IJARIIT INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 6.078
(Volume 6, Issue 4)
Available online at: https://www.ijariit.com

Approach to help choose right data warehousing tool for an enterprise

Madhusudhan Reddy Sureddy
sureddy21@gmail.com
Santander Bank, Holmdel, NJ, USA

Prathyusha Yallamula
prathyusha13y@gmail.com
Natixis CIB Americas, New York, New York, USA

ABSTRACT

Data warehousing tool market space is currently filled with many tools each having its unique pros and cons making it very difficult for enterprises to make the choice of tool. The breadth of data warehousing has also moved beyond data integration, ETL, and presently encompasses data integration, data quality, master data management, metadata management, reference data management, big data Management. Enterprises expect the data warehouse tool providers to meet the requirements of all of these functionalities. This paper provides a point model approach using key performance indicators (KPI), Weightage and Scores to help choose the right data warehousing tool for enterprises.

***Keywords**: Business Intelligence (BI), Data Warehousing (DW), Approach, Choose, Applications, Big Data, Data, Tools, Enterprise, Organization, Data Integration, Data Quality, Master Data Management, Metadata Management, Reference Data Management, Big Data Management, Extraction Transformation Loading (ETL)*

1. INTRODUCTION

Data Warehouses (DW) integrate the data from operational systems and wide range of sources and stores it into a defined reportable structure. Data stored in the data warehouses is used by business intelligence systems to make enterprise growth decisions. Unlike operational systems data warehouses contain aggregate historical data. Commercial online applications or transactional systems came into play in 1960s after the advent of direct-access storage. Direct-access storage provided the means to store the data easily and helped in the introduction of database and database management systems. Initial databases built in 1960s were navigational databases which either followed hierarchy database model or the network database model for database management. 1970s introduced the relational databases, which utilized the E. F. Codd's principles and also programming languages like C and SQL. Easy availability of business data in the online transactional processing system databases paved the way for simpler business intelligence using historical data storage systems. During the period of 1970s to late 1980s various forms of historical data storage mart implementations provided the needed data for the business intelligence platform. Concept of data warehouse was formalized only in late 1980s to early 1990s with the ideas of Barry Devlin, Paul Murphy, Ralph Kimball and Bill Inmon. Bill inmon advocated the use of a top-down development approach where an atomic enterprise data warehouse containing data from all information systems in the enterprise was to be built following third normal form (3NF) principle. In this approach business specific data marts were to be built over the enterprise data warehouse. Ralph Kimball recommended a bottom-up approach where performance centric business specific data marts were to be built following dimensional modeling techniques. Both Bill and Ralph's methodologies coincided on the concepts of extraction transformation loading (ETL) process for doing data integration. Though the methodologies of Bill and Ralph were completely opposite they provided the tools to build successful business intelligence implementations.

Till the early 1990s enterprises predominantly used programming languages like C or Korn shell with SQL language to integrate the data from Online Transactional Processing (OLTP) sources and load data into the historical data storage marts or data warehouses. In this approach most of the data integration programming was done in the evolved programming languages of C or Korn shell, with the SQL language used only to access or load data. In the mid 1990s PL/SQL programming was introduced into the databases making enterprises to shift the loading approach by doing maximum data integration programming in PL/SQL, with a minimum usage of evolved programming languages. Both these approaches required developers to have good programming capabilities to build the enterprise native solutions. Few of the major disadvantages of these solutions were difficulties in managing operations and no metadata management capability. Introduction of the data warehouse ETL concepts gave birth to the idea of ETL tools, which provided a graphical user interface that could help in data warehouse data loading without writing a single line of programming language code. During run time the graphical user interface diagram prepared in the ETL tool was automatically converted into a programming language code by the ETL tool software. ETL tool approach for data warehouse loading eliminated the learning of the complicated programming languages and also provided the capability of metadata management and operational easiness.

Few of the ETL tools to enter the market space in mid 1990s were Informatica, Ab initio, DataStage. Products, which released during this period were mostly proprietary systems. Informatica's product suite contains multiple tools for maintaining data warehouses with a focus on data integration: extract, transform, load, information lifecycle management, business-to-business data exchange, cloud computing integration, complex event processing, data masking, data quality, data replication, data virtualization, master data management, ultra-messaging. Informatica has continued to grow through company acquisitions and sales growth. Ab initio Software specializes in high-volume data processing applications and enterprise application integration. Ab initio provides a wide range of applications in the data warehousing, customer relationship management, real-time analytics and enterprise application integration. DataStage originated at Vmark software and was completely acquired by IBM in mid 2000s and is currently branding itself as IBM Data stage. IBM DataStage has multiple features, which include data integration, metadata management, data quality etc. Informatica, Ab initio and IBM data stage still continue to be the leading players in data warehousing tools with a combined customer base of over 15000 companies. Trend of proprietary software model tools continued to the early 2000s with many more ETL tool players such as Microsoft SQL Server Integration Services, SAS data management etc. entering the market space.

Introduction of open source initiative in 1998 changed the way ETL products branded themselves and started the wave of commercial open source tools. Mid 2000s also saw the emergence of big data after the paper published by Google on map reduce and apache software foundation releasing the open source software apache Hadoop on how to manage massive amounts of data. Few of the commercial open source data warehousing ETL tools, which entered the market in 2000s, were CloverDX (previously CloverETL), Talend, Pentaho etc. 2010s saw the rise of cloud computing platforms with Amazon flagship product Amazon web services becoming a massive success and enterprises moving their infrastructure to various cloud platforms. Data warehousing ETL Tools like confluent, snaplogic etc. which entered the market space in 2010s to present continued the trend of being commercial open source and providing the features of cloud computing, big data, real-time, streaming-data processing. Many new tools continue to add up to the count on a yearly basis to the already crowded market space. Currently the data warehousing tool market space has innumerable number of tools available with around 50 plus key vendors. In this paper author provides an approach to help the enterprises to make this difficult choice of choosing a data-warehousing tool.

2. PROBLEM STATEMENT

Making a choice when there is only one option available is easy. But once the number of options starts growing, there is a strong possibility of choosing a wrong one with major flaws if a proper approach on choice making is not followed. Breadth of data warehousing has moved beyond data integration, ETL and presently encompasses data integration, data quality, master data management, metadata management, reference data management, big data Management. Enterprises expect data warehouse tool providers to meet the requirements of all of these functionalities. Data warehousing tool market space is currently filled with many tools each having their unique pros and cons making it very difficult for enterprises to make the choice. Enterprises are in need of an approach, which can help choose the right data-warehousing tool for their enterprise.

3. APPROACH TO HELP CHOOSE DATA WAREHOUSING TOOL

Approach published in this paper utilizes a point model using key performance indicators (KPI), Weightage and Scores for choosing the data-warehousing tool. Key performance indicators are the characteristics, which help in monitoring and measuring the tool effectiveness. In the detailed description of the KPI's, scoring definitions to rate the KPI's as Low – (0 to 3), Medium – (4 to 6), and High – (7-10) have also been provided. Decision on the KPI Weightage (01-10) and actual scoring (0-10) has to be done by the enterprises during their scoring exercise based on their long-term strategy. Scoring exercise needs to be conducted by enterprises during the planning phase of the data warehouse project. KPI which need to be considered when choosing the DW tool are provided below:

Business Model: This indicates the business model on which the tool was built upon – community-run open source (score – Low), Commercial open source (Score – High) and Proprietary (score – Low). Community run open source products involves zero cost as developers for community's gain rather than personal gain build the source code and the product. This software are released under a full usage license agreement where the copyright holders or vendors grants the users full rights of use, study, change and distribution of the software. Drawbacks of these tools are poor usability, lack of support. In the case of commercial open source vendors developing it work on a for-profit. This means though the open source code forms the foundation, product owners sell advanced features, improvements, guaranteed support, services, and upgrades to the organizations that use the tool. For proprietary software vendors do not provide a visibility on the source code. Major drawbacks of these business models are missing feature implementation on software takes long time; as the only option would be to wait for a newer version of software from product owners.

Platform Architecture: This indicates the efficiency of the holistic platform architecture for the entire data warehousing tool package. Tool package includes vendor suite of software's to achieve the core functionalities for data warehousing - data integration, data quality, master data management, metadata management, and reference data management. Type of architectures includes poorly integrated point to point solutions (Score – Low), comprehensive and united (Score – High). Architecture would be deemed "poorly integrated point to point solutions" when there are separate products built for each core functionality and the same product install does not work for all data warehousing products. This generally happens when vendors acquire products from multiple companies. In the case of "comprehensive and united" architecture there will be a single install for all the vendor products and the core functionalities would be just be additional features in the same product.

Number of Clients Tools: Organizations using windows laptops generally utilize "software center" to automatically push software's to the user machines. There are also strict requirements of testing in organizations from security stand point when new software's are introduced and also re-testing them is required during operating system upgrades. Increased number of client tools raises the effort needed to onboard the software and maintenance activities related to the software. More client tools also complicate data warehousing development in core functionalities - data integration, data quality, master data management, metadata management, and reference data management. More number of client tools is also an indicator of complex tool architecture. Scoring is inversely proportional to number of client tools - 2 client tools with 1 for development and 1 for administration get high, 3 client tools get medium and 4 or more gets low.

Installation Effort: Organizations utilize their own administrative teams for installing and maintaining the software's due to heavy premium charged by vendor software provider's professional services. Complexity in installation for the data warehousing tool package increases the efforts needed in on boarding and maintenance of the data warehouse tool software's. Efforts in upgrading of the software are also high if the installation effort is high. This KPI indicates the effort needed in installation. Increased number of days in installation also indicates complexity in the tool architecture. Scoring is inversely proportional to effort needed, less than 5 person days effort gets high, 5-10 person days gets medium and greater than 10 person days gets low.

Additional Software requirements: Data warehouse tool software's in many cases require additional software's like databases (oracle, SQL server etc.,), middle ware (jboss, tomcat, web sphere etc.) which are not included as part of their vendor provided data warehouse tool software package, but are essential for their functioning. This causes additional cost and also brings additional complexity in platform architecture and increases installation effort. Scoring reduces with increase in number of additional software needed - 0 gets low, 1 gets medium and 2 or more get high.

Project Initiation to Coding Start Time: Project initiation to coding start time attribute needs to be tracked separately as many projects have time sensitive requirements and a big overhead in this attribute would make organizations to go against choosing the software. This attribute is defined with a combination of client tool installation, server software installation, additional software installation, software licensing procurement and shipping time. Scoring should be reduced from high-low based on the increase in this number.

Ease of Development: Complexity in developing the job components in data warehousing tool software will increase the effort and in turn increase the cost needed in development of data warehouse. Friendliness in client tool navigation, feature complexity, code reusability features, number of job components need to be built are some of the factors which to be considered when rating the ease of development KPI. Increase in complexity in tool development lesser the scoring for the tool.

Tool Adoption Time for new developers: Organizations sometimes decide to use relatively new tools in the market due to the great feature availability or in some cases utilize full time resources with no skills on the tool instead of contractors due to cost reasons. In both these cases tool adoption time for new developers KPI plays a critical role in making a decision on the tool. More the time needed for tool adoption less the scoring for the tool.

Breadth of Data Integration Connectors: Connectors are critical in ensuring the data warehouse tool itself to be used for all of the organizational sourcing needs and ensuring no custom code to be written. Sources can be cloud applications like Sales force, work day, SAP, NetSuite etc., REST and SOAP web services access, Message queue access like JMS or MQ etc., business intelligence connectors like tableau etc. Enterprises should decide based on their long term strategy and the source systems in their enterprise to make the list of critical connectors needed and score accordingly. In case needed organizations can split this KPI into Data integration connectors for databases, applications and business intelligence based on their strategies to provide better visibility or increase the Weightage.

Breadth of Cloud Platform Connectors: Understanding the breadth of cloud connectors is critical for organizations with long term strategy of going from on premise infrastructure to cloud. Tool should provide connectors to all the popular cloud solutions of amazon web services (amazon red shift, amazon S3), Microsoft azure, Google cloud platforms, snowflake etc. Scoring should be made from low to high based on number of connectors available to the popular cloud platforms if organizations have not decided on the cloud platform and breadth of connectors in the chosen cloud platform if the decision has been made.

Semi-Structured and unstructured data handling features: Many organizations have source systems with semi structured data like XMLS, JSON and unstructured data like PDF, images etc. Though scope of complicated unstructured data is limited in data warehouses, organizations might consider data warehouse tool having components to easily read, parse and load this data as a key requirement for their long-term strategies. Scoring should be based on what components are considered critical by the enterprises and missing those should be rated from low to high.

Query ability on metadata: Query ability on metadata helps in building an understanding of the way data warehouse tool jobs work internally. As the developers only work on the client tools, which are mainly, drag and drop of the tool components, having this ability is critical. This query ability is very helpful in building testing frameworks during tool upgrades and also in building capacity planning of the data warehouse tool servers when combined with run statistics information. Unavailability of this feature in the tool gets a low score.

Software Configuration Management (SCM): Software configuration management is the ability to track and control changes in the software. It is critical for the data warehousing tools also to have this ability as multiple developers work on the same project at the same time and in some cases same code at the same time. Unavailability of SCM feature in the tool gets a low score.

Run Time Statistics Capture: Data warehouse tool job run time statistics include the job start, job end time, source success rows, source fail rows, transformation fail rows, target success rows, target fail rows, CPU usage, Memory usage, job process id, error messages. This information is critical in setting audit batch control model for the loading and also for building data quality in the data warehouse loading process. Unavailability of the run time statistics feature gets a low score.

Data lineage capability: Data lineage capability helps in providing an understanding of how attributes flow from the source system to the end report utilized by management for making decisions. Data lineage has become a requirement for compliance and it's critical to have this capability in the data warehouse tool package as most of the business logic happens in the data warehouse tool itself. Capability should be robust enough to load metadata of various databases and reporting tools also. Organizations again need to spend money on a new data lineage tool if this feature is unavailable or partial. Unavailability of this feature gets a low score, medium score for less robust feature availability and robust feature availability gets high score.

Breadth in data quality features: Common data quality features are data profiling, data standardization, and data matching components in the tool. Advanced data quality features which provide breadth include address validation using USPS addresses, automated data exception handling, reference table management, score carding of data profiling results, data virtualization. Scoring should be based on what components are considered critical and missing them should be rated from low to high.

ELT Processing: Organizations sometimes prefer pushing the entire transformation processing to the data warehouse database itself rather than executing on the ETL server. This is preferred when MPP databases like Teradata, Netezza or fast processing databases like Exadata are used as the data warehouse database. Availability of this feature with an ability of every component in the tool to be converted into a database code gets the highest score. Missing tool components for conversion get a medium score and no ELT processing feature availability gets a low score.

Job Parallelization: Data warehouses have strict service level agreements (SLA) for data availability and hence it's critical that data warehousing tools provide an option to optimize tool long running jobs using features of job parallelization. Job parallelization helps in automatic splitting of the same job into multiple parallel processes or threads and reducing the time of execution by the number of parallel splits. High score is provided if the feature exists and the granularity of parallelization is at each component level, medium score for availability of parallelization option and low score for no availability of this feature.

Architecture for Big Data Integration: As the name suggests big data works on huge amount of data and hence it's critical to understand the efficiency of the big data integration architecture. Tools generally convert the tool components to map reduce or spark or flink or HIVE map reduce or HIVE Tez or native big data variants. Due to the performance number availability in the market, open source variants are preferred over the native tool solutions. High score is provided to the tools allowing conversion to spark or flink and medium score to map reduce or HIVE map reduce or Hive Tez and low score to native big data variants.

Breadth of Big Data Connectors: Big data connectors include the components and connectors related to big data. These include Apache Hadoop components - HDFS, HBASE, Hive, Pig, Flume, NoSQL, In-memory databases - Cassandra, Couch DB, MongoDB, Amazon DynamoDB, Couchbase, Influx Data, DataStax, Redis, Maria DB, Nuo DB, Greenplum etc, Graph databases. Scoring should be made from low to high based on number of connectors available if organizations have not decided on the big data solutions and breadth of connectors in the chosen solution if the decision has been made.

Real Time Big Data Connectors: Real big data connectors are listed as a separate KPI as organizations sometimes have specific big data real time requirements in addition to batch. Tools should provide options to all the streaming variants of Spark Streaming, Kafka, Flink, Storm, flume. Scoring should be made from low to high based on number of connectors available if organizations have not decided on the big data solution and breadth of connectors in the chosen solution if the decision has been made.

Total cost of ownership: Total cost of ownership includes the per year cost for the tool license, connectors license (if sold separately), additional software license and the vendor support. Cost is obviously the most important KPI for consideration for organizations and the Weightage and scoring is hence based on the organization's discretion.

Resource Availability in Market: Enterprises tend to hire contracting resources for building their data warehousing projects and hence it's critical to have resources with the data warehouse tool skill sets to be available in the market. Enterprises make the choice of contractors against the full-time resources (FTE) due to the skill set gaps between them. Even with additional time allocation in the project for training FTE's, final deliverable out of a newly trained resource would be subpar from an experienced resource. Tools, which have newly entered the market or not explored by multiple clients tend to have fewer resource availability in the market and get a low score. Tools, which have been around for a long time and have a huge client base get a high score.

Documentation: Good documentation helps in quickly building the skill set of the enterprises FTE's and is an indirect contributor to cost reduction to data warehouse projects. Scoring is based on the depth of documentation with "freely available web documentation with search capability and real time examples" getting high score; "freely available web documentation with search capability" getting medium score and anything lower getting low scores.

Community Support: Enterprises though initially starting with contractors tend to switch towards FTE for cost reduction and knowledge retaining reasons. Inexperienced trained resources require great community support availability more than tool documentation to quickly build using the data-warehousing tool. Community support has multiple advantages, which include solution for unique problems, product related issue resolutions and in many cases simpler way of explaining data warehousing development than the tool documentation. Tools with freely available documentation, community network of their own and longer availability in the market get a high score and the tool which miss one or more those features get medium to low score.

Below table provides the template for the choosing exercise which needs to be performed by organizations to choose the data warehousing tool. As observed in the template, points for a KPI is achieved by multiplying weightage of KPI with the score of the KPI and the performance of the tool is the sum of points for all the KPIs. Based on that in the below table Tool <n> emerges as the winner.

Table 1: Data warehouse tool choosing exercise template

Key performance indicator (KPI)	Weightage (1-10)	Tool <1> Score (1-10)	Tool <1> Points (Weightage*Score)	Tool <n> Score (1-10)	Tool <n> Point (Weightage*score)
KPI <1>	2	6	12	7	14
KPI <..>	..	..	..	..	..
KPI <n>	4	5	20	6	24
Performance			32		38

Enterprises can also group certain KPIs and calculate the performance for that group alone to ensure their core KPIs also get the same points while comparing tools. During this exercise enterprises might decide they might need multiple tools to achieve all their requirements or might even decide to build a native tool of their own.

4. CONCLUSION

When choosing a data warehousing tool for an enterprise there is no one size fits all option, due to each enterprise's unique requirements and their long term strategies. Every enterprise during the data warehousing initiation needs to go through the tool choosing exercise to pick the correct tool. Enterprises in some unique cases might need to choose multiple tools in case a single tool does not satisfy all of their needs. In this paper author provides an approach utilizing key performance indicators to help choose right data warehousing tool for enterprises.

5. REFERENCES

[1] Madhusudhan Reddy Sureddy, Prathyusha Yallamula (2020), DATA QUALITY ARCHITECTURE FOR DATA WAREHOUSES, International Journal of Research Cultural Society , 4(6), pp 95-100
[2] Madhusudhan Reddy Sureddy, Prathyusha Yallamula (2020), A Framework for Monitoring Data Warehousing Applications, International Research Journal of Engineering and Technology, 7(6), pp 7023-7029
[3] Madhusudhan Reddy Sureddy, Prathyusha Yallamula (2020), DEBUGGING METHODOLOGY FOR ISSUES IN DATA WAREHOUSING AND BUSINESS INTELLIGENCE PLATFORMS, Proceedings of IFERP International conference, Luxor, Egypt, August 10-11, 2020
[4] https://en.wikipedia.org/wiki/History_of_software
[5] https://www.quickbase.com/articles/timeline-of-database-history
[6] https://blog.syncsort.com/2017/08/mainframe/mainframe-history/
[7] https://www.betterbuys.com/bi/history-of-business-intelligence/
[8] https://en.wikipedia.org/wiki/Data_warehouse#History
[9] https://en.wikipedia.org/wiki/Database#1960s,_navigational_DBMS
[10] https://dbanotes.com/the-history-of-pl-sql-f8b7a6c5eff8
[11] https://en.wikipedia.org/wiki/History_of_programming_languages#Early_history
[12] https://en.wikipedia.org/wiki/SQL_Server_Integration_Services
[13] https://en.wikipedia.org/wiki/Informatica
[14] https://www.informatica.com/about-us.html
[15] https://en.wikipedia.org/wiki/Ab_Initio_Software
[16] https://www.alooma.com/blog/etl-tools-comparison
[17] https://solutionsreview.com/data-integration/top-free-and-open-source-etl-tools-for-data-integration
[18] https://verify.wiki/wiki/Talend (Big_Data)
[19] https://www.limswiki.org/index.php/CloverETL
[20] https://en.wikipedia.org/wiki/Big_data
[21] https://en.wikipedia.org/wiki/Open-source_software
[22] https://en.wikipedia.org/wiki/The_Apache_Software_Foundation
[23] https://en.wikipedia.org/wiki/Apache_Hadoop
[24] https://drawio-app.com/commercial-open-source/
[25] https://www.dataversity.net/brief-history-data-warehouse/