Agenda

4	7	CAP Theorem and implications for Big data Analytics	T1 Sec. 3.12 and 3.13; AR
	8	Big Data Lifecycle: Data Acquisition, Data Extraction –Validation and Cleaning, Data Loading, Data Transformation, Data Analysis and Visualization. Case study – Big data application	T1 Sec. 2.9 to 2.12; R1 Ch. 6 and Ch. 7

Contents

Terminologies Used in Big Data Environment

- In-Memory Analytics
- In-Database Processing
- Symmetric Multiprocessor System
- Massively Parallel Processing
- Difference between Parallel and Distributed Systems
- Shared Nothing Architecture
- Consistency, Availability, Partition Tolerance (CAP) Theorem Explained
- Basically Available Soft State Eventual Consistency (BASE)

- Few Top Analytics Tools

3.12 Terminologies Used in Big data Environments

In order to get a good handle on the big data environment, let us get familiar with a few key terminologies in this arena.

3.12.1 In-Memory Analytics

- Data access from non-volatile storage such as hard disk is a slow process.
- The more the data is required to be fetched from hard disk or secondary storage, the slower the process gets.
- One way to combat this challenge is to pre-process and store data (cubes, aggregate tables, query sets, etc.) so that the CPU has to fetch a small subset of records.
- But this requires thinking in advance as to what data will be required for analysis.
- If there is a need for different or more data, it is back to the initial process of pre-computing and storing data or fetching it from secondary storage.

This problem has been addressed using in-memory analytics.

- Here all the relevant data is stored in Random Access Memory (RAM) or primary storage thus eliminating the need to access the data from hard disk.
- The advantage is faster access, rapid deployment, better insights, and minimal IT involvement.

3.12.2 In-Database Processing

- In-database processing is also called as in-database analytics.
- It works by fusing data warehouses with analytical systems.
- Typically the data from various enterprise On Line Transaction Processing (OLTP) systems after cleaning up (de-duplication, scrubbing, etc.) through the process of ETL is stored in the Enterprise Data Warehouse (EDW) or data marts.
- The huge datasets are then exported to analytical programs for complex and extensive computations.
- With in-database processing, the database program itself can run the computations eliminating the need for export and thereby saving on time.
- Leading database vendors are offering this feature to large businesses.

3.12.3 Symmetric Multiprocessor System (SMP)

- In SMP, there is a single common main memory that is shared by two or more identical processors.
- The processors have full access to all I/O devices and are controlled by a single operating system instance.
- SMP are tightly coupled multiprocessor systems.
- Each processor has its own high-speed memory, called cache memory and are connected using a system bus.
- Refer Figure 3.9.

3.12.4 Massively Parallel Processing

- Massive Parallel Processing (MPP) refers to the coordinated processing of programs by a number of processors working parallel.
- The processors, each have their own operating systems and dedicated memory.
- They work on different parts of the same program. The MPP processors communicate using some sort of messaging interface.
- The MPP systems are more difficult to program as the application must be divided in such a way that all the executing segments can communicate with each other.
- MPP is different from Symmetrically Multiprocessing (SMP) in that SMP works with the processors sharing the same operating system and same memory.
- SMP is also referred to as tightly-coupled multiprocessing.

3.12.5 Difference Between Parallel and Distributed Systems

- The next two terms that we discuss are parallel and distributed systems.
- As is evident from Figure 3.10, a parallel database system is a tightly coupled system.
- The processors co-operate for query processing.
- The user is unaware of the parallelism since he/she has no access to a specific processor of the system.

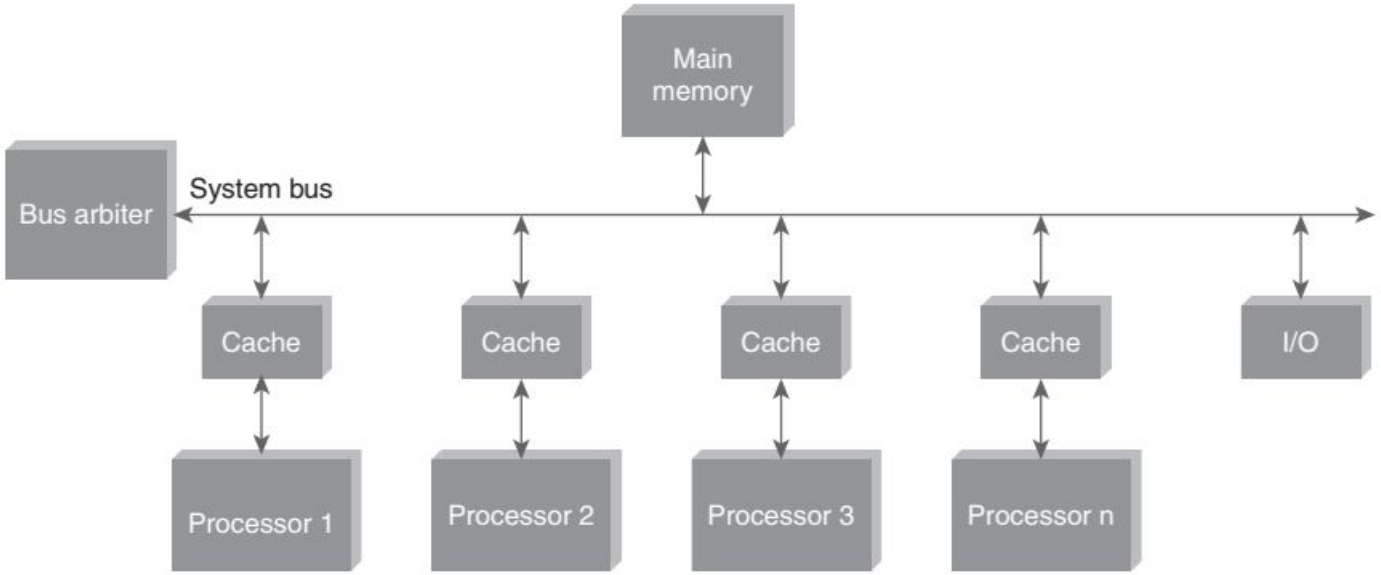

Figure 3.9 Symmetric Multiprocessor System.

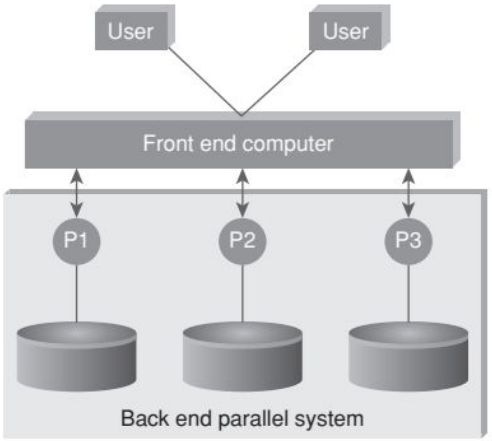

Figure 3.10 Parallel system.

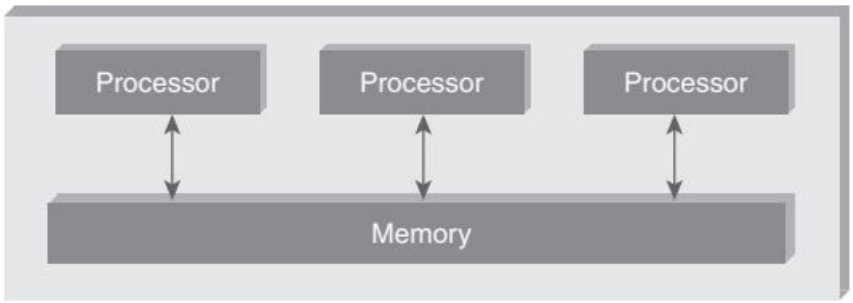

Figure 3.11 Parallel system.

- Either the processors have access to a common memory (Refer Fig 3.11) or make use of message passing for communication.
- Distributed database systems are known to be loosely coupled and are composed by individual machines.
- Refer Figure 3.12.
- Each of the machines can run their individual application and serve their own respective user.
- The data is usually distributed across several machines, thereby necessitating quite a number of machines to be accessed to answer a user query.
- Refer Figure 3.13.

3.12.6 Shared Nothing Architecture

Let us look at the three most common types of architecture for multiprocessor high transaction rate systems. They are:

1. Shared Memory (SM).
2. Shared Disk (SD).
3. Shared Nothing (SN).

In shared memory architecture, a common central memory is shared by multiple processors. In shared disk architecture, multiple processors share a common collection of disks while having their own private memory. In shared nothing architecture, neither memory nor disk is shared among multiple processors.

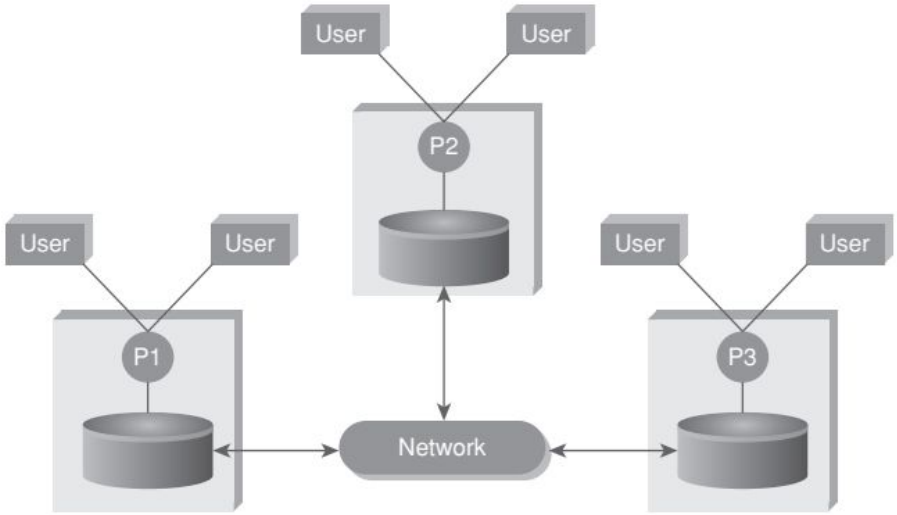

Figure 3.12 Distributed system.

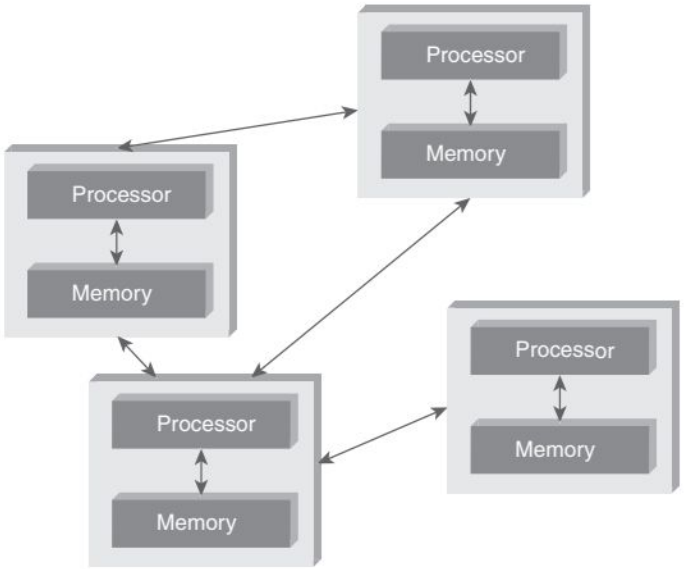

Figure 3.13 Distributed system.

3.12.6.1 Advantages of a "Shared Nothing Architecture"

1. **Fault Isolation:** A "Shared Nothing Architecture" provides the benefit of isolating fault. A fault in a single node is contained and confined to that node exclusively and exposed only through messages (or lack of it).
2. **Scalability:** Assume that the disk is a shared resource. It implies that the controller and the disk bandwidth are also shared. Synchronization will have to be implemented to maintain a consistent shared state. This would mean that different nodes will have to take turns to access the critical data. This imposes a limit on how many nodes can be added to the distributed shared disk system, thus compromising on scalability.

3.12.7 CAP Theorem Explained

The CAP theorem is also called the *Brewer's Theorem*. It states that in a distributed computing environment (a collection of interconnected nodes that share data), it is impossible to provide the following guarantees. Refer Figure 3.14. At best you can have two of the following three – one must be sacrificed.

1. Consistency
2. Availability
3. Partition tolerance

3.12.7.1 CAP Theorem

Let us spend some time understanding the earlier mentioned terms.

1. Consistency implies that every read fetches the last write.
2. Availability implies that reads and writes always succeed. In other words, each non-failing node will return a response in a reasonable amount of time.
3. Partition tolerance implies that the system will continue to function when network partition occurs.

Let us try to understand this using a real-life situation.

You work for a training institute, "XYZ." The institute has 50 instructors including you. All of you report to a training coordinator. At the end of the month, all the instructors together with the training coordinator peruse through the training requests received from the various corporate houses and prepare a training schedule for each instructor. These training schedules (one for each instructor) are shared with "Amey," the office administrator. Each morning, you either call the office helpdesk (essentially Amey's desk) or check in-person with Amey for your schedule for the day. In case a training request has been cancelled or updated (updates can be in the form of change in course, change in duration, change of the training timings, etc.), Amey is informed of the updates and the schedules are subsequently updated by him.

Things were good until now. Few corporate houses were your clients and the schedules of each instructor could be smoothly managed without any major hiccups. But your training institute has been implementing promotion campaigns to expand the business. As a result of advertising in the media and word of mouth publicity by your existing clients, you suddenly see an upsurge in training requests from existing and new clients. In consequence of that, more instructors have been recruited. Few trainers/consultants have also been roped in from other training institutes to help tackle the load.

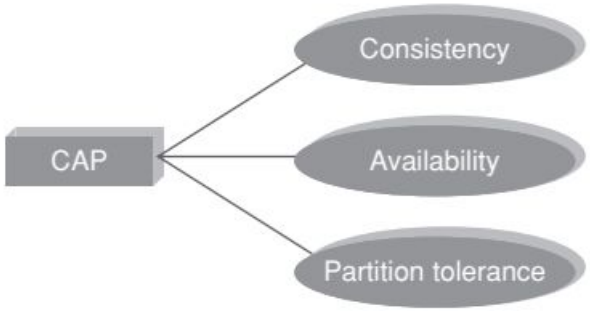

Figure 3.14 Brewer's CAP.

Now when you go to Amey to check your schedule or call in at the helpdesk, you are prepared for a wait in the queue. Looking at the current state of affairs, the training coordinator decides to recruit an additional office administrator "Joey." The helpdesk number will remain the same and will be shared by both the office administrators.

This arrangement works well for a couple of days. Then one day…

You: Hey Amey!

Amey: Hi! How can I help?

You: I think I am scheduled to anchor a training at 3:00 pm today. Can I please have the details?

Amey: Sure! Just a minute.

Amey browses through the file where he maintains the schedules. He does not see a training scheduled against your name at 3:00 pm today and responds back, "You do not have any training to conduct at 3:00 pm."

You: How is that possible? The training coordinator called up yesterday evening to inform of the same and said he has updated the office administrators of the same.

Amey: Oh! Did he say which office administrator? It could have been Joey. Please check with Joey.

Amey: Hey Joey! Please check the schedule for Paul here… Do you see something scheduled at 3:00 pm today?

Joey: Sure enough! He is anchoring the training for client "Z" today at 3:00 pm.

A clear case of inconsistent system!!! The updates in the schedule were shared by the training coordinator with Joey and you were checking for your schedule with Amey.

You share this incident with the training coordinator and that gets him thinking. The issue has to be addressed immediately otherwise it will be difficult to avoid a chaotic situation. He comes up with a plan and shares it with both the office administrators the following day.

Training Coordinator: Folks, each time that either an instructor or me calls any one of you to update a schedule, make sure that both of you update it in your respective files. This way the instructor will always get the most recent and consistent information irrespective of whom amongst the two of you he/she speaks to.

Joey: But that could mean a delay in answering either a phone call or sharing the schedule with the instructor waiting in queue.

Training Coordinator: Yes, I understand. But there is no way that we can give incorrect information.

Amey: There is this other problem as well. Suppose one of us is on leave on a particular day. That would mean that we cannot take any update related calls as we will not be able to simultaneously update both the files (my file and Joey's).

Training Coordinator: Well, good point! *That's the availability problem!!!* But I have thought about that as well. Here is the plan:

1. If one of you receives the update call (any updates to any schedule), ensure that you inform the other person if he is available.
2. In case the other person is not available, ensure that you inform him of all the updates to all schedules via email. It is a must!!!
3. When the other person resumes duty, the first thing he will do is update his file with all the updates to all schedules that he has received via email.

Wow!!! That is sure a Consistent and Available system!!!

Looks like everything is in control. Wait a minute! There is a tiff that has taken place between the office administrators. The two are pretty much available but are not talking to each other which, in other words, means that the updates are not flowing from one to the other. *We have to be partition tolerant!!!* As a training coordinator, you instruct them saying that none of you are taking any calls requesting for schedules or updates to schedules till you patch up. This implies that the system is partition tolerant but not available at that time.

In summary, one can at most decide to go with two of the three.

1. **Consistent:** The instructors or the training coordinator, once they have updated information with you, will always get the most updated information when they call subsequently.
2. **Availability:** The instructors or the training coordinators will always get the schedule if any or both of the office administrators have reported to work.
3. **Partition Tolerance:** Work will go on as usual even if there is communication loss between the office administrators owing to a spat or a tiff!

When to choose consistency over availability and vice-versa…

1. Choose availability over consistency when your business requirements allow some flexibility around when the data in the system synchronizes.
2. Choose consistency over availability when your business requirements demand atomic reads and writes.

Examples of databases that follow one of the possible three combinations:

1. Availability and Partition Tolerance (AP)
2. Consistency and Partition Tolerance (CP)
3. Consistency and Availability (CA)

Refer Figure 3.15 to get a glimpse of databases that adhere to two of the three characteristics of CAP theorem.

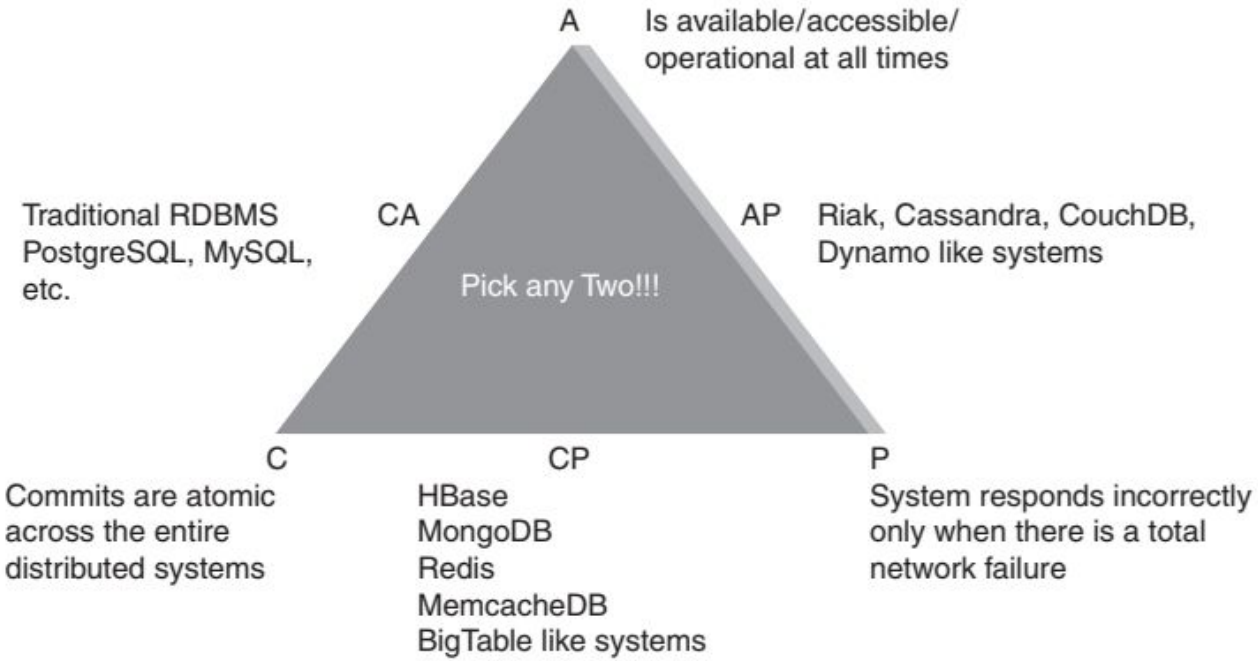

Figure 3.15 Databases and CAP.

3.13 BASICALLY AVAILABLE SOFT STATE EVENTUAL CONSISTENCY (BASE)

A few basic questions to start with:

1. ***Where is it used?***
   In distributed computing.

2. ***Why is it used?***
   To achieve high availability.

3. ***How is it achieved?***
   Assume a given data item. If no new updates are made to this given data item for a stipulated period of time, eventually all accesses to this data item will return the updated value. In other words, if no new updates are made to a given data item for a stipulated period of time, all updates that were made in the past and not applied to this given data item and the several replicas of it will percolate to this data item so that it stays as current/recent as is possible.

4. ***What is replica convergence?***
   A system that has achieved eventual consistency is said to have converged or achieved *replica convergence*.

5. ***Conflict resolution*: How is the conflict resolved?**
   (a) **Read repair:** If the read leads to discrepancy or inconsistency, a correction is initiated. It slows down the read operation.
   (b) **Write repair:** If the write leads to discrepancy or inconsistency, a correction is initiated. This will cause the write operation to slow down.
   (c) **Asynchronous repair:** Here, the correction is not part of a read or write operation.

3.14 FEW TOP ANALYTICS TOOLS

There is no dearth of analytical tools in the market. Please find below our list of few top analytics tools. We have also provided the links after each tool for you to explore more…

1. MS Excel
   https://support.office.microsoft.com/en-in/article/Whats-new-in-Excel-2013-1cbc42cd-bfaf-43d7-9031-5688ef1392fd?CorrelationId=1a2171cc-191f-47de-8a55-08a5f2e9c739&ui=en-US&rs=en-IN&ad=IN
2. SAS
   http://www.sas.com/en_us/home.html
3. IBM SPSS Modeler
   http://www-01.ibm.com/software/analytics/spss/products/modeler/
4. Statistica
   http://www.statsoft.com/
5. Salford systems (World Programming Systems)
   http://www.salford-systems.com/
6. WPS
   http://www.teamwpc.co.uk/products/wps

3.14.1 Open Source Analytics Tools

Let us look at a couple of open source analytics tools. We have also provided the links after each tool for you to explore more…

1. R analytics
   http://www.revolutionanalytics.com/
2. Weka
   http://www.cs.waikato.ac.nz/ml/weka/

- Quite a few data analytics and visualization tools are available in the market today from leading vendors such as IBM, Tableau, SAS, R Analytics, Statistica, World Programming Systems (WPS), etc. to help process and analyze your big data.
- Big data analytics is a about a tight handshake between three communities: IT, business users, and data scientists.
- *Data science* is the science of extracting knowledge from data.
- The CAP theorem is also called the Brewer's Theorem. It states that in a distributed computing environment (a collection of interconnected nodes that share data), it is impossible to provide the following guarantees. At best you can have two of the following three – one must be sacrificed.
  - Consistency
  - Availability
  - Partition tolerance

CONNECT ME (INTERNET RESOURCES)

- http://en.wikipedia.org/wiki/Data_science
- http://simplystatistics.org/2013/12/12/the-key-word-in-data-science-is-not-data-it-is-science/
- http://www.oralytics.com/2012/06/data-science-is-multidisciplinary.html
- http://spotfire.tibco.com/blog/?p=4240
- http://reports.informationweek.com/abstract/106/1255/Financial/tech-center-taking-advantage-of-in-memory-analytics.html
- http://www.informationweek.com/software/information-management/oracle-analytics-package-expands-in-database-processing-options/d/d-id/1102712?

IMP Note to Self