Unit 7 Notes

Technical Architecture

Overview

A **technical architecture** is a framework of rules, decisions, and structures for the overall design of a system. Data architecture, client-server architecture, networking architecture, and product architecture are some of the many types of architectures. Data architecture defines how data will move through the system. A **technical infrastructure** is a physical way of implementing a technical architecture. It is made of technologies, platforms, databases, and other components. Sometimes the terms architecture and infrastructure are used interchangeably.

Kimball's textbook presents the DW technical architecture and infrastructure together in chapters 4 and 5. The first chapter focuses on the application architecture and is required reading. The second chapter is optional. If you are a **technical architect** or a **technical support specialist** in your organization, you could consider both chapters as required reading for your job. Data architecture is presented in chapter 6, which we already covered in week 5.

A technical architecture for our data warehouse system helps the project manager plan out the project, serves as a great communication tool, and helps team members understand the big picture and their roles in it. We can say that having the architecture in place increases the likelihood of the project to succeed in meeting the business requirements. A top-level view of the DW/BI architecture is provided by Figure 4-1 (pp 114). We notice the data flow from **Source Systems** into the ETL system (also called **Back Room** of the data warehouse), followed by the **Presentation Server** (hosting the dimensional model) and then the BI Applications (also called the **Front Room**).

We also notice the **metadata** layer underlying the entire architecture. Metadata, or data about data, is the information that describes the structures, operations, and contents of the DW/BI system. Metadata is so important in this architecture, that we can call it metadata-driven. Ideally, all metadata should be stored in a single place shared by all DW/BI components. This would allow us to make a change only once, and make it available to the entire system. It would also facilitate the audit, compliance, quality management, and lineage. Unfortunately, very few tools are willing to share their metadata with others. One way of achieving integration is to buy all the DW/BI components from the same vendor. Some vendors also support the Common Warehouse Metamodel (CWM) standard, along with its storage facility called MetaObject Facility (MOF).

The DW/BI architecture **changes** over time. That is why many DW development techniques that we learned in the previous lectures are designed to be resilient to change. For example, the dimensional model's symmetry enables it respond in the same way to any request from the users. Storing fact data at the lowest possible granularity is also more resilient than storing them in an aggregated form.

Back Room Architecture

The back room architecture (Figure 4-2, pp 122) consists primarily of the ETL system, which we will present in detail in the next lecture. This system performs three major operations: (E) extracts the data from the data sources, (T) cleanses and conforms them, and (L) delivers them to the presentation server. Managing these operations can be considered a fourth operation. Most ETL services are available today as off-the-shelf tools, which are a must for any serious DW/BI project. However, ETL still takes roughly 70% of the total DW development time. It is particularly important that the ETL tools work with our key data sources – otherwise, we need to implement the extract code ourselves.

Common data sources include Enterprise Resource Planning (ERP) systems, Operational Data Store (ODS) systems, Master Data Management (MDM) systems, and XML files. ERP systems are made up of dozens of modules that cover major functional areas of business such as human resources, sales, and manufacturing. This data integration is not easy to leverage in a DW/BI project because we need to sort through thousands of tables, looking for the relevant ones. Also, these systems are not designed for analytical querying. Major ERP vendors offer complete DW/BI solutions to address these problems. ETL tool vendors also provide solutions for standard ERP systems.

ODS systems have a different definition in Kimball's DW/BI system than in Inmon's CIF. Kimball basically "phases out" this concept with the exception of the reporting ODS. This is a copy of the operational DB to support reporting on data less than 24h old. This ODS should be integrated into the DW/BI. MDM systems hold "the truth", or master copies, about key entities such as customers or products. They must solve data integration problems such as reconciling different synonym attributes (see our discussion forums) from different sources, thus doing a lot of work that would otherwise be done by ETL. Therefore, MDM are useful for the DW/BI system, particularly when implemented as SOA.

Presentation Server Architecture

Presentation servers (Figure 4-3, pp 135) store the data for direct querying by business users. Analytic queries are unpredictable in many aspects, thus the presentation server must be flexible and resilient to change. That is why it is based on the **atomic level business process dimensional models**, using the conformed dimensions from the enterprise bus matrix.

For large datasets, atomic data have to be accompanied by **aggregate** data. Aggregates are rebuilt on a daily basis, and they change over time according to their actual usage in queries. That is, aggregates that have not been used frequently enough will be dropped, and some new aggregates that have been used frequently will be added. This is called **usage-based optimization**, and it should be done by an **aggregate navigation system**.

Large datasets may also need to be partitioned onto multiple servers, either vertically or horizontally. Vertical partitioning breaks up the components of the presentation server onto separate servers – a server for the atomic data, another server for the aggregate data, and yet another server for ETL, for example. Horizontal partitioning distributes the load based on datasets rather than components.

Front Room Architecture

The front room (Figure 4-4, pp 142) is what most DW business users work with daily. It consists of several **BI application types** (pp 142) delivered to business users through a variety of **application interfaces** (pp 143), and **BI management services**, which are divided between shared services and user to desktop services. Shared services include metadata services, security services, usage monitoring, query management, enterprise reporting services, and web access. Many of these services are offered by vendors as fully-featured packages.

Technical Infrastructure

The DW/BI infrastructure includes the hardware, network, and low-level functions. Hardware architectures for parallel processing include **Symmetric Multiprocessing** (**SMP**), **Massive Parallel Processing** (**MPP**), and **Non**-**Uniform Memory Arhitecture** (**NUMA**), all illustrated in Figure 4-5 on pp 160. The SMP machine is a single machine with multiple processors, all managed by the same OS, and all sharing the same disk and memory. This architecture is well suited for ad hoc queries. A MPP system is a set of relatively independent machines, each with its own OS, memory, and disk. These machines communicate by passing messages back and forth. This architecture is good for storing atomic data and applying brute-force approach to answering canned queries. NUMA is similar to a cluster of SMP machines with more bandwidth and greater coordination among nodes. A cluster is a set of computers connected to act as a single server. A **cluster** can **scale out** (be expanded by adding more boxes) or **scale up** (be upgraded by getting a bigger machine).

Back room and presentation server infrastructure factors include data size, volatility, number of users, number of business processes, and nature of use. Front room infrastructure factors include application server considerations, desktop considerations, and connectivity and networking factors.

The various components of the technical architecture and infrastructure are largely determined by the business requirements. For instance, the architecture and infrastructure cost for a highly available application would be significantly different from an application that can accommodate some downtime. A system that is expected to be readily available to users 24x7 might need clustered database servers. Clustering them means utilizing additional servers, spending additional money towards resources, support, operational and maintenance costs. The need for high performance/response times also means utilizing high performance servers and/or dedicated server resources and/or multiple servers with load balancing solutions.