**CS286 Solving Big Data Problems – Exam #1 Study Guide**

**By: Zayd Hammoudeh**

**Lecture #01 – Introduction to Big Data**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Categories** | | **Data** – Raw values  **Information** – Set of data with meaning  **Knowledge** –Interpretation of the data with meaning.  **Wisdom** –Appropriate application of knowledge. |  |
| **Quantitative**   * Observable and **measureable** * Structured and **objective** * Numerical   **Example:** Income, Height | **Qualitative**   * Observable but not **measureable** * Unstructured and subjective * **Descriptive**   **Example:** Favorite Color |

**Storage Terminology**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Directly Attached**  **Storage (DAS)**   * Storage attached directly to the processing node. * Lowest capacity * Minimal data sharing * **Highest Speed.** | **Network Attached Storage (NAS)**   * Storage accessible via a network connection. * **Capable of using NFS** | **Relational Database  Management System (RDBMS)**   * Traditional database providers. * **Examples:** Oracle, MySQL, IBM DB2 | **Storage Area**  **Network (SAN)**   * Storage accessible via a network connection. * Uses different protocols than NAS. | **Network File System (NFS)**  Allows a computer to view and store data on remote disk as if that disk was directly attached to the local computer.  **Access Transparency** – Access data the same way whether it is remote or local. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Analysis Categories** | | **Four Steps in Traditional Data Mining**   1. Problem Definition 2. Data gathering and preparation 3. Model building and evaluation 4. Knowledge Deployment   Process is **cyclical** and **may repeat multiple times**. |  |
| **Descriptive**   * **Backward** looking. * Hindsight * Explain a previous phenomenon. * **Analysis** | **Predictive**   * **Forward** looking * Foresight * Investigate future trends. * **Mining** |

**Big Data**

|  |  |  |  |
| --- | --- | --- | --- |
| **Big Data** – Data whose scale, diversity, and complexity require new architecture, techniques, algorithms, and analytics to manage it and to extract value and hidden knowledge from it. | **3 V’s of Big Data** | | |
| **Volume** – The amount of data is too large for traditional database software tools to cope with.  **Example:** Image server | **Velocity** – The data is being produced at a rate that is beyond the performance limits of traditional systems.  **Example:** Social media site | **Variety** – Data lacks the structure to make it suitable for storage and analysis in traditional databases and data warehouses.  **Example:** Data organization variety. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data Organization** | | | **Scaling to Process Big Data** | | |
| **Structured** – Every piece of data and its format is known. Fits in a database.  **Example:** RDBMS | **Semi-structured** – For some fields, data may not exist and some fields can have different formats. Not in a typical database but has structure.  **Example:** XML, CSV, JSON | **Unstructured** – Does not fit into a database well. Most data is in this category.  **Examples:** Text document, multimedia content. | **Scale Up**  **Limitations:**   * Large capital and operating expense. * Lower availability and scalability.   **Example:** Monolithic Database | **Scale Out**  **Limitations:**   * Synchronization overhead * Programming Complexity * Specialized hardware.   **Example:** Grid Cluster | **Sampling**  **Limitation:**   * Lower accuracy and precision.   **Example:** Any approach |

|  |  |  |  |
| --- | --- | --- | --- |
| **Exploiting Locality of Reference** – In Big Data, accessing the data can be very time consuming. **Solution:** Keep the data and program close together.  **Distribute Data and Computation** – Map the data to multiple nodes and the program with it to decrease execution time. | **Three Laws of Big Data** | | |
| **Moore’s Law** – Every two years, the number of transistors per chip doubles.  **Kryder’s Law** – Every two years, storage capacity doubles. (**Storage version of Moore’s Law**) | **Amdahl’s Law** – The extent to which a program’s execution can be sped up is dependent on its level of parallelism. | **Murphy’s Law** – What can go wrong will go wrong.  **Big data must be resistant to failures.** |

**Hadoop**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Summary of the Hadoop Strategy** | | | **Core of Hadoop**   1. **Hadoop File System (HDFS)** 2. **Map Reduce** | **Name Node**  Key component in HDFS that **stores the location of distributed data in the file system**. | **Job Tracker**  Manages **computation tasks in the Hadoop system**. |
| **Distribute Data**  Processing nodes share no data. | **Distribute Computation**  Achieve **parallelism without synchronization.** | **Tolerate Failures**  Eliminate **single points of failure**. |

**Lecture #02 – Introduction to HDFS and MapR-FS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **File System**  Like a database. **A system to store data so that the data can be accessed later.**  **Typical Structure:** A rooted tree. | **Storage in a File System**  **Data** – Actual file in the FS.  **Metadata** – Information about the data/file.  **Example:** Size, location | **Block Structure in an ext2 File System** | | | | |
| **Typical Block Size:** 8KB | **Inode** – Data structure used to represent a file system object. This includes the location of the disk block location. | **Direct Block** – File block location **pointed to directly by the inode**. | **Indirect Block** – Block **pointed to by the inode through exactly one intermediary block**. | **Double Indirect Block** – Block **pointed to by the inode through exactly two intermediary blocks**. |

|  |  |
| --- | --- |
| **Virtual File System**   * **Transition layer** between a generic file system and a real file system. * Virtualizes different file system types into a single common interface. * **Enables standard POSIX** file access. * **HDFS is not compatible with a virtual file system while MapR-FS is.** | **Distributed File System**   * **Centrally stores metadata** (e.g. **name node**) **and distributes actual data** (e.g. **data node**) * Overcomes **space, performance, and availability limitations of a single machine.** * **Location Transparency** – Abstracts data locality from client access. |

**Hadoop File System (HDFS) Architecture**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **User Process**   * Connected to the HDFS through the network. * **Communicates with the name node to know where to read and write data**. | **Name Node**   * Manages file names and file locations on disk. * May have a secondary name node | **Network File System (NFS)**  Allows a computer to view and store data on remote disk as if that disk was directly attached to the local computer.  **Access Transparency** – Access data the same way whether it is remote or local. |