Key Technologies of Large Data Stream System

Dong Zhang*
Liaoning Jianzhu Vocational College, Liaoyang, Liaoning 111000, China
*Corresponding author: 27456900@qq.com

Abstract. In general, a large number of data streams are regarded as a continuous, continuous and infinite data series. It is used in network monitoring, sensor network, aerospace, meteorological measurement and control, financial services and other fields. Stream computing has the characteristics of large data scale, continuous, fast and disordered data arrival, easy data loss and diverse data processing. In view of these characteristics, there are new challenges in resource scheduling, data distribution and fault tolerance in task management of flow computing. This paper uses multi-layer Association big data rules to process the data wheel. The experimental results show that compared with the traditional task management method, the proposed method can provide the efficiency and accuracy of task management.

Keywords: Big Data, Streaming Computing, Task Management, Multi-level Association Rules

1. Introduction
With the explosive growth of global information, the era of big data is coming quietly [1-2]. Stream computing is an important computing mode of big data. Different from the traditional batch computing based on the determination of data size, flow computing has the characteristics of unlimited data scale, continuous, fast and disordered data arrival, easy loss of data and diversification of data processing [3]. Facing the new characteristics of streaming big data, how to build a low latency, high throughput and high reliability big data streaming computing system is still an open technical problem.

Task is the basic unit of data processing logic execution in streaming computing system, and also the basic unit of resource scheduling in streaming computing system. Task management is one of the core functions of streaming computing system, which performs resource scheduling and life cycle management on tasks included in streaming applications, including key technologies such as task resource scheduling, data distribution and task fault tolerance [4-7]. Good task management design guarantees the high efficiency and high reliability of streaming computing system.

At present, the research and review of big data streaming computing system mainly analyzes and summarizes the computing requirements, application scenarios, architecture, programming interface, and overall technology of big data streaming computing system, and lacks the detailed investigation and analysis of task management technology in big data streaming computing system. In this paper, the abstract function model of task management of flow computing system is given, and based on this model, the key technologies of task management are investigated and analyzed. Finally, the application, integration and optimization of the above key technologies in the existing mainstream flow computing systems are analyzed.
2. Flow Computing System

Stream computing and batch computing are two main processing modes of big data. Batch computing is to store the data statically first, and then to calculate the stored data; while the streaming computing needs to provide continuous and dynamic services for the data flowing into the system dynamically, and real-time calculation is needed for the streaming data [8-10].

2.1 Application Scenarios and Data Characteristics of Stream Computing

Big data stream computing has a very broad demand and application prospects. At present, big data streaming computing technology has been applied in the fields of Internet data flow monitoring, telecommunication network data management, online transaction data such as finance, stock, e-commerce, real-time data flow processing of Internet of things sensors, social network data flow analysis, health care and intelligent transportation.

In conclusion, in the above-mentioned big data applications, streaming data has the characteristics of unlimited data scale, disordered data arrival, diverse data processing logic and data loss. Among them, the unlimited data scale means that the data is not expected to be stored in static state, but continuously arrives in the process of data processing, and the data scale continues to accumulate. Data arrival disorder refers to that the arrival rule of streaming data changes according to the characteristics of data service, and the arrival order of data is often not known in advance. The diversity of data processing logic means that the application of stream computing has different data processing logic, and the data operations contained in each logic have different time sequence relationship and data dependence relationship. Data volatility refers to that every data unit in the streaming data stream is invalid after one-time acquisition and processing. The life cycle of data unit is short and the historical streaming data is difficult to reproduce.

2.2 Classification and Performance Target of Flow Computing System

The core value of stream computing is to integrate the data from different data sources in real time and process the dynamic data continuously and in real time. To obtain the value of data within its effective time is the primary design goal of big data stream computing system. In order to effectively process the continuous, fast and disorderly arrival of streaming big data, low latency, high throughput and high reliability constitute the goal of streaming computing system. Among them, low latency and high throughput constitute the quantitative characterization of the timeliness of streaming data processing, and high reliability is the characterization of service stability of streaming computing system.

According to the different timeliness of data processing pursued by stream computing, streaming data processing is divided into batch flow processing model and continuous operator model, as shown in Figure 1.

![Figure 1 Classification of streaming data processing](image)

The continuous operator model abstracts the process of flow processing into a directed graph, which is processed by a record as a unit, that is, the data is processed immediately after arrival, which can achieve sub second delay. This model is suitable for the application with low delay as the main goal and high timeliness. Batch stream processing model divides the received data stream into data blocks according to the time slice, and uses MapReduce like batch computing to process multiple records in a short time window. The model is suitable for applications with low timeliness requirements and high throughput as its main goal.
3. Task Management

In the existing big data stream computing system, the data processing function is divided into several processing sub functions, which are executed in sequence or in parallel according to a certain logical relationship, as shown in Figure 2. In the running environment of streaming computing system, there are multiple tasks in each processing sub function to execute in parallel. When the task obtains the corresponding computing resources, it starts to receive the data units continuously arriving, completes the corresponding processing logic, and transmits the result data to the parallel task corresponding to the next processing sub function to continue to execute.

Task is the smallest unit of resource scheduling and data processing in streaming computing system. Task management is resource scheduling and life cycle management of tasks contained in streaming applications. The process of task management is inseparable from the support of resource scheduling, load balancing, task monitoring, fault tolerance and other related technologies.

3.1 Abstract Functional Model

We define the abstract model of task management function of flow computing system, which provides the basis for the comparison of different stream computing system architectures. The first mock exam shows that the model is an abstraction of task management function of general flow computation system, and describes the function unit and control information of the flow computing system as shown in Figure 3. The existing stream computing systems implement these functions with different technical solutions. This abstract model provides a basis for analyzing and comparing the implementation of different stream computing systems.

![Figure 2 Data processing function diagram](image)

![Figure 3 Abstract function model of task management in flow computing system](image)
completed and other information. According to the monitoring information, whether the delay constraints are met can be checked.

3.2 Task Management Technology Analysis

In the abstract function model of task management in streaming computing, there are relatively mature technologies in user/resource interaction and control information storage in current cluster management. Flow computing system mainly focuses on the core function unit of task management in cluster.

The system needs to process it in time to ensure low delay, so it is important for the system to schedule the tasks to be processed to ensure the efficiency of the system. Since the scale of streaming data is infinite and small, in the process of data processing, the distribution strategy of tasks among processing sub functions can affect the load balance between nodes, so the data distribution mode is also important for the timely completion of tasks. Stream data is invalid immediately after one-time acquisition and processing, so it is necessary to provide fault-tolerant technology to ensure the normal execution of tasks in case of node or computing failure.

In order to improve the service quality of streaming computing system, according to the characteristics of streaming data, we mainly analyze the following three aspects of task management technology: resource scheduling, data distribution and fault tolerance technology. Resource scheduling is to allocate resources for computing tasks according to task information, which mainly depends on the scheduling algorithm of tasks; data distribution refers to how the intermediate results of tasks are distributed among more than processing sub functions after tasks are scheduled. Fault tolerant technology refers to how to make fault tolerance to ensure that the task is completed within the specified time when the node or calculation fails during the running process of the task. These technologies not only guarantee the efficiency and reliability of the system, but also guarantee the service quality of the system.

4. Key Technologies of Task Management in Streaming Computing System

This section introduces the task management technology of streaming computing system from resource scheduling, data distribution and fault tolerance technology.

4.1 Resource Scheduling

Resource scheduling is to allocate computing, storage and network resources to streaming computing tasks according to their resource requirements and quality of service requirements. In the multi task concurrent flow computing system, resource scheduling mainly includes task scheduling and resource allocation. Among them, task scheduling is responsible for sorting multiple computing tasks according to certain rules; resource allocation is to allocate resources for computing tasks according to the sorting results.

As shown in Figure 4, according to the rules of task scheduling, task scheduling technology can be divided into two categories: one is based on first come, first served, and the second is based on task priority.

![Figure 4](attachment:image.png)

**Figure 4** Classification of task scheduling

First come, first served scheduling is based on the sequence of waiting services in the job queue. It is generally used in the flow computing system with continuous operator model. Priority based scheduling is based on the priority of tasks and executed according to the priority order of tasks. It is generally used in scenarios with strict timing requirements (such as financial industry) and flow computing systems with batch streaming processing. In this scheduling, tasks are executed in a certain
order, and each task has a priority. The higher the priority is, the higher the execution order of the job is.

4.2 Data Distribution / Segmentation
Data distribution represents how data is sent between processing sub functions. According to the different ways of data distribution, it is divided into random distribution, field distribution and broadcast distribution, as shown in Figure 5.

Random distribution is to randomly send the data processed by upstream tasks to downstream tasks to ensure that each task receives the same number, so random distribution can ensure load balance. Field distribution is grouped according to the same ID or the same key value, so that the data with the same field is distributed to a task for processing. When the amount of data in a field accounts for a large proportion, this distribution strategy will lead to data skew, thus affecting the load balance. Broadcast distribution means that for each data in the upstream task, all the downstream tasks can receive it. This distribution strategy can ensure the semantics of task execution at least once, and also cause data redundancy. Batch processing model mainly involves random distribution and field distribution; continuous operator model includes the above three distribution methods.

Figure 5 Classification of data distribution

Data segmentation for the flow computing system of batch flow processing model, this type of system divides the input data into data blocks according to the time slice, and takes the data block as the unit during the task execution. The size of the data block is determined by the arrival rate of the input data. When the arrival rate is high, the amount of data received in the same time slice is large. The task is divided into time slices, and users can configure the size of time slices when submitting applications.

4.3 Fault Tolerant Technology
Fault tolerance refers to the characteristic that the computer system can still work normally in the presence of faults. Fault is very common in distributed system, and in streaming computing system, data is usually processed on multiple computing nodes. The failure or failure of processing nodes may increase the delay of data processing. If the primary node fails, it may lead to data loss or error results. These failures have adverse effects on the query results of data.

All fault-tolerant methods depend on some form of replication. In this paper, we divide fault-tolerant techniques into data fault-tolerant and computational fault-tolerant. Data fault tolerance is divided into immediate checkpoint and periodic checkpoint; computational fault tolerance is divided into active fault tolerance, passive fault tolerance and upstream fault tolerance, as shown in Figure 6.
5. Introduction of Task Management Technology in Flow Computing System

At present, there are many stream computing systems, and the research on these systems is in progress. Although the case of system research is not thorough, we try to classify the task management technology of the streaming computing system as much as possible. In this section, we mainly focus on the task management technology of streaming computing system, rather than the overall characteristics of a system.

5.1 Spark Streaming

Spark streaming is an application framework based on spark, which is used to support continuous stream computing. Its design goal is to provide near real-time data processing for statistical analysis of websites. Spark streaming is to decompose stream computing into a series of tiny batch jobs according to time slice. Each job is divided into a series of tasks. The Spark scheduler distributes tasks to the executor of each node, which is the smallest unit of scheduling and execution in Spark.

5.2 Storm

Storm is a free and open source distributed real-time computing system. It provides solutions for real-time analysis, online machine learning, continuous computing, distributed remote call, etc. Storm adopts master-slave architecture. Nimbus daemons of master node are responsible for assigning tasks to other machines and fault detection; supervisor daemons on work nodes are nodes where real-time data processing jobs run. Among them, topology is the activity unit of calculation, and task is the logical unit of task; in addition, there is a special coordination node to maintain and manage the cluster state, that is, the node running zookeeper service process. Zookeeper is a high-performance, distributed application coordination service, which provides consistency service for distributed applications. The default resource allocation in storm is static, and the processing components are evenly placed on each node by the master node in a polling way for each topology submitted by the user. This allocation method does not consider the performance of nodes, load changes and communication pressure between nodes.

Storm's multiple tasks run in one process, and it is difficult to locate problems. Storm adopts upstream fault tolerance to ensure data transmission, and adds ack mechanism in data transmission. The Acker will track whether the tuple generated in each task is completed according to the ID of the stream sent from the data source. At the same time, the data will be saved by checkpoint immediately when the data enters the system, so that the data can be sent again and executed in case of processing error.

Yahoo! S4 is a general, distributed, scalable, fault tolerant, pluggable platform for processing continuous data streams. The motivation is to solve the practical problems of commercial search engines. In addition, S4 system is also used to process global flight monitoring data flow.

S4 adopts decentralized and symmetric architecture, and each node has the same function. The logical unit of a stream in S4 is an event. Each event is A key attribute (K, A) data item, and its type is
represented by the event type; S4 applies the smallest unit PE (Processing Element) of data processing, and each PE is responsible for processing the events of the specified event type, and only processing the events of its corresponding key value.

6. Conclusion
As an important computing mode of big data, the importance of streaming computing is also increasing. As one of the core functions of big data stream computing, task management mainly manages resource scheduling and life cycle of streaming computing tasks. Stream computing has the characteristics of large data scale, continuous, fast and disordered data arrival, easy data loss and diverse data processing. In view of these characteristics, there are new challenges in resource scheduling, data distribution and fault tolerance in task management of flow computing.

References
[1] Wang Yizhi, Zhang Xudong, Xiong Wei. Optical Field Camera Vision Measurement Error Analysis. Acta Photonica Sinica, 2017, 046 (011): 113-124
[2] Long C, Li G, Hongxing Y. 3d Dynamic Object Reconstruction Technology Based on Light Field Rendering. Journal of the University of Chinese Academy of Sciences, 2009, 26 (6): 781-788
[3] Yang Fan, Yuan Yan, Zhou Zhiliang. Study on Evaluation Method of Optical Field Camera Imaging Quality. Modern Electronic Technology, 2011, 191(1):146-156
[4] Yin Xiaogen, Zhang Xiaofang, Zhang Weichao. Study on 3d Reconstruction Method Based on Optical Field Digital Focusing. Photoelectron. Laser, 2015 (26): 991
[5] Xia Zhengde, Song Na, Liu Bin. Dense Light Field Reconstruction Algorithm Based on Dictionary Learning. Acta Physica Sinica, 2020, V.69 (06): 63-71
[6] Liu Yanlei, Yuan Libo. Multi Directional Fourier Contour Recognition Method for Steep Edge of Objects. 2013, 7:2(2):729-734.
[7] Tang Yi, Liu Weining, Sun Dihua. Application of Improved Time Series Model in Expressway Short-term Traffic Flow Prediction. Computer Application Research, 2015, 32 (1): 146-149
[8] Wan Ying, Han Yi, Lu Hanqing. Discussion on moving target detection algorithm. Computer Simulation, 2006, 023 (010): 221-226
[9] Liu ya, AI Haizhou, Xu Guangyou. A moving target detection and tracking algorithm based on background model. Information and Control, 2002, 12: 14-19
[10] Pan Quan, Cheng Yongmei, Du Yajuan. Discrete moment invariant algorithm and its application in target recognition. Acta Sinica Sinica, 2001, 23 (001): 30-36