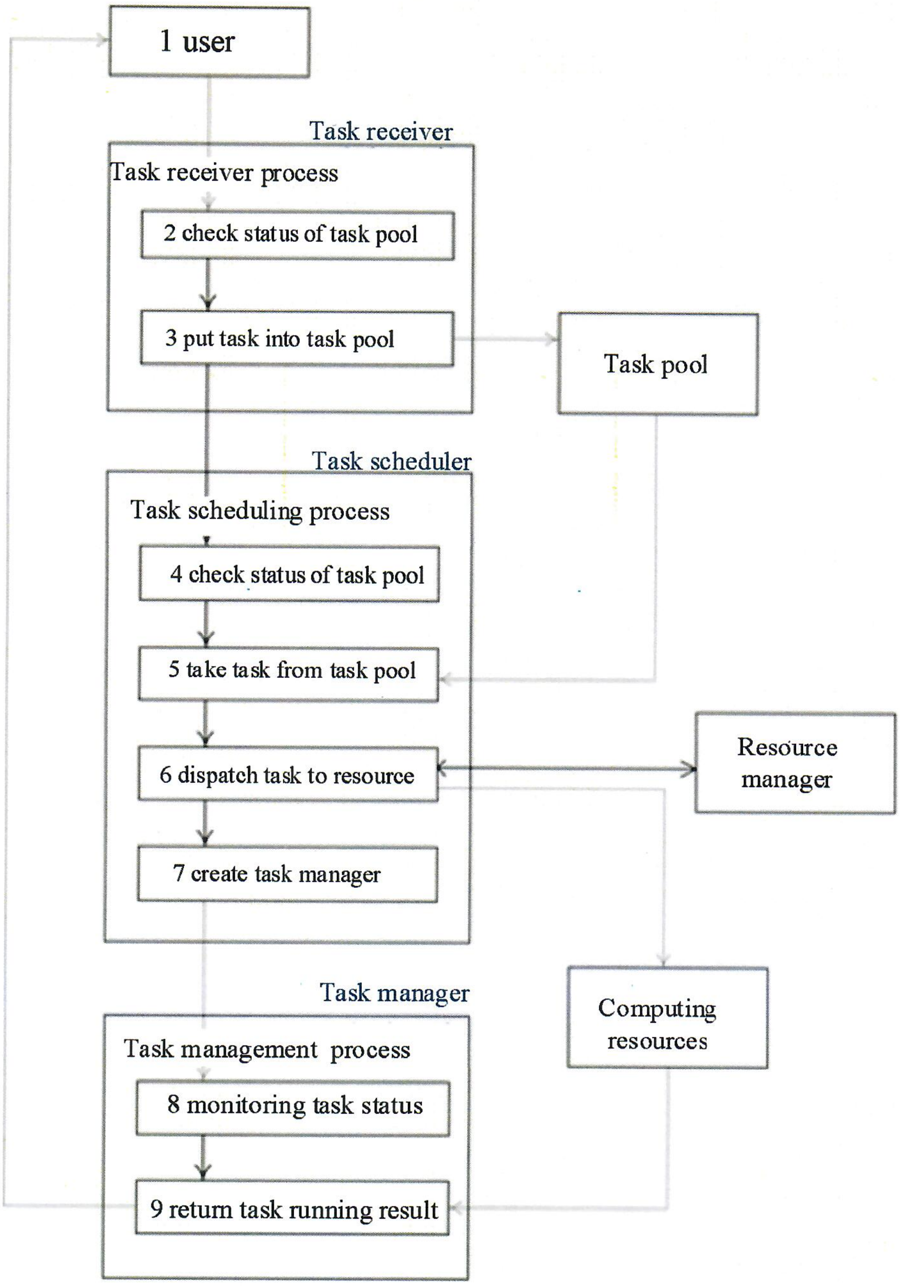

Figure 3 The workflow of the task scheduling system

The working process of the task scheduling system is shown in Figure 3. The specific process is as follows:

(1) The client uses the socket Socket to communicate through the network to send the task to the task receiver.

(2)Task receiving step: Determine whether there is space in the task pool, make corresponding operations, and then continue to listen to the port.

a)If the task pool is empty or there are idle nodes, the task receiver puts the task into the task pool and sends a signal to the task scheduler.

b) If the task pool is full, the task receiver executes a system call and enters the suspended state

(3) Task receiving step: add the task to the end of the job_list in the task pool.

(4) Task scheduling step: Determine whether there are tasks in the job_list of the task pool.

(5) Task scheduling step: If there is a task to be scheduled, the task scheduler takes out a task from the head of the job_list.

(6) Task scheduling step: Analyze the RSL statement of the task, according to the task's demand for resources, assign computing resources to it through the resource manager, and assign tasks to the resources.

(7) Task scheduling step: create a child process (task manager), and hand the handle of the task just scheduled to the task manager.

(8) Task management steps: monitor the current status of tasks, and listen to the port waiting for client requests.

(9) Task management steps: read the results from the task output file, and return the results to the customer.

References

GB/T 1.1-2009 Standardization Guidelines

JR/T 0096.6-2012 China Financial Mobile Payment Networking Joint Part 6: Security Specifications

GB 4943.1-2011 Information Technology Equipment Safety Part 1: General Requirements

JR/T 0097-2012 China Financial Mobile Payment Trusted Service Management Technical Specification

ISO/IEC 23006-4-2013 Information Technology Multimedia Service Platform Technology Part 4: Basic Services

ISO/IEC 23006-1-2013 Information Technology Multimedia Service Platform Technology Part 1: Architecture

GA/T 739.1-2007 Application Specification of Public Security Request Service Platform Part 1: Application Service Description

ISO/IEC 23006-5-2013 Information technology Multimedia service platform technology Part 5: Service aggregation

GB/T 25470-2010 The functional specification of common technology resource service platform for manufacturing informatization

GA/T 739.2-2007 Public Security Request Service Platform Application Specification Part 2: Request Service Application Interface

GA/T 1038.3-2012 Technical Specification for Fire Protection Public Service Platform Part 3: Information Exchange Interface

GB/T 30290.2-2013 Satellite positioning vehicle information service system Part 2: On-board terminal and service center information exchange protocol

GB/T 30290.1-2013 Satellite Positioning Vehicle Information Service System Part 1: Function Description

GB/T 29746-2013 Real-time traffic information service data structure

AS 3965-1991 Information Technology, Open System and Interconnection, Public Management Information Service Definition

GB/T 29841.4-2013 Satellite Positioning Personal Location Information Service System Part 4: Terminal General Specification