

Research on Data Layout Method of Scientific Workflow Pre-placement Stage in Cloud Environment

Shangwei Gao^{1,a,*}, Cai Chen^{2,b} and Yi Liang^{3,c}

¹ Beijing University of Technology, Beijing, China

² Beijing University of Technology, Beijing, China

³ Beijing University of Technology, Beijing, China

^agaoshangwei520@163.com, ^bchencai@bjut.edu.cn, ^cyliang@bjut.edu.cn

*Corresponding author

Keywords: Cloud computing, Science workflow, Data placement, Intelligent clustering.

Abstract. The data placement of scientific workflow in the cloud environment has been the key to reducing data transmission. The flow of data across data centers not only leads to time overhead but also affects the efficiency of scientific workflow. In view of this situation, this paper, presented a more effective data placement layout strategy to place data than the traditional data clustering method, considers the initial state of data clustering to generate initial centroid, and then based on the data related to the degree of clustering data, resulting in layout scheme. The experiments show that this strategy can effectively reduce the cross-data flow between data and improve the efficiency of scientific workflow.

Introduction

With the development of the Internet, there has been an explosion of various kinds of information and data. In the field of science, data is often processed and the amount of data processed is often quite large. For example, astronomy, high energy physics, biological information and other fields, which the data often reaches TB, and even the order of PB. Prior to that, data processing was more simple and analyzed for small orders of magnitude. For these new scenarios, data processing could require complex processing to utilize the data effectively. In this series of problems in the scientific workflow system, scientists gradually shift the data processing in the process to the Cloud computing platform for better handling. The characteristics of the shared infrastructure provided by the Cloud computing environment itself provide a good platform for scientific workflow. However, there is more consideration needs to be given to make these resources better to improve workflow execution efficiency. In the Cloud computing environment, it can provide the powerful computing resources and high-capacity storage resources, and the cost is relatively lower and simpler.

It is the development and maturity of these technologies that provide a new way to solve the problems encountered in the implementation of scientific workflows. In real life, different organizations are scattered across different locations and have their own computing and storage resources at each location, which can be viewed as an independent data center. These data centers can be accessed through the Internet to create a more powerful Cloud computing platform. For real-world scientific workflows, data as an important part of the data is limited by the size of data and network bandwidth constraints, can't be fully uploaded to a data center, so the need for cross-data center data transmission needs to be carried out Reasonable place.

This paper fully considers the number of data transmission, the amount of data transmitted, the load of data center, etc. in the execution of scientific workflow in the cloud environment and proposes an improved clustering algorithm. First, the initial data centroid is clustered by considering the dependencies between the data. Then combined the multidimensional data dependencies with the size of data for K-Means clustering. Finally, in ensuring the data dependencies, the amount of data size and data center load balancing, the output of a reasonable data layout program will reduce the amount of transmission and improve the efficiency of the entire scientific workflow.

Related work

Scientific workflow is based on the workflow technology, in the field of scientific research, "according to a series of rules in the selected data, the process of specific calculations^[1]". Since scientific workflow^[2] is mainly data-oriented, data processing becomes the main feature of its implementation. The large-scale nature of data makes the research and processing of scientific workflow different from ordinary workflow. In the Cloud computing environment^[3], a new platform is provided for scientific workflows that are computationally complex and resource-intensive. Today's scientific workflows generally have their own approach to managing data, such as Taverna, Pegasus, Kepler, ASKALON and others. These systems all work in specialized areas, but the data management strategy is not fully concerned about the placement of data and the overhead of transferring data between data centers.

In the face of the challenge of data layout of scientific workflow, some scholars and researchers have carried out corresponding research and achieved some results. Literature^[4] proposed a k-means clustering strategy to layout data. In scientific workflows, the complexity of workflows, using the interdependencies of tasks on data, enables data to be stored locally and efficiently when used by tasks. Literature^[5] proposed a data placement strategy using genetic algorithms. Through the modeling of scientific workflow, data and data center, the objective function is defined to evaluate each individual in the population, and its fitness is evaluated. Under the rules of survival of the fittest, the individual with low fitness value should be continuously eliminated and selected, variation, after iteration, produces a suitable data layout scheme. In^[6], by means of hypergraph, partitioning is used to divide the workflow and data into one place. However, the dependence on the data is not enough, the initial state of the data clustering is not properly set up, and the follow-up clustering results have a certain impact.

For scientific workflow, it is a particularly important issue how to place reasonably data because it can significantly reduce the execution time of scientific workflow and improve the efficiency of scientific workflow. Therefore, in this paper, for the data pre-placement phase, the initial state obtained by the hierarchical clustering^[7] is firstly obtained, and then K-means clustering^[8] is performed subsequently to obtain the pre-placement strategy to obtain the optimal layout scheme.

Problem description and Modeling

Define 1 the data set

$$\text{DataSet} = \{\text{data}_1, \text{data}_2, \dots, \text{data}_i, \dots, \text{data}_m\}. \quad (1)$$

DataSet represents all data sets in the workflow. data_i represents a data set. data_i consists of a triplet $\{\text{size}_i, \text{Dl}_i, \text{isMid}\}$. size_i represents the size of data_i . Dl_i represents the mapped data center location. isMid indicates whether it is the intermediate data.

Define 2 the data center

$$\text{DataCenters} = \{\text{Dc}_1, \text{Dc}_2, \dots, \text{Dc}_i, \dots, \text{Dc}_n\}. \quad (2)$$

$$\text{Dc}_i = \{\text{Compute}_i, \text{Capacity}_i\}. \quad (3)$$

DataCenters represents a collection of data centers in a workflow execution environment. Dc_i represents the i th data center. Dc_i represents a two-tuple. Compute_i represents the computing power of the i th data center. Capacity_i represents the carrying capacity of the i th data center. b_{ij} represents the network bandwidth between data centers Dc_i and $\text{Dc}_j (i \neq j)$.

Define 3 the tasks in the workflow

$$\text{Tasks} = \{t_1, t_2, \dots, t_i, \dots, t_n\}. \quad (4)$$

$$t_i = \{task_i, InputDataSet_i, OutputDataSet_i\}. \tag{5}$$

Tasks represents a collection of tasks in the workflow. t_i consists of a triple. $task_i$ represents a task in the workflow. $InputDataSet_i$ represents the input data set of the $task_i$. $OutputDataSet_i$ represents the output data set of the $task_i$.

Define 4 scientific workflow(SWF)

$$SWF = \{Tasks, DataSet, Control, Relationship\}. \tag{6}$$

Tasks represents the task set in the workflow. DataSet represents the data collection in the workflow. Control represents the control relationship in the workflow.

$$Control = \begin{bmatrix} C_{11} & C_{12} & \dots & C_{1|Tasks|} \\ C_{21} & C_{22} & \dots & C_{2|Tasks|} \\ \vdots & \vdots & & \vdots \\ C_{|Tasks|1} & C_{||Tasks||2} & \dots & C_{|Tasks|||Tasks|} \end{bmatrix}. \tag{7}$$

where

$$C_{ij} = \begin{cases} 1 & \text{there is a relationship from } task_i \text{ to } task_j, \\ 0 & \text{there is no relationship from } task_i \text{ to } task_j, \end{cases}$$

Relationship represents the dependencies between tasks.

$$Relationship = \begin{bmatrix} R_{11} & R_{12} & \dots & R_{1|Tasks|} \\ R_{21} & R_{22} & \dots & R_{2|Tasks|} \\ \vdots & \vdots & & \vdots \\ R_{|DataSet|1} & R_{|DataSet|2} & \dots & R_{|DataSet||Tasks|} \end{bmatrix}. \tag{8}$$

where

$$R_{ij} = \begin{cases} 1 & \text{data}_i \text{ is used by } task_j, \\ 0 & \text{others,} \end{cases}$$

Define 5 Data Task Multidimensional Vector

$$DataTaskVector = \{DTV_1, DTV_2, \dots, DTV_i \dots DTV_n\}$$

DataTaskVector represents the relationship between data and tasks. The usage of data is expressed by vectors. DTV_i represents a two-tuple $DTV_i = \{Vector_i, DataSize_i\}$. $Vector_i$ represents a set of vector values $\{Vit_1, Vit_2, Vit_3 \dots Vit_j\}$. Vit_j indicates whether the task t_j requires the data d_i . The value of Vit_j is 0 or 1. 1 represents that t_j requires d_i and 0 represents that t_j does not require d_i . $DataSize_i$ represents the size of data $data_i$. $Vector_i$ represents the normalized vector $\{Vit_1', Vit_2', Vit_3' \dots Vit_j'\}$.

$$Vit_j' = (DataSize_i - U_{avg}) / \partial_i. \tag{9}$$

U_{avg} represents the average size of all the data contained in the $task_i$. ∂_i is the standard deviation of the data size used by the $task_i$.

6. Transmission cost model

$$CostModel_{single}(d_k, Dc_i, Dc_j) \approx d_k \cdot size / b_{ij}. \tag{10}$$

Scientific workflow data is huge. The cost of data transmission across the data center mainly comes from the time delay of transmission. The data-center transmission cost model for the data set is $\text{CostModel}_{\text{single}}(d_k, Dc_i, Dc_j)$. d_k represents the transmitted data. Dc_i represents the source data center for transmission. The Dc_j represents the destination data center for transmission.

cloud environment based on intelligent clustering data layout strategy

In a cloud environment, the scientific workflow system can be implemented in two stages. They are pre-processing and run-time phases.

This paper uses clustering strategy to lay out the data of scientific workflow. In the first phase, we use the improved clustering strategy for data pre-placement. In the second stage, we use the dynamic data adjustment strategy proposed in literature^[9] to place the intermediate data set generated after the task execution into the data center with the minimum transmission time. New tasks are scheduled to the appropriate data center according to the principle of minimum data transmission overhead.

The proposed workflow strategy for scientific workflow data is structured as follows:

- 1) Using hierarchical clustering, the original data set is clustered according to the relationship between data and tasks. Hierarchical clustering results in K clusters ($HCluster_1, HCluster_1, \dots, HCluster_k$);
- 2) Calculate the centroid of each cluster as the initial state of K -Means clustering in the next step. In the i -th cluster dataset, Centroid_i of the i -th cluster is calculated.

$$\text{Centroid}_i = \left(\sum_{j=1}^{|\text{HCluster}_i|} \text{Vector}_j \right) / |\text{HCluster}_i|. \quad (11)$$

- 3) Using the centroid obtained as the initial state, and then performing K -Means clustering, and taking into account the load adjustment in the data center, output the pre-layout;
- 4) In the run-time phase, the data is placed based on the dynamic data adjustment strategy.

In the data pre-placement stage, the following will introduce the use of clustering strategy, whoes difficulties and innovations made corresponding instructions.

In the re-placement stage, data pre-placement makes the initial data eventually produce a static data layout program through the clustering strategy. The program can logically process the original input data set, which enables the initial data to be stored in the local data center as much as possible according to the close relationship between the data and the size of the data. In this placement strategy, taking into account the data center computing, storage capacity and data center load problems, the data is placed in the data center which is computing power and performance. In this way, the static layout of the preprocessing stage can speed up the execution and processing of data and reduce the data transmission time and the transmission volume.

In the pre-placement stage of scientific workflow, the first step is to preprocess the out-of-order metadata and use the method of hierarchical clustering to make the preliminary regionalization by using the correlation between data and task.

SWF Pre-placement Stage Layout Algorithm 1

Input: DataSet, DataTaskVector, DataCenter

Output: kRegions

```

1  Begin
    Input DataSet, Tasks, DataTaskVector, DataCenter
2  // Based on data task relevance, hierarchical clustering
3  Create a new dataSet hDataSet, hDatai ∈ hDataSet
4  Foreach Vi In DataTaskVector
5      hDatai = Vi
6      InitHierarchicalClusterData()
7  hClusterNum = |hDataSet|, k = |DataCenter|

```

```

8      While k<= hClusterNum
9          dist=calculateDistance(hDatai, hDataj), hDatai ∈ hDataSet, hDataj ∈ hDataSet, i ≠ j
10         mergeNearestData()
11         hClusterNum--
12         produceNewClusterDataSet()
13         kRegions=outputHClusterResult() // The DataSet is divided into k regions
14         Output kRegions
15     End

```

In Algorithm 1, line 2 inputs data sets, data tasks multi-dimensional vectors and task sets. Lines 3 to 5 assign the data to the data set hDataSet based on the task and data related vectors. Line 6 initializes the initial state of the hierarchical clustering, using each piece of data in the hDataSet as an initial data block. On the 9th to 12th rows, the data are clustered hierarchically, the distance between any two data blocks is calculated, the two nearest data blocks are aggregated into a new block, and the number of blocks in the cluster is updated Until the number of clustering blocks meets the number of partitions. The 13th to 14th lines output the divided block kRegions results. The next step is to use the region block obtained from the above process to provide a more reasonable source of centroid selection for the initial state of K-Means clustering.

SWF Pre-placement Stage Layout Algorithm 2

Input: DataSet, kRegions,DataTaskVector, DataCenter

Output: dataMapping

```

1      Begin
2          Input DataSet, kRegions,DataTaskVector,DataCenter
           // Based on data task relevance and data size, K-Means clustering
3          Create a new dataSet kDataSet, kDatai ∈ kDataSet
4          Foreach Vi ∈ DataTaskVector
5              kDatai = Vi
6          centroidList =initKMeansData(kDataSet,kRegions) //calculate each region to get
           //clustered centroid
7          While centroidList not change
8              Foreach kDatai In kDataSet
9                  calculateDistanceEachDataToCentroid(kDatai, centroidList)
10                 placeDataToNearestCentroid(kDatai)
11                 recalculateClusteredCentroids()
12             kmeansClusterGroup = kmeansClusterResult()
13             While isExceededDataCenterLoad(kmeansClusterGroup)
14                 AdjustDataToAdaptDataCenterLoad (kmeansClusterGroup)
15             dataMapping = dataMappingDataCenter(kmeansClusterGroup,DataCenter)
16             Output dataMapping //data to datacenter mapping
17         End

```

In Algorithm 2, line 2 inputs the data set, the partitioned data block, the data task multidimensional vector, and the data center. Line 3 to line 5 sets the data set kDataSet. Line 6 selects the centroid of the initial state of the K-Means algorithm and calculates the initial |kDataSet|centroids from the partitioned region obtained in Algorithm 1 and kDataSet. Lines 7-11 calculate the distance from each data KData_i to each centroid, select the centroid closest to it, classify it, recalculate its centroid for each cluster, and iterate until all centroids move again. Line 12 stores the summary results in the group. The first 13 ~ 14 lines to determine whether there is more than data center cluster load case, if more than the load, the last clustering from the center of mass farthest removed in turn. Then the cull data is recalculated to the distance of each centroid at the last clustering, and the nearest ones are classified according to the requirements of the load until all the data meet the data center load. Lines 15-16 put the group in the appropriate data center to implement the data layout mapping scheme.

Simulation experiment

This section mainly focuses on the data clustering strategy based on intelligent clustering in cloud environment. First of all, the experimental environment is introduced, then the simulation experiment is carried out and the experimental results are analyzed. Finally, the data cluster placement of this strategy in the preprocessing stage is evaluated efficiently.

Experimental environment

This experiment runs on Windows operating system. Hardware configurations include CPU i7-3770, quad-core 3.40 GHz, and memory 8GB. In order to ensure the reliability of the experiment, the experimental data were sampled repeatedly for each experiment, and then the experimental data were averaged.

In order to prove the effectiveness of the placement strategy proposed in this paper,

We compare and evaluate the method (PSL) of this article with the Buildtime Only [4] (BO) data layout strategy and the Random [10] data placement strategy.

Evaluation indicators are as follows:

Data Transfer Volume: The amount of data transferred is an important criterion for scientific workflow, and the smaller the amount of data transferred, the more effective the placement strategy is.

Data Transfer Times: The number of data transmission can show the level of implementation of scientific workflow efficiency, but also reflect the effectiveness of placement strategy.

Experimental results

During the experiment, the effect of data placement strategy on data throughput was evaluated by setting changes to the data set.

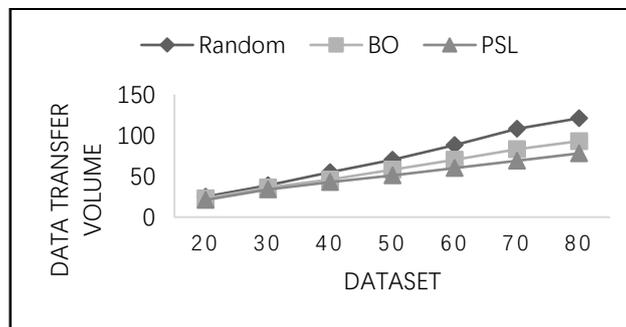


Fig. 1 Impact of number of data sets on data transfer volume with initial data placement

As shown in the figure.1, in the case of an increasing number of data sets, different data placement strategies continuously increase the amount of data transferred and show a gradual upward trend; but the performance of the three strategies is not exactly the same. Compared with the BO and Random strategies, the PSL strategy in this paper has lower transmission cost and slower growth rate.

Under different placement strategies, the ever-increasing data set has an impact on the number of data transfers.

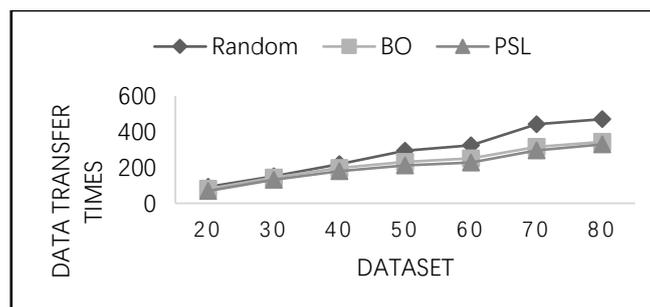


Fig. 2 Impact of number of data centers on data transfer times with initial data placement

As shown in the figure.2, with the increasing number of data sets, the number of data transmission maintains a gradual increasing trend. The strategy proposed in this article can be roughly the same as other strategies at the very beginning. With the increasing number of times, the number of transmission is reduced compared with other strategies.

By changing the number of data centers to assess the impact of data placement strategy on the amount of data transfer.

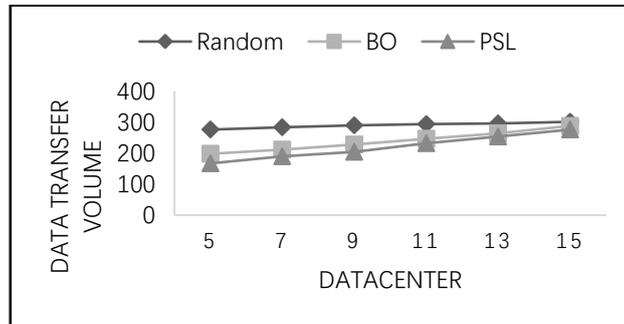


Fig. 3 Impact of number of data centers on data transfer volume with initial data placement

As the figure.3 shows,, the three data placement strategies begin to flatten the data transmission overhead as the number of data centers increases. In the method of this article, the data transmission overhead is less than that of other strategies.

Under different placement strategies, the number of data centers has an impact on the number of data transfers.

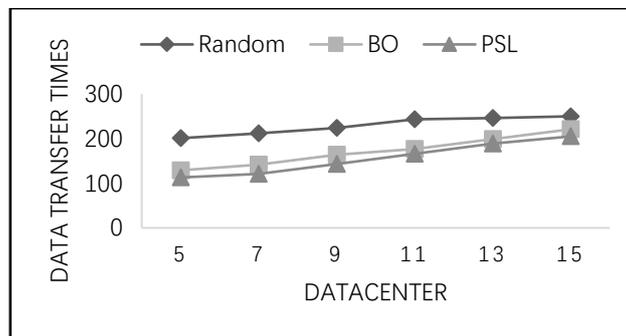


Fig. 4 Impact of number of data centers on data transfer times with initial data placement

As the figure.4 shows, with the number of data centers increasing, the number of data transfers has maintained a gradual upward trend. In the method of this article, the relative number of transmissions is lower.

Based on the above analysis, the PSL placement strategy proposed in this paper is less costly in terms of the amount of data transferred and the number of data centers in the dataset and data center. There is also no significant change in the number of data transfers. This shows that the PSL data placement strategy in this paper has greatly optimized the performance of data layout and improved the execution efficiency of the entire scientific workflow.

summary

This paper analyzes the problem of data placement in the execution of scientific workflow in cloud environment. After considering the factors of data transmission and data center load, this paper puts forward the strategy of PSL data placement, and puts forward the strategy of data placement with other data Experimental comparison verifies the effectiveness of this strategy. In the future, we plan to implement the dynamic adjustment strategy in the implementation stage of data placement. Based on this article, we will further optimize the overall performance of scientific workflow.

References

- [1] LUDÄSCHER B, ALTINTAS I, BOWERS S, et al. Scientific process automation and workflow management[J]. *Scientific Data Management: Challenges, Existing Technology, and Deployment*, Computational Science Series, 2009, 230: 476–508.
- [2] LUDÄSCHER B, ALTINTAS I, BERKLEY C, et al. Scientific workflow management and the Kepler system[J]. *Concurrency and Computation: Practice and Experience*, Wiley Online Library, 2006, 18(10): 1039–1065.
- [3] KUNDU A, BANERJEE A, SAHA P. Introducing new services in cloud computing environment[C]//*International Journal of Digital Content Technology and its Applications*, AICIT. Citeseer, 2010.
- [4] YUAN D, YANG Y, LIU X, et al. A data placement strategy in scientific cloud workflows[J]. *Future Generation Computer Systems*, 2010, 26(8): 1200–1214.
- [5] ER-DUN Z, YONG-QIANG Q, XING-XING X, et al. A data placement strategy based on genetic algorithm for scientific workflows[C]//*Computational Intelligence and Security (CIS)*, 2012 Eighth International Conference on. IEEE, 2012: 146–149.
- [6] LIU X, DATTA A. Towards intelligent data placement for scientific workflows in collaborative cloud environment[C]//*Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW)*, 2011 IEEE International Symposium on. IEEE, 2011: 1052–1061.
- [7] COHENADDAD V, KANADE V, MALLMANN TRENN F, et al. Hierarchical Clustering: Objective Functions and Algorithms[J]. 2017.
- [8] KANUNGO T, MOUNT D M, NETANYAHU N S, et al. An efficient k-means clustering algorithm: analysis and implementation[J]. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2002, 24(7): 881–892.
- [9] Runping WANG , Wanghu CHEN, Ju DUAN. A Cloud Data Placement and Task Scheduling Strategy for Scientific Workflow [J]. *Computer Simulation*, 2015, 32(3): 421–425.
- [10] BHARATHI S, CHERVENAK A, DEELMAN E, et al. Characterization of scientific workflows[C]//*Workflows in Support of Large-Scale Science*, 2008. WORKS 2008. Third Workshop on. 2008: 1–10.