INTERNATIONAL JOURNAL OF
RECENT TRENDS IN MULTIDISCIPLINARY RESEARCH
September-October 2022, Vol 2(05), 01-03.
ISSN No: 2583-0368

Using distributed computing in weather prediction and analysis

Kshitij Singh
Delhi Skills and Entrepreneurship University, India.

Article Type: Research

OPENACCESS
Article Citation:
Kshitij Singh "Using distributed computing in weather prediction and analysis", International Journal of Recent Trends In Multidisciplinary Research, September-October 2022, Vol 2(05), 01-03.

Accepted date: October 02,2022

Published date : October 03,2022

Abstract: Distributed computer system is a paradigm used to denote a system of computers consisting of multiple software components in various machines, but running together as a single system. Weather forecasting is a complex phenomenon which requires computing capacity of very high configuration. Weather forecasting starts with an analysis of the current state of the atmosphere, ocean, and land surface. Observations which are reliable are drawn from many sources, including satellites, surface stations, aircraft, radar, weather balloons, etc. In the present work we explore various types of distributed computing configurations to suggest the best suited one for weather forecasting and analysis.
Key words: Weather forecasting; distributed computing; analysis; weather prediction.

1. Introduction

Distributed computing is a method of using the various components of computers distributed in different spatial locations to obtain information, process and execute it to obtain a suitable output.[10] [11] Distributed computing is a science which solves a large problem by giving small parts of that problem to many computers to solve and then combining the solutions for the parts into a solution for the problem. Distributed computing have been designed to use the computers of hundreds of thousands all over the world.[1] [2] The computers are connected by any form of communication network, whether local or wide-area in the coverage, or via the Internet. Using such a connected system of computers for weather prediction and analysis has been described as an approach in this paper.

2. Distributed computing

A computer is a complex system with calculations ranging from million calculations per second to zillion per second. Majority of processes of a computer can be categorized into three phases:

1. In Phase I, the master computer initially investigates the resource status of the slave computers, divides the job into a number of tasks and distributes the tasks onto these slaves according to the resources availability.
2. In Phase II: All the computers perform computations. Each computer executes processes allocated by the central computer while simultaneously executing the necessary local processes.
3. In Phase III: The master computer collects partial results from the slaves and computes the final result.

Distributed computing has emerged as a foreground essential for operating multi-tier computer systems in same or different geo-graphical locations. We first discuss about various models of distributed computing and then, recommend the more suitable ones to be used and being used for weather forecasting. The various types of distributed computing consists of:

A. Minicomputer model
B. Workstation model
C. Workstation-server model
D. Processor-pool model
E. Hybrid model

A comparative about these is described as follows

Minicomputer Model

A few minicomputers interconnected by a communication network.Several terminals are connected to each minicomputer so multiple users can simultaneously logged on to it.This model is an extension of the centralized time-sharing system. Drawbacks of Minicomputer Model:

1. Each user is logged on to one specific minicomputer, with remote access to other minicomputers.

2. The processing of tasks are done by the processors of a single minicomputer.
3. This model is very expensive.

Workstation Model

Several workstations (PCs) interconnected by a high-speed communication network (LAN).Each workstation is equipped with its own disk and serving as a single-user computer (the WS is called diskfull). The shortfalls in workstation model includes:

(i) A user logs onto one of the workstations called his or her "home" workstation.
(ii) Several issues must be resolved to distribute tasks on different WS:
(iii) How does the system find an idle workstation?
(iv) How a process can be transferred from one workstation to get it executed on another workstation?
(v) What happens to a remote process if a user logs onto a workstation that was idle until now and was being used to execute a process of another workstation. In order to solve these problems a few suggestions has been suggested [3] according to which:

1. To handle the first issue, the current master WS asks each machine for its current sate by sending a request status and each slave WS should reply by the current sate.
2. To handle the second issue, allocation/scheduling algorithm should be used to distribute tasks.
3. To handle the third issue, three commonly used approaches are used as follows:

Workstation-Server Model

A few minicomputers represent servers and several workstations (most of which are diskless) interconnected by a communication network. The workstation-server model has various advantages:

(i) It is much cheaperto use a few minicomputers equipped with large, fast disks that are accessed over the network than a large number of workstations, with each workstation having a small, slow disk.
(ii) Backup and hardware maintenance are easierto perform with a few large disks than with many small disks scattered all over a building or campus.[7]
(iii) Installing new releases of software (such as a file server with new functionalities) is easierwhen the software is to be installed on a few file server machines than on every workstation.
(iv) In a workstation-server model, users have the flexibility to use any workstation and access the files in the same manner irrespective of which workstation the user is currently logged on, because all files are managed by the file servers. [8]
(v) A user has guaranteed response time because workstations are not used for executing remote processes.

Processor-Pool Model

The pool of processors consists of a large number of microcomputers and minicomputers attached to the network.Each processor in the pool has its own memory to load and run a system program or an application program of the distributed computing system. [4] [5] Unlike the workstation-server model in which a processor is allocated to each user, in the processor pool model the processors are pooled together to be shared by the users as needed. The important advantages of processor pool model include:

(i) The entire processing power of the system is available for use by the currently logged-on users.
(ii) The system's services can be easily expanded without the need to install any more computers.

Hybrid Model

A hybrid model combines the advantages of both the workstation-server and processor-poolmodels. This model is based on the workstation-server model but with the addition of a pool of processors. [9] The processors in the pool can be allocated dynamically for computations that are too large for workstations or that requires several computers concurrently for efficient execution. Pros of hybrid model include:

(i) The hybrid model gives guaranteed response to interactive jobs by allowing them to be processed on local workstations of the users.
(ii) The hybrid model is more expensive to implement than the workstation-server model or the processor pool model.

3. Distributed Computing For Weather Analysis

On the basis of architecture of data and system management the following two types are described:

Grid type	Cloud type
Handle massive amounts of data. The full atmospheric model isgenerates large amounts of data. Through Grid tools like gridftp Allcock et al. (2002) we were able to efficiently transfer and store all simulation data.Access to HPC which suits parallel applications (e.g. Message Passing Interface; MPI). The model used in MeteoAG, as many other meteorological models, is a massive parallelapplication parallelized with MPI. On single	Costs can easily be determined and planned. Full control of software environment, including operating system (OS) with root access.This proved to be one of the biggest advantages for our workflows. It is easy to install neededsoftware, special libraries or modify any component of the system. Cloud providers usually offer most standard operating systems as

systems they run efficiently, however across different HPC clusters latencies become too high. A middleware can leverage the advantage ofaccess to multiple machines and run applications on suitable machines and appropriately distribute parts of workflows in parallel. Different hardware architectures. During tests in MeteoAG problems due todifferent hardware architectures were discovered (Schüller et al., 2007). . The exact cause was unclear, but most likely a combination of programming, the used libraries and setup down to the hardware level. - Difficult to setup and maintain as well as inflexible handling. Small hindrance make it usually negotiable for the Grid pattern to be followed for analysing complex atmospheric models.	images/AMI (Amazon Machine Image), but tuned images can also be saved permanently and made publicly available. Simple on-demand self-service. For applications with varying requirements for compute resources or with repeated but short needs for compute power, this is an important characteristic. As long as funds are available the required amount of compute power can be purchased. Ourworkflow was never forced to wait for instances to be available. Usually our standard on demand Linux instances were up and running within 5-10 seconds Slow data transfer and hardly any support for MPI computing. Data transfer to and from cloud instances is slow as well as a higher network latency between the instances.This limitation makes Cloud computingunsuitable for large-scale complex atmospheric models.

4.Conclusion

From the above detailed analysis we can safely conclude that a hybrid model of computing with grid type architecture will be best suited for analyzing and predicting large-scale complex atmospheric models. The system must check patterns in each record for issues such as spikes, flat liners, outliers, excessive ranges, and change points. Weather data are also checked for consistency across a region.

References

[1]. *Chang Liu, Rajiv Ranjan, Chi Yang, Xuyun Zhang, Lizhe Wang and Jinjun Chen (2015), "MuR-DPA: Top-Down Leveled Multi-Replica Merkle Hash Tree Based Secure Public Auditing for Dynamic Big Data Storage on Cloud", IEEE Transactions on Computers, Vol. 64, No. 9, pp. 2609-2622, Sept. 2015.*

[2]. *Chen B. and Curtmola R (2012), "Robust dynamic provable data possession", 2nd international conference IEEE on distributed computing systems workshops, pp. 515–525, 2012.*

[3]. *Chen L (2013), "Using algebraic signatures to check data possession in cloud storage", Future Gener Comput Syst, Vol. 29, No. 7, pp. 1709–1715, Sep. 2013.*

[4]. *Chun-I Fan, Jun-Cheng Chen, Shi-Yuan Huang, Jheng-Jia Huang and Wen-Tsuen Chen (2015), "Provably Secure Timed-Release Proxy Conditional Reencryption", IEEE Systems Journal, No. 99, pp. 1-12, 2015.*

[5]. *Clark G C and Cain J B (1981), "Error-correction coding for digital communications", New York: Perseus Publishing; 1981.*

[6]. *Cong W, Kui R, Wenjing L and Jin L (2010), "Toward publicly auditable secure cloud data storage services", IEEE Transactions on Networks, Vol. 24, pp. 19–24, 2010.*

[7]. *Deswarte. Y, Quisquater. J.-J and Saidane. A (2003), "Remote integrity checking: How to Trust Files Stored on Untrusted Servers", In Proc. of Conference on Integrity and Internal Control in Information Systems(IICIS'03), Belgium, Nov. 2003.*

[8]. *Duncan, Adrian, Sadie Creese and Michael Goldsmith (2012), "Insider attacks in cloud computing", Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on. IEEE, 2012.*

[9]. *Erdos, P. and Gallai, T. (1960), "Graphs with prescribed degrees of vertices"(Hungarian), Mat. Lapok, 11, pp. 264-274, 1960.*

[10]. *Erway C, Küpçü A, Papamanthou C and Tamassia R (2009), "Dynamic provable data possession", Proceedings of the 16th ACM conference on computer and communications security, USA, pp. 213–222, 2009.*

[11]. *Esiner E, Kachkeev A and Ozkasap O (2013), "Flex List: optimized skip list for secure cloud storage", Koç University, (Technical Report), 2013.*