Use of distributed computing in processing big data

Distributed Systems is an upcoming area in computer science and has the ability to have a large impact on the many aspects in[G1]the medical, scientific, financial and commercial sector. This document will provide an overview of distributed systems along with their current applications and application in big data.

The most commonly used definition for a distributed system is, a system comprised of geographically dispersed computing components interacting on a hardware or software level [1]. The rise in interest for distributed computing can be attributed to two major factors. The first factor is the creation and advancements in local and wide area networks which allow for large amounts of data to be transmitted over great distances in a short period of time [2].

The second factor is the new craze of the Internet of Things (IoT), where nearly every physical device manufacture having some sort of internet connectivity allowing for the possibility of tens of billions of devices that are able to interact. This large network of interconnected devices can be utilised to compute large amounts of data in a fraction of the time it would currently take to process.

Characteristics of a Distributed System

Heterogeneity

Heterogeneity refers to the ability for the system to operate on a variety of different hardware and software components. This is achieved through the implementation of middleware in the software layer. The goal of the middleware is to abstract and interpret the programming procedural calls

such that the distributed processing can be achieved on a variety of differing nodes [3].

Openness

The openness of a distributed system is defined as the difficulty involved to extend or improve an existing system. This characteristic allows us to reuse a distributed system for multiple functions or to process varying sets of data.

Concurrency

Concurrency refers to the system's ability to handle the access and use of shared recourses. This is important because if there is no measure implemented it is possible for data to get corrupted or lost by two nodes making different changes to the same resource such that the system can carry this error through different processes causing an incorrect result.

One way to counteract these errors is to implement a locking mechanism making a node unable to access a resource whilst it is being used by another node.[G2][G3]

Scalability

Scalability is one of the major characteristics that effectiveness of a distributed system, it refers to how easily the system can adapt to a changing size. This is due tothe volatile nature of computers, such that a device is prone to leaving and joining the system at will. This volatility is caused by computers powering down, or unstable networks causing connectivity issues.[G4][G5]

One factor that affects scalability is the degree at which the system is centralised. This is due to if a system relies on a centralised component or

process (e. g. a central[G6]server), the more nodes that try to communicate or use this component, the more likely it is that there will be a bottleneck at this point in the system.[G7]

Fault Tolerance

Due to a distributed system having many computers comprised of different aged hardware, it is very likely for a part to fail in such a way that a node can no longer operate. Fault Tolerance is the ability for the system tohandle such failures, this is achieved by using recovery and redundancy. Recovery is[G8]where a component will act in a predictable, controlled way if it relies on a component. Redundancy is where crucial systems and processes will have a backup that takes over if a system fails.[G9][G10]

Transparency

Transparency in a distributed system refers to the idea that the user perceives that they are interacting with a whole quantity rather than a collection of cooperating components. Transparency can be split into the following 8 sub-characteristics defined in Table 1.

Table 1

Different forms of transparency in a distributed system [2].

Transparency	Description
Access	Hide differences in data representation and how an object is accessed

Location	Hide where an object is located
Relocation	Hide that an object may be moved to another location while in use
Migration	Hide that an object may move to another location
Replication	Hide that an object is replicated
Concurrency	Hide that an object may be shared by several independent users
Failure	Hide the failure and recovery of an object

The Internet

The internet is the largest and most well-known decentralised distributed system ever created. It is currently comprised of millions of geographically distributed interconnected web servers that can communicate autonomously with each other and the billions of endpoint nodes [4].

The internet is constantly growing with more website and nodes added every day. One of the major factors contributing to the growth of nodes is the boost in IoT or smart devices.

ATM Machines

ATM machines are an example of a centralised distributed system that has been implemented globally. This is a centralised system because each ATM machine will[G11]only communicate with its bank central server.

Centralisation is enforced as a measure to increase the security of the sensitive information stored on the bank’s databases[G12].

Each bank’s ATM network has the ability to communicate with another banks server[G13]such that a user can withdraw money from any ATM around the world.

Botnets

Botnets are an example of a malicious distributed system. They are can either be operated by a central server or based off a peer-[G14]to-peer network. A botnet is comprised of a collection of zombie machines which have been infected with malware allowing the bot master to control it and a command and control server whose role is to control the zombie computers allowing the zombie machines to execute any command that the botmaster desires.

Data is any accumulation of facts and statistics to be analysed or referenced. Big data is most commonly defined as extremely large sets of data, both structured and unstructured,[G15]that can be analysed to reveal patterns and trends. This data is sufficiently complex or large enough that conventional data processing processes and applications are unable to deal with it [5].

Crowdsourcing is not a new idea in the software world, it is not an uncommon sight to see a developer pose a task to the masses and have someone else complete the task. This is mostly done free of charge.

A similar concept is starting to be applied to big data, where researchers and institutes have started to crowdsource data for people to process[G16]. Currently, most data that has been crowdsourced is[G17]for scientific or medical research. A factor that contributes to the success of data processing on distributed systems is the relatively low cost of[G18]transferring data compared the cost incurred from doing the data processing internally [6]. [G19]

Play to Cure: Genes in Space

Play to Cure: Genes in Space is a mobile gaming application developed by Cancer Research UK. Its main purpose is to allow the general public to process large amounts of data for the scientist at Cambridge University. [G20]The data is processed by the user controlling a spaceship to try and collect as much ‘ Element Alpha’ as possible. What the user is not aware that the placement of ‘ Element Alpha’ directly correlates to a singular piece of plotted data [7].

In the first month alone the application has managed to analyse 1. 5 million data sample. To process a similar number of samples the research team achieve a similar amount of samples processed, it would take the research team 125, 000 man hours [7].

Whilst it is a rudimentary implementation of a distributed system, Play to Cure: Genes in Space is a successful implementation and can show how important large distributed systems can processing big data.

SETI@home

SETI@home is currently the largest distributed computing program and was created by the SETI (Search for Extraterrestrial Intelligence) Institute and hosted out at UC Berkeley. It currently has approximately 3 million active users donating their computer's[G21][G22]spare processing power to process data obtained from SETI's radio telescopes [8].

Since SETI@home is a voluntary program, each node needs to be able to process data in a way that the user is not negatively affected and choose to leave the program. This is achieved through the application processing data when it is detected that a machine's CPU is[G23]idling [9].

As of the 10 March 2017, the[G24]SETI@home program has come close to processing 18 years' worth of data from the Arecibo Observatory radio telescope [10]. This achievement displays how easily large amounts of data can be processed by large distributed systems.

There are endless possibilities when it comes to the potential applications[G25]for distributed systems. Processing big data is a lucrative market, this might cause a lot of large multinational organisation to try and utilise their own hardware to implement their own personal distributed system to process the terabytes of data that they can extrapolate from their Enterprise resource planning (ERP) software and from data obtained from the media and other sources.

Stock trading is a cut throat industry, and the ability to predict market trends faster than a competitor can allow a particular firm to make millions of dollars. It is plausible for large firms to implement their own distributed

system to analyse previous market trends and current global and local affairs to predict the upcoming state of the market.

In the future, distributed systems will allow for big data to be processed potentially at a near real-time timeframe. This document has outlined how distributed systems can assist in the faster and more effective processing of big data.

References

[1]H. Karatza and G. Theodoropoulos, “ Distributed Systems Simulation”, *Simulation Modelling Practice and Theory* , vol. 14, no. 6, pp. 677-678, 2006.

[2]M. van Steen and A. Tanenbaum, “ A brief introduction to distributed systems”, *Computing* , vol. 98, no. 10, pp. 967-1009, 2016.

[3]G. Coulouris, J. Dollimore, T. Kindberg and G. Blair, *Distributed systems* , 1st ed. Harlow, England: Addison-Wesley, 2012, pp. 16-25.

[4]G. Coulouris, J. Dollimore, T. Kindberg and G. Blair, *Distributed systems* , 1st ed. Harlow, England: Addison-Wesley, 2012, pp. 8-9.

[5]P. Grover and R. Johari, “ BCD: BigData, cloud computing and distributed computing”, *2015 Global Conference on Communication Technologies (GCCT)* , 2015.

[6]J. Gray, “ Distributed Computing Economics”, *Queue* , vol. 6, no. 3, pp. 63-68, 2008.

[7]O. Childs, “ Download our revolutionary mobile game to help speed up cancer research”, *Cancer Research UK – Science blog* , 2017. [Online]. Available: http://scienceblog. cancerresearchuk. org/2014/02/04/download-our-revolutionary-mobile-game-to-help-speed-up-cancer-research/. [Accessed: 24- Mar- 2017].

[8]B. Marr, *Big Data: Using SMART Big Data; Analytics and Metrics To Make Better Decisions and Improve Performance* , 1st ed. Wiley, 2015, pp. 208-209.

[9]E. Korpela, D. Werthimer, D. Anderson, J. Cobb and M. Leboisky, “ SETI@home-massively distributed computing for SETI”, *Computing in Science & Engineering* , vol. 3, no. 1, pp. 78-83, 2001.

[10]” SETI@home”, *Setiathome. berkeley. edu* , 2017. [Online]. Available: https://setiathome. berkeley. edu/. [Accessed: 24- Mar- 2017].

[11]D. Anderson, J. Cobb, E. Korpela, M. Lebofsky and D. Werthimer, “ SETI@home: an experiment in public-resource computing”, *Communications of the ACM* , vol. 45, no. 11, pp. 56-61, 2002.

[12]S. Khan, “ The Curious Case of Distributed Systems and Continuous Computing”, *IT Professional* , vol. 18, no. 2, pp. 4-7, 2016.

[13]E. Albert, J. Correas, G. Puebla and G. Román-Díez, “ Quantified abstract configurations of distributed systems”, *Formal Aspects of Computing* , vol. 27, no. 4, pp. 665-699, 2014.

[14]S. Vinoski, “ Rediscovering Distributed Systems”, *IEEE Internet Computing* , vol. 18, no. 2, pp. 3-6, 2014.

[15]I. Foster, C. Kesselman, J. Nick and S. Tuecke, “ Grid services for distributed system integration”, *Computer* , vol. 35, no. 6, pp. 37-46, 2002.

[G1]Inserted: s

[G2]Inserted: is

[G3]Deleted: are

[G4]Inserted: by

[G5]Deleted: from

[G6]Inserted: a

[G7]Deleted: e

[G8]Inserted: by

[G9]Inserted: s

[G10]Deleted: through

[G11]Inserted: an

[G12]Inserted: ‘

[G13]Inserted: ‘

[G14]Inserted: f

[G15]Inserted: d

[G16]Inserted: r

[G17]Inserted: ,

[G18]Inserted: f

[G19]Deleted: t

[G20]Inserted: the

[G21]Inserted: ‘

[G22]Inserted: s

[G23]Inserted: ‘

[G24]Inserted: ,

[G25]Inserted: s

[G26]Inserted: ,