INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN: 2278 – 0211 (Online)

The Replica-Based Data Access Optimization in Distributed Environment

P. Sathiya
Member, IEEE, Department of Computer Science and Engineering,
V.S.B. Engineering College, Karur
K. N. Vimal Shankar
Department of Computer Science and Engineering,
V.S.B. Engineering College, Karur

Abstract:

Distributed systems are groups of networked computers, which have the same goal for their work. According to that, achieve the goal for data distribution that can improve the allocating file chunks on the computing environment. When a new process arrives at the system, the approach verifies if there is any automaton capable of representing it. If automaton is used to estimate resource requirements for this next process. In this system applies many strategies for supporting the online prediction of application behavior, even though the data access operation finds to be though in optimizing the operation. Which checks the operation conducting on the application through the reducing the iteration in migration and replication process with strategies models designed for Schedulers. Distribution and Data access Process has also been made simple through PSO based on scheduler by tracking the Metadata of Data placed in the Datacenters.

Keywords*: Distributed System, PSO (particle swarm optimization), Optimization.*

1.Introduction

The distributed environment is a software system that includes a framework and toolkit for developing client/server applications. Some major components uses in the distributed environment are 1) The Security Server (SS) that is responsible for authentication 2) The Cell Directory Server (CDS) that is the repository of resources and ACLs and 3) The Distributed Time Server (DTS) that provides an accurate clock for proper functioning of the entire cell.

Distributed system mainly available for powerful yet cheap microprocessors continuing advances in communication technology and applications rely on cluster and grid environments. If the systems have uses any computing technologies for a goal-oriented Technologies mainly used for large-scale computing infrastructure currently available in the system that can characterize by both hardware and software heterogeneity. Some challenges have been made from system that involving a different objects designed models such as job scheduling, load balancing, communication protocols, high-performance in hardware architectures, and data access optimization. Migration is also considered in the system for ordering the data that use bring data closer to where it is requested.

In this system mainly consider for replication schema that to increase performance and resilience against failures to access their data optimizing in distributed system. The data access in distributed manner within that organizations then it allows greater flexibility in structure and more redundancy for a system. This work proposes two prediction approaches uses for Particle Swarm Optimization (PSO) technique: 1) the first considers one prediction model for every data access request; 2) the second models and predicts observations based on an average behavior of requests.

In this Particle Swarm Optimization (PSO) technique mainly applies the concept for social interaction to problem solving. The basic concept of PSO technique lies in accelerating each particle toward its pbest and the gbest locations that is, this technique use to easily find position and velocity of the resources.

In this propose system, it invokes a computational method for optimizes a problem that is, iteratively trying to improve a candidate solution with regard to a given measure of quality. This technique to optimize a problem for having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. It

uses a number of agents (particles) that constitute a swarm moving around in the search space looking for the best solution.

So that, it easily to predicts and access replicate resources. Then works still need to employ an extra effort for reducing application execution time, what depends on data access patterns and dynamically predict resources in distributed system.

2.Related Works

Some related works done in the prediction of application behavior and data access prediction approaches.

2.1.Prediction On Behavior

As one of the first works in this area is statistical approach to predict the consumption of CPU, file system I/O, and memory. If so, this automaton is used to estimate resource requirements for this next process. Trace-driven experiments confirm a strong correlation in between next and past executions. It employs [1] machine learning techniques to model requests to storage devices, based on Classification and Regression Trees (CART). It's easy to understand what resources are important in making the prediction. If some resources are missing, they can still make a prediction by averaging all the resources the system.

The model gives a jagged [4] response, so it can work when the true regression surface is not smooth. If it is smooth, though, the piecewise-constant surface can approximate it arbitrarily closely. There are fast, reliable algorithms to learn these predictions.

Mello employ Chaos-Theory [3] concepts and nonlinear prediction techniques to model and predict process behavior over time. Results confirm behaviors can be modeled using Dynamical Systems and, particularly, that Radial Basis Functions are good estimators for future process states. This work mainly based on framework to execute, control, and monitor applications.

A new protocol study that adjusts sending rate according to calculated backlog presents a model to predict the current number of flows which could be useful to predict the future number of flow All of the models presented have either poor accuracy or they need a lot of information to be collected from then work or from the user.

2.2.Data access prediction

To improve performances data access prediction in distributed environments and it as mainly worked on data replication, distribution, and consistency.

Armada [6] builds graph structures to represent processing and data flows. Graphs, representing process also used in resources dependencies and to support decisions on moving data toward processes, thus reducing execution time. Experiments compare applications running on a traditional environment and also on Armada [4] improve network throughput in the resource utilization.

Kim et al. [9] propose a heuristic to improve resources access of the application, i.e., reduce execution time of data-intensive applications. This heuristic attempts to find the lowest cost data source for every process and then results confirm that heuristic outperforms conventional latency based and random allocation approaches.

AL-Mistarihi and Yong [2] propose an approach to select the best replica for applications, that is, which presents the lowest access cost. It considers the Analytical Hierarchy Process (AHP) to solve that problem; [1] they evaluate this approach using the OptorSim simulator and compare it to a random one.

The main drawback of work only considers read operations, what tends to limit their applicability in real-world scenarios. They employ static approaches to optimize access, i.e., mechanisms that do not consider the dynamic system behavior.

3.The Replica-Based Data Access Optimization

This paper proposes an approach to support the prediction of application behavior in an attempt to optimize data access operations on distributed environment. In that to evaluate process behavior and select modeling techniques, which are used to predict of replica based schema.

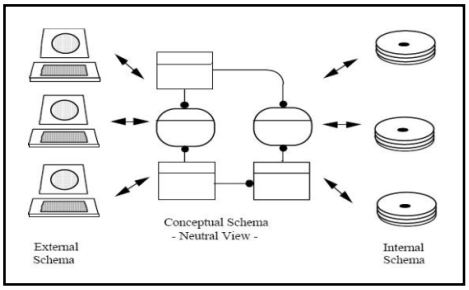

Figure 1: Schema Approach

3.1.Applying Application Knowledge Acquisition

If constructing the grid architecture and it architecture consist of gridlet, user .etc. and also construct transaction application and batch application. It is responsible for monitoring process behavior by using event interception. The interception mechanism is associated with the process under execution. At every function call, this mechanism is informed, being capable of taking further decisions or simply log information. Different libraries and tools provide interception. The most common ones are the Unix DLSym and Unix Ptrace.

3.2.Organization Of Process Behaviors As Time Series

After extracting the application behavior, transform the sequence of read-and-write events in a multidimensional time series. Every event is described by quintuple as which pid is the process identifier that performed the operation, i node is the file identifier, amt is the amount of data read or written to disk, time represents the time interval in between consecutive operations, and op is the operation type.

$$\mathrm{tr} = \{\mathrm{pid}, \mathrm{inode}, \mathrm{amt}, \mathrm{time}, \mathrm{op}\}$$

3.3.*Classifying Data Sets and Assessing the Time Series Generation Process*

All tools are organized according to a tree view and also indicate models for every branch. When a series is deterministic, it is better studied using Chaos-Theory tools, making unnecessary further evaluations. When under that branch, series still must be evaluated in order to select the best modeling approach. Thus, proceed with the third level which verifies series linearity. Nonlinear series can be addressed by using nonlinear modeling approaches.

$$W_{t+1} = W_t \times (1 - \beta) + \mathrm{Op}^{2 \times \frac{\mathrm{Op}}{W_t}} \times \beta.$$

3.4.Prediction Of Process Behavior And Implementation Of Optimization

The prediction is performed on the time series, which represents process behaviors. The measurement considered to evaluate prediction results was the Normalized Root Mean Squared Error (NRMSE defined in which $x_{\max}$ is the maximum and x-min is the minimum observed values, xi is the expected value at time instant is the obtained value at time instant. $z = 1.96\mathrm{f}$

$$\mathrm{NRMSE} = \frac{\sqrt{\frac{\sum_{i=1}^{n} (\hat{x}_i - x_i)^2}{n}}}{x_{\max} - x_{\min}}.$$

In this use a PSO technique for accessing at dynamitic calculation of CPU time get a better performance i.e., to predicate performance in time series. PSO use each particle's movement is influenced by its local best known position and is also guided toward the best known positions in the search-space, which are updated as better positions are found by other particles..The parameter for the data optimization namely –

- Execution time of job request
- Cost of completion of job ie.CPU cost.
- Expected Completion time of job etc,

are given as input to the PSO algorithm.PSO is a meta heuristic as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions.

The goal is to find a solution a for which $f(a) \leq f(b)$ for all b in the search-space, which would mean a is the global minimum. Maximization can be performed by considering the function $h = -f$ instead. Here, the x, v and p and the result g are arrived based on the input given to the PSO. This is expected to move the swarm toward the best solutions

A basic PSO algorithm is then:

1) For each particle $i = 1 \ldots S$ does:
   a) Initialize the particle's position with a uniformly distributed random vector: $x_i \sim U(b_{lo}, b_{up})$, where b_{lo} and b_{up} are the lower and upper boundaries of the search-space.
   b) Initialize the particle's best known position to its initial position: $p_i \leftarrow x_i$
   c) If ($f(p_i) < f(g)$) update the swarm's best known position: $g \leftarrow p_i$
   d) Initialize the particle's velocity: $v_i \sim U(-|b_{up}-b_{lo}|, |b_{up}-b_{lo}|)$.

2) Until a termination criterion is met (e.g. number of iterations performed, or a solution with adequate objective function value is found), repeat:
   a) For each particle $i = 1, \ldots, S$ do:
   b) For each dimension $d = 1, \ldots, n$ do: Pick random numbers: $r_p, r_g \sim U(0,1)$.
   c) Update the particle's velocity: $v_{i,d} \leftarrow \omega\, v_{i,d} + \varphi_p\, r_p\, (p_{i,d}-x_{i,d}) + \varphi_g\, r_g\, (g_d-x_{i,d})$. Update the particle's position: $x_i \leftarrow x_i + v_i$.

d) If ($f(x_i) < f(p_i)$) do:

Update the particle's best known position: $p_i \leftarrow x_i$.

3) If ($f(p_i) < f(g)$) update the swarm's best known position: $g \leftarrow p_i$

a) Now g holds the best found solution

b) Update the particle's position: $x_i \leftarrow x_i + v_i$

c) If ($f(x_i) < f(p_i)$) do:

Update the particle's best known position: $p_i \leftarrow x_i$.

4) If ($f(p_i) < f(g)$) update the swarm's best known position: $g \leftarrow p_i$.

5) Now g holds the best found solution.

4.Simulation Experiments

A data access optimization approach which uses predictive techniques for distributed computing environments. The main objective is to calculating CPU time dynamically i.e., to predicate performance in time series and to get a better performance for accessing a data optimizing application time.

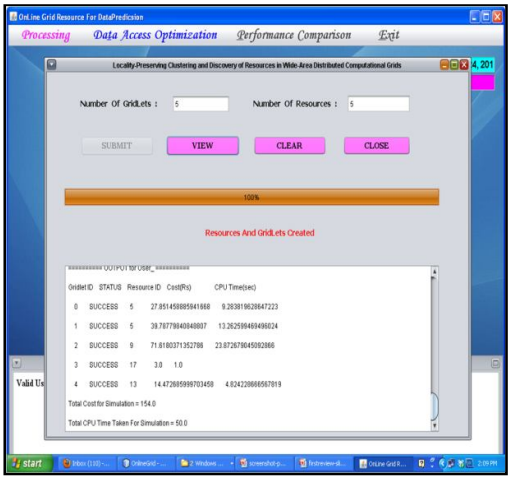

Figure 2: Resource Creation and CPU Time

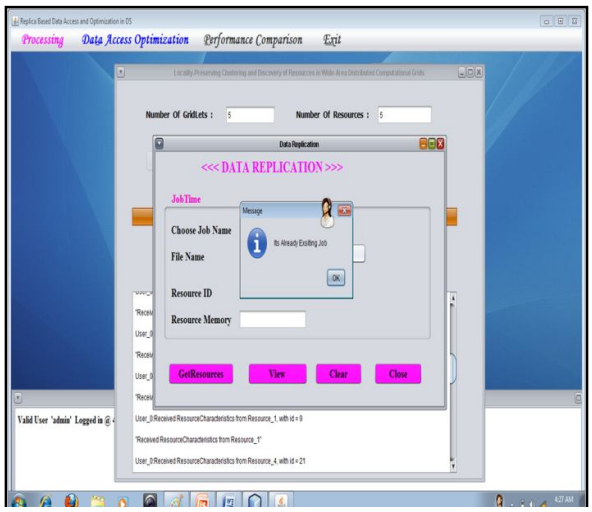

Figure 3: Resource Replica Schema

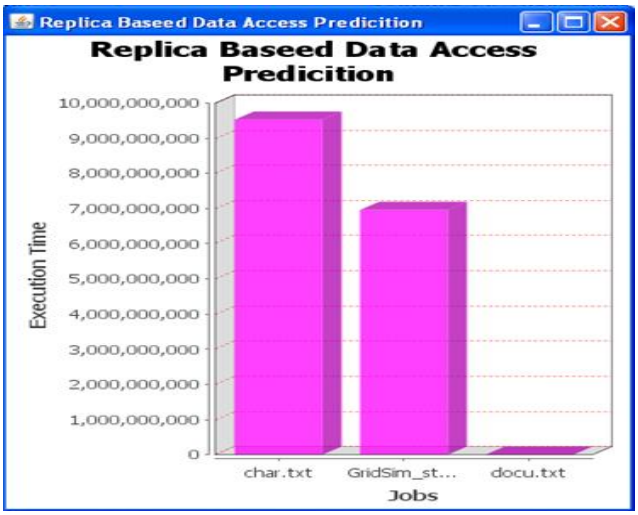

Figure 4:Replica Time Series (Predicate)

5.Future Direction

The data access operation finds to automatic and online prediction of read-and-write operations performed by processes. So, to extend the strategies model for application monitoring and prediction using PSO based Technique i.e., dynamically to predict CPU performances to accessing the resources in efficient way.

6.Conclusion

In this paper present that technique as replica based data access optimization, the objective is to minimize the application execution time by optimizing data accesses and therefore, improving decisions on replication, migration, and consistency. From that, data access operations are transformed into time series. By modeling those series, understand the behavior of applications. Therefore, to predicate time series and to get a better performance for accessing a data optimizing application.

7.Reference

1. Renato Porfirio Ishii and Rodrigo Fernandes de Mello "An Online Data Access Prediction and Optimization Approach for Distributed Systems" IEEE transactions on parallel and distributed systems, vol. 23, no. 6, june 2012.
2. H. Stockinger, "Defining the Grid: A Snapshot on the Current View," The J. Supercomputing,vol. 42, pp. 3-17, http://www.springerlink.com/content/906476116167673m, 2007.
3. T. Hey and A.E. Trefethen, "Cyber infrastructure for E-Science," Science, vol. 308 no. 5723, pp .817 - 821, http://www.sciencemag.org/cgi/content/abstract/308/5723/817, 2005.
4. J.P. Collins, "Sailing on an Ocean of 0s and 1s," Science, vol. 327, pp. 1455-1456, Mar. 2010.
5. G. Fox and D. Gannon, "Computational Grids," Computing in Science and Eng., vol. 3, no. 4, pp. 74-77, 2001.
6. J. Gray, "eScience: A Transformed Scientific Method," The Fourth Paradigm: Data-Intensive Scientific Discovery, T. Hey, S. Tansley, and K. Tolle, eds., Microsoft Research, 2009.
7. R. Ranjan, A. Harwood, and R. Buyya, "A Study on Peer-to-Peer Based Alhajj, "A Predictive Technique for Replica Selection in Grid Environment," Proc. IEEE Seventh Int'l Symp. Cluster Computing and Grid, pp. 163-170, May 2007.
8. M. Nielsen, "A Guide to the Day of Big Data," Nature, vol. 462, pp. 722-723, Dec. 2009.
9. R.P. Ishii and R.F. de Mello, "An Adaptive and Historical Approach to Optimize Data Access in Grid Computing Environments," INFOCOMP J. Computer Science, vol. 10, no. 2,pp. 26-43, http://www.dcc.ufla.br/infocomp/, 2011.
10. H.H.E. AL-Mistarihi and C.H. Yong, "On Fairness, Optimizing Replica Selection in Data Grids," IEEE Trans. Parallel Distributed Systems, vol. 20, no. 8, pp. 1102-1111, Aug. 2009.
11. M. Devarakonda and R. Iyer, "Predictability of Process Resource Usage: A Measurement-Based Study on Unix," IEEE Trans. Software Eng., vol. 15, no. 2, pp.1579-1586,http://dx.doi.org/10.1109/ Dec. 1989.

12. M. Faerman, A. Su, R. Wolski, and F. Berman, "Adaptive Performance Prediction for Distributed Data-intensive Applications," Proc. ACM/IEEE Conf. Supercomputing (Supercomputing '99), p. 36, 1999.
13. M. Wang, K. Au, A. Ailamaki, A. Brockwell, C. Faloutsos, and G.R. Ganger, "Storage Device Performance Prediction with Cart Models," Proc. IEEE CS 12th Ann. Int'l Symp. Modeling, Analysis, and Simulation of Computer and Telecomm. Systems (MASCOTS '04), pp. 588-595, 2004.
14. L. Senger, R.F. Mello, M.J. Santana, and R.H.C. Santana, "An On-Line Approach for Classifying and Extracting Application Behavior on Linux," High Performance Computing: Paradigm and Infrastructure, pp. 381-401, John Wiley and Sons Inc., 2005.
15. L.J. Senger, M. Santana, and R. Santana, "An Instance-based Learning Approach for Predicting Parallel Applications Execution Times," Proc. Third Int'l Information and Telecomm. Technologies Symp., pp. 9-15, Dec. 2005.