Association for Information Systems
AIS Electronic Library (AISeL)

SAIS 2011 Proceedings | Southern (SAIS)

2011

ETL Pipeline Resource Predictions in Distributed Data Warehouses

Weiwen Yang
Colorado Technical University, weiwen.yang@my.cs.coloradotech.edu

Yanzhen Qu
Colorado Technical University, yqu@coloradotech.edu

ETL PIPELINE RESOURCE PREDICTIONS IN DISTRIBUTED DATA WAREHOUSES

Weiwen Yang
Colorado Technical University
weiwen.yang@my.cs.coloradotech.edu

Yanzhen Qu
Colorado Technical University
yqu@coloradotech.edu

ABSTRACT

Data warehouses of large corporations are increasing in size. Many companies have adopted a distributed data warehouse system, which may store data on many machines. Every day, millions of ETL jobs send data to those warehouses, but some jobs fail due to lack of resources and need to be restarted. Predicting ETL resource demands in distributed data warehouse systems is crucial for efficient use of resources and improved ETL pipeline tasks execution performance. The subject of resource-demand predictions for the ETL data pipeline has not yet been discussed in the literature. This paper discusses a method of predicting resource demands based on history. The linear regression function $y = k\ x + b$ is used to predict memory, as well as disk usage, thus enabling improvement of accuracy of resource usage and the performance of ETL pipeline tasks execution.

Keywords

Data warehouse, database, ETL, resource prediction, business intelligence, linear regression, standard deviation

INTRODUCTION

When a corporation has a large amount of data or many databases, this leads to demand for building a data warehouse in order to support operations, business intelligence decisions, and business reports. In many companies, data warehouse designs often use the de-normalized Star or Snowflake design. These designs use a centralized fact table, which is joined to two or more description dimension tables. The data sources are databases or text files. The raw data is extracted, transformed, and cleaned before it is stored in the staging area, the back-room facility where the data is prepared to be loaded into the data warehouse. ETL refers to the extraction, transformation and loading operations of the data warehouse system. The ETL process moves data from the source to the data warehouse; each job is an execution process to extract, transform, and load data to the data warehouse. There are millions of jobs in the ETL process. The ETL jobs run on anywhere from a few to millions of machines. Some jobs may have too many resources, which leads to wasted resources; other jobs may have too few resources, which leads to failure to run.

The following are the three possible execution cases for resource assignment:

1) If a task requests fewer resources than the task needs, the task will not run.
2) If the task requests too many resources for the task, the task will run but too many of the resources will be wasted.
3) If the task requests just enough resources as needed, the task will execute successfully with resources fully utilized.

The ad hoc resource assignment may lead to many job executions that are not ideal in terms of execution success and resource usage optimization (as shown in Figure 1).

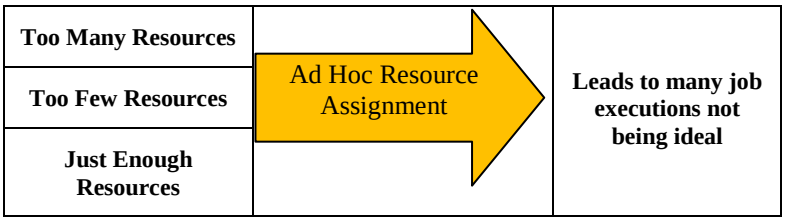

Figure 1. Problems with Ad Hoc Resource Assignment

In order to use resources efficiently, resource-prediction algorithms are needed to allocate resources before the ETL data pipeline is started. Resource-demand prediction for the ETL data pipeline has as of yet not been discussed in the literature.

There are several types of data warehouse architecture (Kimball and Caserta, 2004; Kimball and Ross, 2002; Kimball, Ross, Thornthwaite, Mundy, and Becker, 2008), of which two have most commonly been adopted: The first is the structure-

oriented architecture, in which one layer depends on a number of other layers; the second is the architecture, which depends on how layers are used to create enterprise-oriented or department-oriented views of data warehouses. The data of a warehouse can be from other databases and text files. The raw data is extracted, transformed, and cleaned before it is stored in the staging area. The staging area is considered the kitchen of the data warehouse. Besides extraction, transformation, and loading, ETL operations also include integration, cleaning, filtering, validation, domain verification, logical conversion, merging data, removal of redundancy, deletion of extra data, summation of data, adding time stamps, creation of default values, conversion from one database to another, and restructuring keys.

RELATED WORK IN RESOURCE PREDICITIONS

A number of studies have been done on resource-demand prediction and tasks scheduling for distributed computer networks in the past. In "Predicting Resource Demand Profiles by Periodicity Mining," Andrzejak and Ceyran (2004) stated that corporate data centers utilize the majority of the world's computing resources (p.482). They proposed a method for predicting probabilistic profiles that simulates future behavior. Liang, Klara, and Zhou (2004) proposed a multi-resource prediction model using both correlations to achieve higher prediction accuracy. Tian, Zhou, He, and Zha (2005) proposed that resources be scheduled based on job classifications. Dushyanth., Eno, and Anastassia (2005) described a method to estimate the resource usage for a database system. Dinda (2006) described the design, implementation, and performance of a resource-prediction system that was based on extensible sensor, prediction, wavelet, and communication libraries. Shi, Guo, Yang, and Wu (2009) proposed a resource-prediction system that was based on current state and historic records. The previous research focused on resource prediction for upgrading network systems, resource prediction-based admission control algorithm, resource prediction by probabilistic profiles simulating future behavior, resource prediction by IO-bound and CPU-bound jobs, resource prediction by past values, and resource prediction method using nu-support vector regression. Of all these studies concerning distributed computer networks, none have been conducted on the ETL of data warehouses.

PROBLEM STATEMENT

The ETL data pipeline of the distributed data warehouse runs each day to satisfy the *service-level agreement* (SLA), which contains the quality and data freshness requirement. This mandates that the total runtime of the data pipeline must fall within a certain time limit. However, from the discussion in the introduction, clearly the ad hoc resource assignment method has the problem in ensuring that the tasks in an ETL data pipeline are executed successfully within a predictable total runtime with optimal resource usage (in terms of the least number of executors). This is the foundation for the claim of the problem statement of this paper: ***The ad hoc resource assignment method cannot always guarantee the total runtime and the usage of the computer resources to be optimal.***

METHODOLOGIES

To address the resource assignment issues, this paper presents a linear-regression-model–based general prediction function that computes the resources, leading to lowered task execution failure rates, and thus further improving ETL data pipeline performance.

Linear Regression

The linear regression function $y = k\,x + b$ is used to predict the memory, as well as disk usage, where *y* is the variable for the current value; *k* is the slope; *b* is the intercept; and *x* is the variable for the data size. Variable *y* is also called the *response variable* or *dependent variable*. Variable *x* is also called the *explanatory variable* or *independent variable*. In linear regression, variables *y* and *x* have a linear relationship.

Z score is $\frac{y-u}{\sigma}$, where *u* is the mean, *y* is actual value, and σ is the standard deviation. Given a set of scatter data, the correlation coefficient is used to determine the correlation of the data (Veaux and Bock, 2008).

The correlation coefficient is

$r = \frac{\sum z_x z_y}{n-1}$, where *n* is the number of sample size, z_x is the *z* score of variable *x*, and z_y is the *z* score of variable *y*.

The *z* score of *x* is

$z_x = \frac{x - \bar{x}}{s_x}$, where *x* is the actual value, $\bar{x}$ is the mean of *x*, and s_x is the standard deviation of *x*.

$z_y = \frac{y - \bar{y}}{s_y}$, where *y* is the actual value, $\bar{y}$ is the mean of *y*, and s_y is the standard deviation of *y*.

After substitution, the formula is

$$r = \frac{\sum z_x z_y}{n-1} = (\frac{1}{n-1})\sum z_x z_y = (\frac{1}{n-1})\sum \frac{x-\bar{x}}{s_x}\frac{y-\bar{y}}{s_y} = (\frac{1}{n-1})\sum \frac{(x-\bar{x})(y-\bar{y})}{s_x s_y}.$$

The standard deviation of *x*-axis:

$$s_x = \sqrt{\frac{\sum (x-\bar{x})^2}{n-1}}$$

The standard deviation of *y*-axis:

$$s_y = \sqrt{\frac{\sum (y-\bar{y})^2}{n-1}}$$

Substitute the correlation coefficient *r* by the standard deviation of *y*-axis and *x*-axis.

$$r = (\frac{1}{n-1})\sum \frac{(x-\bar{x})(y-\bar{y})}{s_x s_y}$$

$$= (\frac{1}{n-1})\sum \frac{(x-\bar{x})(y-\bar{y})}{\sqrt{\frac{\sum (x-\bar{x})^2}{n-1}}\sqrt{\frac{\sum (x-\bar{x})^2}{n-1}}}$$

$$= \frac{\sum (x-\bar{x})(y-\bar{y})}{\sqrt{\sum (x-\bar{x})^2 (y-\bar{y})^2}}$$

The correlation coefficient can be calculated from the raw values and the mean. As in the literature, the correlation measures the strength of the linear relationship between the two associated quantitative variables.

The following factors usually are considered to affect the correlation coefficient:

1) The variables must be quantitative.
2) The correlation coefficient can be calculated for any pair of variables. This will make sense only if the two variables have a linear association relationship.
3) An outlier data point can affect the correlation coefficient dramatically: The coefficient should be reported with and without the outlier points.
4) A lurking variable is the hidden variable behind the correlated relationship, which simultaneously affects the other two variables of the linear regression.

Computing the Linear Regression Function

The slope of the line is

$k = r\frac{s_y}{s_x}$, where *r* is the correlation coefficient, s_y is the standard deviation of *y*, and the intercept is:

$b = \bar{y} - k\bar{x} = \bar{y} - r\frac{s_y}{s_x}\bar{x}$, where $\bar{y}$ is the mean of *y*, and $\bar{x}$ is the mean of *x*.

The linear function is

$$y = kx + b = r\frac{s_y}{s_x}x + (\bar{y} - r\frac{s_y}{s_x}\bar{x}).$$

Below, Table 2 is an example of the computing of a linear regression function. The data is processed by an operation. There

may be an intermediate data set generated during processing of the data. It appears that the bigger the data size, the bigger the peak disk usage is, but there are exceptions with some data points.

Peak disk usage (G)	Data size (mb)	Date
2.25	950	03-01-2008
2.38	1050	03-02-2008
2.6	1150	03-03-2008
2.6	1250	03-04-2008
2.75	1350	03-05-2008
3.1	1450	03-06-2008
3.2	1550	03-07-2008

Table 2. Peak Disk Usage vs. Data Size for One Operation

Let *y* stand for peak disk usage; *x* stands for data size:

$$r = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{\sum (x - \bar{x})^2 \sum (y - \bar{y})^2}}$$

Therefore
Mean of peak disk usage: $y = 2.7$
Mean of data size: $x = 1250$
The correlation coefficient is 0.977.
There is very strong evidence that data size is a good indicator of peak disk usage.
Standard deviation on data size:

$$s_x = 200$$

Standard deviation of peak disk usage:

$$s_y = 0.325$$

$$b_1 = r\frac{s_y}{s_x} = 0.977 * \frac{0.325}{200} = 0.00159$$

Because the expression is $\bar{y} = b_0 + b_1\bar{x}$, it is true that

$$b_0 = \bar{y} - b_1\bar{x} = 2.7 - 0.00159 * 1250 = 0.713$$

The intercept is 0.713.
The linear function:

$$y = kx + b = 0.000159x + 0.713$$

Experimental Result

Table 3 shows the 10 operations of the distributed data warehouse system. The operation names are A, B, C, D, E, F, G, H, I and J. The history of disk usage of previous 5 days is in the table. The predicted value is calculated for each operation based on linear regression.

	A (GB)	B (GB)	C (GB)	D (GB)	E (GB)	F (GB)	G (GB)	H (GB)	I (GB)	J (GB)
Day 1	1.40	3.15	0.35	0.50	1.13	0.43	0.82	0.19	2.11	1.31
Day 2	1.58	3.00	0.37	0.52	1.22	0.60	0.65	0.18	2.16	1.43
Day 3	1.79	2.77	0.41	0.57	1.44	0.71	0.62	0.23	2.54	1.65
Day 4	1.91	2.6	0.49	0.65	1.65	0.69	0.59	0.29	2.82	1.85
Day 5	1.95	2.2	0.56	0.73	1.56	0.85	0.50	0.37	3.73	2.21
Prediction of Day 6	2.10	2.07	0.594	0.762	1.85	0.938	0.438	0.386	3.75	2.32

Table 3. The History Disk Usage and Prediction Value

Table 4 compares the result of average estimation and the actual disk usage. The disk usage of sixth day can be estimated by the average of the disk usage of last 5 days. The average deviation of 10 operations by the average estimation method is 24.99% in table 4. If the disk is not enough, the operation will fail. The result from the average estimation method shows that 8 operations don't have enough resource to run. The operation failure rate is 8/10 that is 80%.

MB	A	B	C	D	E	F	G	H	I	J
Average	1.73	2.74	0.436	0.594	1.4	0.656	0.636	0.252	2.67	1.69
Actual	2.02	2.23	0.571	0.76	1.81	0.91	0.481	0.38	3.62	2.25
Difference	-0.29	0.51	-0.135	-0.166	-0.41	-0.254	0.155	-0.128	-0.95	-0.56
% Deviation	14.3	22.8	23.6	21.8	22.6	27.9	32.2	33.6	26.2	24.9
Average % Deviation	24.99									

Table 4. The Average estimation Versus the Actual Disk Usage

Table 5 compares the result of linear regression prediction and the actual disk usage. The average deviation is 4%. If the predicted resource is less than the actual usage, the job will fail. Two of the operations will fail. The failure rate is 2/10 that is 20%. Therefore, the linear regression method can predict the resource properly and reduce operation failure rate.

	A (GB)	B (GB)	C (GB)	D (GB)	E (GB)	F (GB)	G (GB)	H (GB)	I (GB)	J (GB)
Prediction of Day 6	2.10	2.07	0.594	0.762	1.85	0.938	0.438	0.386	3.75	2.32
Actual	2.02	2.23	0.571	0.76	1.81	0.91	0.481	0.38	3.62	2.25
Difference	0.08	-0.16	0.023	0.002	0.04	0.028	-0.043	0.006	0.13	0.07
% Deviation	3.9	7.1	4	2.6	2.2	3	8.9	1.6	3.6	3.1
Average % Deviation	4									

Table 5. Linear Regression Prediction Versus Actual Disk Usage

Analysis

For linear regression, a data point is called a *high leverage point*, so that its *x* value is far away from the mean of *x* values. Too many high leverage points will affect the result of linear regression. A point is called an *influential point* if its omission from the analysis renders a very different model. Outlier points are away from the body of the distribution. Too many outlier points will affect the linear regression. Some data points are possible influential points. A case with high leverage whose *y* value sits on the line, fitting the rest of the data, is not influential. A case with modest leverage but very large residual is influential. It is reasonable to check the linear model using a display of residuals. The *residue* is the difference between data value and expected data value. Some points are high residual points, getting bends when the residual plot is not straight. A data point can be unusual if its *x* value is far from the mean of *x* values. Such a point is said to have high leverage.

To predict resource demands using mean and standard error, the result depends on mean and standard error, as well as the correlation coefficient . The mean and standard error can be computed from the data.

FUTURE WORK

A more complex function can be used to predict resource demands:

In the normal distribution function $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-u)^2}{2\sigma^2}}$ (where *u* is the mean, *m* is the constant, and σ is the standard deviation) or in the autoregressive model, prediction can be studied. Resources can be classified into disk, memory, network bandwidth, and CPU. Different resources may have different behaviors in prediction.

CONCLUSION

Predicting ETL resource demands in distributed data warehouse systems is paramount for efficient use of resources and improved ETL pipeline tasks execution performance. The subject of resource-demand predictions for the ETL data pipeline has not yet been discussed in the literature. In this paper we have presented a method of predicting resource demands based on history. The linear regression function $y = k\,x + b$ is used to predict the memory, in addition to disk usage (where *y* is the current value; *k* is the slope; *b* is the intercept; and *x* is the variable for the data size), thus improving accuracy of resource usage and the performance of ETL pipeline tasks execution.

REFERENCES

1. Andrzejak, A., and Ceyran, M., (2004) Predicting resource demand profiles by periodicity mining, *Sixth IEEE International Conference on Cluster Computing (CLUSTER'04)*, 482.
2. Dinda, P., (2006) Design, implementation, and performance of an extensible toolkit for resource prediction in distributed systems, *IEEE*.
3. Dushyanth N., Eno T., and Anastassia, A., (2005) Continuous resource monitoring for self-predicting DBMS, *MASCOTS, IEEE*.
4. Kimball, R. and Caserta, . (2004) The data warehouse ETL toolkit, Wiley Publishing.
5. Kimball, R., and Ross, M., (2002) The data warehouse toolkit, 2nd ed., John Wiley and Sons, 2002.
6. Kimball, R., Ross, M., Thornthwaite, W., Mundy, J., and Becker, B. (2008) The data warehouse lifecycle toolkit, Wiley Publishing.
7. Liang, J., Klara, N., and Zhou, Y., (2004) Adaptive multi-resource prediction in distributed resource sharing environment, *IEEE*.
8. Shi, L., Guo, L., Yang, S., and Wu B. (2009) A Markov chain based resource prediction in computational grid, *IEEE*.
9. Tian, C., Zhou, H., He, Y., and Zha, L. (2005) A dynamic MapReduce scheduler for heterogeneous workloads, *IEEE*.
10. Veaux, D., and Bock, V. (2008) Stats data and model, Pearson Addison Wesley.