DataStage

Usama Tahir
JUNIOR CONSULTANT | DATA ANALYTICS

Contents

Description

These are the following points I try cover in this document

- The basic overview of an ETL process.
- Stages of ETL process
- Overview of IBM DataStage
- Data stage main functions
- Jobs in data stage

- Parallelism in data stage
- File stages in data stage
- Processing stages in data stage
- Database stages in data stage
- Best practices in data stage

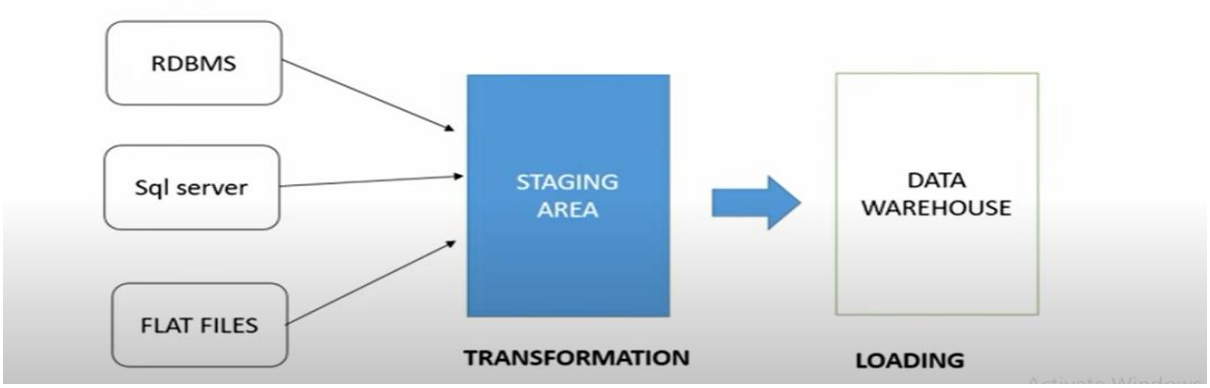
What is ETL

In computing, extract, transform, load is the general procedure of copying data from one or more sources into a destination system which represents the data differently from the source or in a different context than the source.

In data warehousing it is a process of pulling data out of source system and putting it in data warehouse

It consists of three steps

1) Extraction of data
2) Transformation of data
3) Loading of data



Extraction of data

- Data can be extracted from different sources
- Sources can be relational databases and files (JSON, XML)
- The main objective of this step is to retrieve all the data from source systems
- The extraction step should have designed in such a way that it should not have negative effect on source systems

Transformation of data

- This step include cleaning, filtering, validating and applying rule to extracted data
- The main objective is to load the extracted data into target database with clean and general format
- Each source system has its own format. For example, there are two sources A and B. Date format in A is dd/mm/yyyy and date format in B is yyyy/mm/dd. In transformation state their data is bring into a general format.

Loading of data

- Data extracted and transformed is of no use until its loaded in a target database
- In this step extracted and transformed data is loaded in target database
- In order to load data efficiently it is necessary to index database

ETL process can be run in parallel. As data extraction take time so the second step of transformation take place simultaneously. This prepare the data for third step of loading as soon as some data is ready it is loaded with out the completion of pervious steps.

What is DataStage

DataStage is an ETL tool used to extract, transform, and load data from the source to the target destination. The source of these data might include

- sequential files
- indexed files
- relational databases
- external data sources
- archives
- enterprise applications

DataStage is used to facilitate business analysis by providing quality data to help in gaining business intelligence. DataStage ETL tool is used in a large organization as an interface between different systems. It takes care of extraction, translation, and loading of data from source to the target destination.

DataStage Overview

DataStage has following Capabilities.

- It can integrate data from the widest range of enterprise and external data sources
- Implements data validation rules
- It is useful in processing and transforming large amounts of data
- It uses scalable parallel processing approach
- It can handle complex transformations and manage multiple integration processes
- Leverage direct connectivity to enterprise applications as sources or targets
- Leverage metadata for analysis and maintenance
- Operates in batch, real time, or as a Web service

DataStage main functions

DataStage performs data transformation and movement from source systems to target systems in batch and in real time. The data sources might include indexed files, sequential files, relational databases, archives, external data sources, enterprise applications, and message queues.

DataStage manages data that arrives and data that is received on a periodic or scheduled basis. It enables companies to solve large-scale business problems with high-performance processing of massive data volumes. By leveraging the parallel processing capabilities of multiprocessor hardware platforms, DataStage can scale to satisfy the demands of ever-growing data volumes, stringent real-time requirements, and ever-shrinking batch windows.

DataStage delivers four core capabilities:

- Connectivity to a wide range of mainframe, legacy, and enterprise applications, databases, file formats, and external information sources.
- Prebuilt library of more than 300 functions including data validation rules and very complex transformations.
- Maximum throughput using a parallel, high-performance processing architecture.
- Enterprise-class capabilities for development, deployment, maintenance, and high-availability. It leverages metadata for analysis and maintenance. It also operates in batch, real time, or as a Web service.

In the following sections, we will see the following aspects of IBM DataStage:

- Data transformation
- Jobs
- Parallel processing

Data transformation

Data transformation Data transformation and movement is the process by which source data is selected, converted, and mapped to the format required by targeted systems. The process manipulates data to bring it into compliance with business, domain, and integrity rules and with other data in the target environment. Transformation can take some of the following forms:

Aggregation

Consolidating or summarizing data values into a single value. Collecting daily sales data to be aggregated to the weekly level is a common example of aggregation.

Basic conversion

Ensuring that data types are correctly converted and mapped from source to target columns.

Cleansing

Resolving inconsistencies and fixing the anomalies in source data.

Derivation

Transforming data from multiple sources by using a complex business rule or algorithm.

Enrichment

Combining data from internal or external sources to provide additional meaning to the data.

Normalizing

Reducing the amount of redundant and potentially duplicated data.

Combining

The process of combining data from multiple sources via parallel Lookup, Join, or Merge operations.

Pivoting

Converting records in an input stream to many records in the appropriate table in the data warehouse or data mart.

Sorting

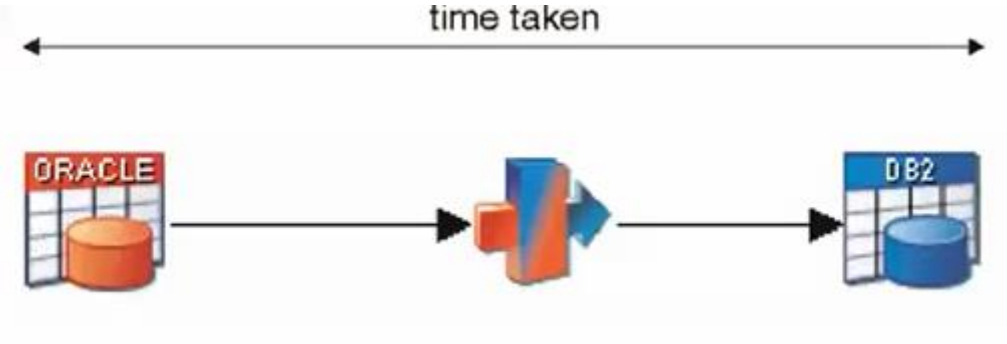
Grouping related records and sequencing data based on data or string values.

Jobs

DataStage job consists of individual stages linked together which describe the flow of data from a data source to a data target. A stage usually has at least one data input and/or one data output. However, some stages can accept more than one data input, and output to more than one stage. Each stage has a set of predefined and editable properties that tell it how to perform or process data. Properties might include the file name for the Sequential File stage, the columns to sort, the transformations to perform, and the database table name for the DB2 stage. These properties are viewed or edited using stage editors. Stages are added to a job and linked together using the Designer. Figure 1-7 shows some of the stages and their iconic representations.

Parallelism in DataStage

This job below consists of three stages. This is not a parallel job It works in a way that the source stage would read all the source records once it completed reading all the records then it transfer all the records to second stage which is transformer stage when transformer stage has completed all its transformations on data then data is passed to the third stage which is target stage.



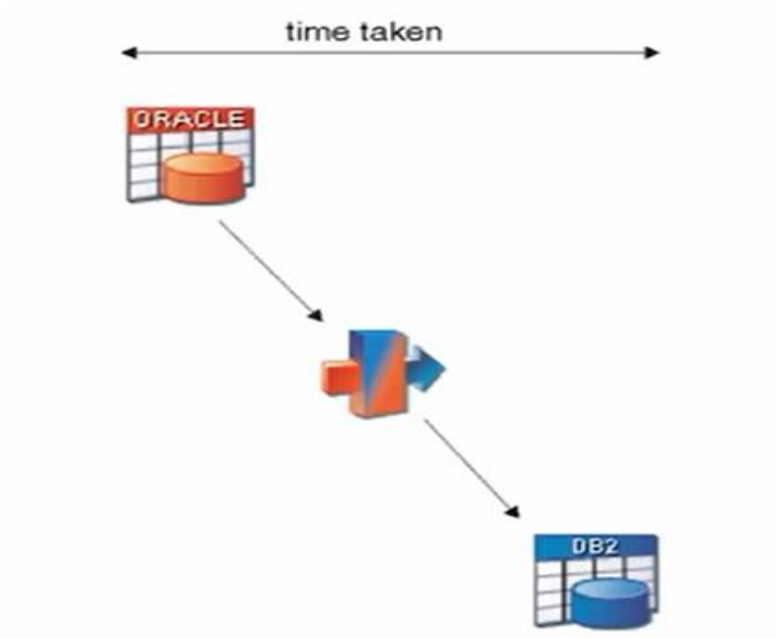
In a parallel job, each stage would normally (but not always) correspond to a process. You can have multiple instances of each process to run on the available processors in your system.

A parallel DataStage job incorporates two basic types of parallel processing.

- Pipeline parallelism
- Partition parallelism

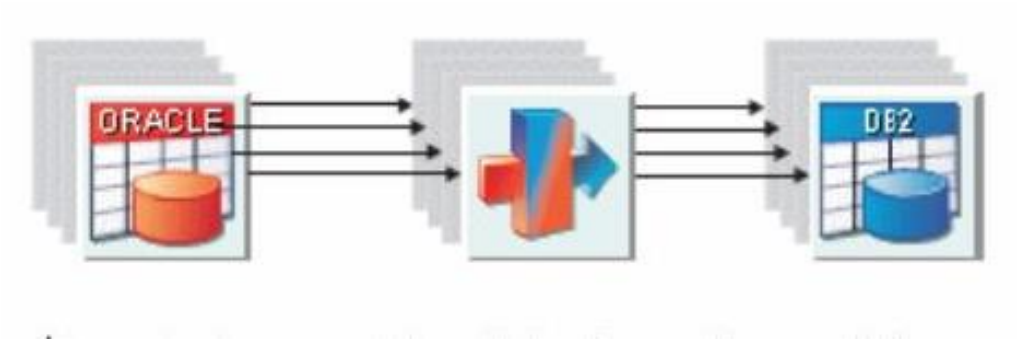
Pipeline parallelism

All stages run concurrently, even in a single-node configuration. As data is read from the Oracle source, it is passed to the Transformer stage for transformation, where it is then passed to the DB2 target. Instead of waiting for all source data to be read, as soon as the source data stream starts to produce rows, these are passed to the subsequent stages. This method is called pipeline parallelism, and all three stages in our example operate simultaneously regardless of the degree of parallelism of the configuration file. The Information Server Engine always executes jobs with pipeline parallelism. If you ran the example job on a system with multiple processors, the stage reading would start on one processor and start filling a pipeline with the data it had read. The transformer stage would start running as soon as there was data in the pipeline, process it and start filling another pipeline. The stage writing the transformed data to the target database would similarly start writing as soon as there was data available Thus all three stages are operating simultaneously.



Partition parallelism

When large volumes of data are involved, you can use the power of parallel processing to your best advantage by partitioning the data into a number of separate sets, with each partition being handled by a separate instance of the job stages. Partition parallelism is accomplished at runtime, instead of a manual process that would be required by traditional systems. The DataStage developer only needs to specify the algorithm to partition the data, not the degree of parallelism or where the job will execute. Using partition parallelism, the same job would effectively be run simultaneously by several processors, each handling a separate subset of the total data. At the end of the job the data partitions can be collected back together again and written to a single data source.



Types of Partitioning

Partitioning can be of two types

- Keyless partitioning
- Key based partitioning

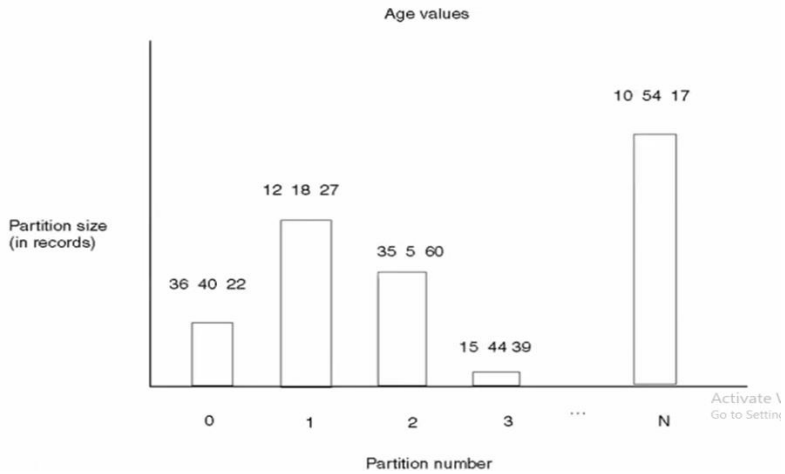
Key based partitioning

All those type of operations in which we need to define a key are called keyed operations like joining the data based on some key or lookup data. So, all those kind of operations in which we need to define a key are used key based partitioning

Examples are hash, range, modulus, DB2

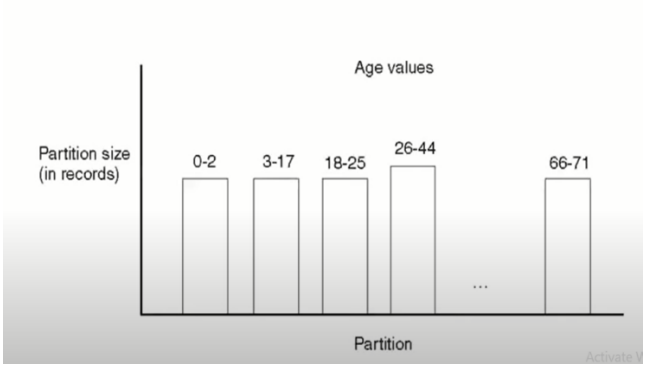
Hash Partitioning

One of the most commonly used partitioning method which is key based in data stage parallel jobs whenever you have a key base operation weather it's a join stage weather it's a sort stage weather you used derivations based on some key columns.



Range Partitioning

Type of key based partitioning It is mostly used in case of age values because mostly age in put into range



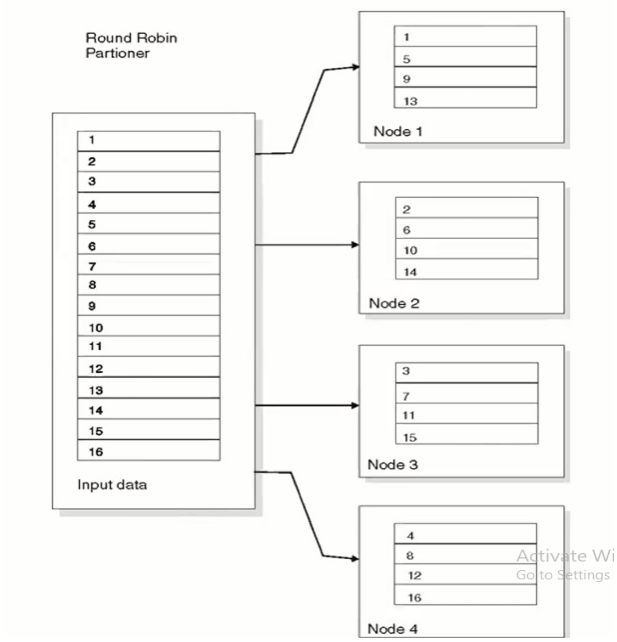
Key less partitioning

All those cases in which we don't require any key to partition data we use key less partitioning

Examples are round robin, random, entire, same

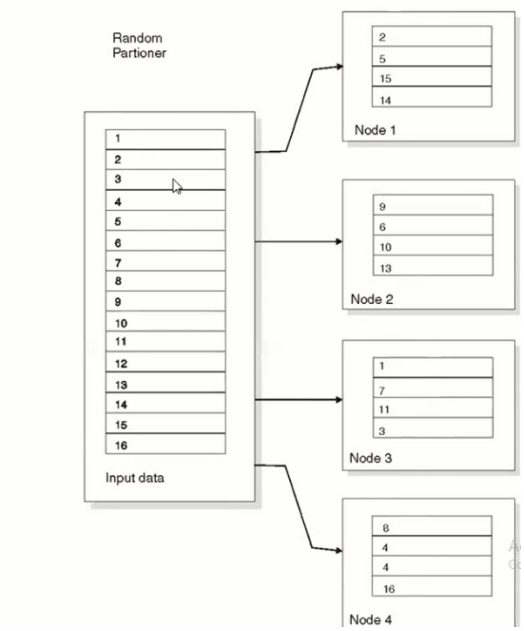
Round Robin Partitioning:

Let's say we have 16 records and 4 nodes then 1st record will go to the first partition second record goes to second partition third record goes to third partition and 4th goes to fourth partition 5th record goes again to node and so on



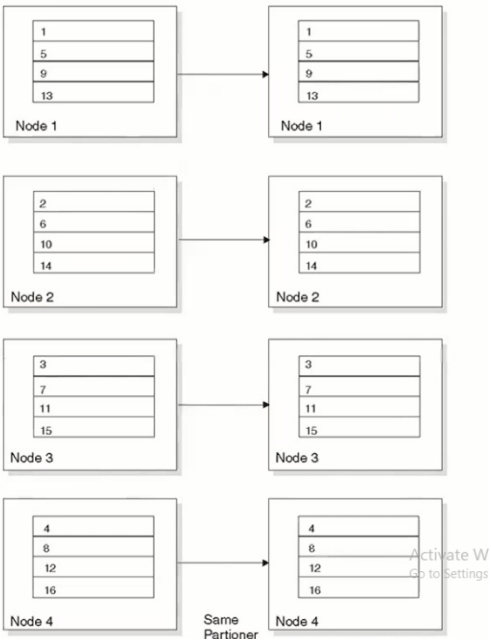
Random Partitioning:

Let's say we have 16 records and 4 nodes then in random partitioning all the records will be assigned to different partitions randomly



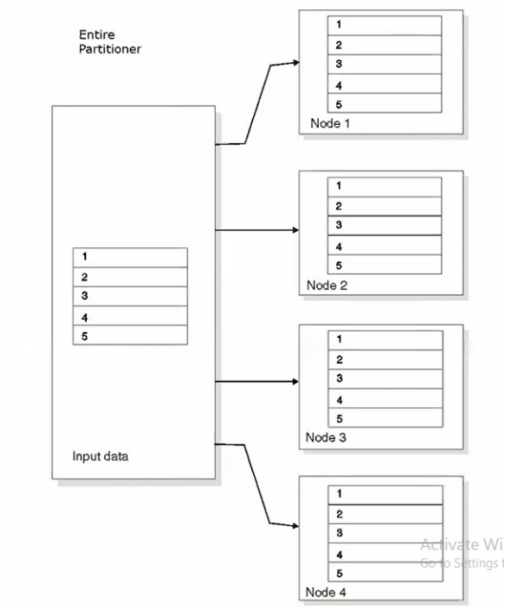
Same Partitioning:

This partitioning is used for intermediate stages It basically keep the partitioning same as the pervious stage has. So, it is used we do need to repartition the data



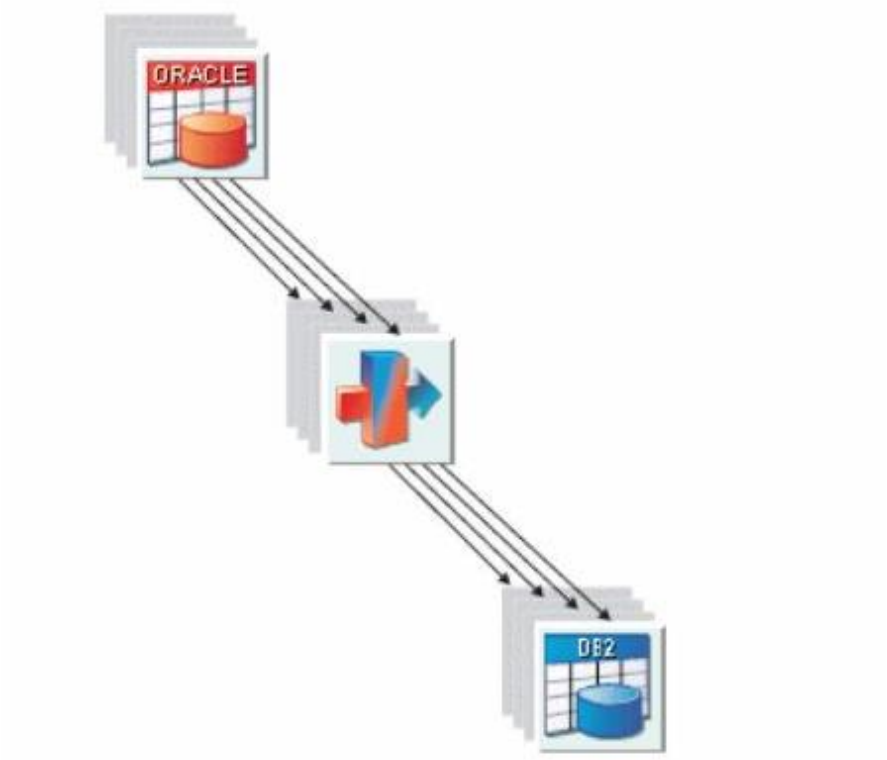
Entire Partitioning:

This type of partition should be chosen carefully. In this partition all the data from source is copied to all target nodes Mostly used for the lookup stage when data is small .



Combining pipeline and partition parallelism

The Information Server engine combines pipeline and partition parallel processing to achieve even greater performance gains. In this scenario you would have stages processing partitioned data and filling pipelines so the next one could start on that partition before the previous one had finished. This is shown in Figure



Collecting

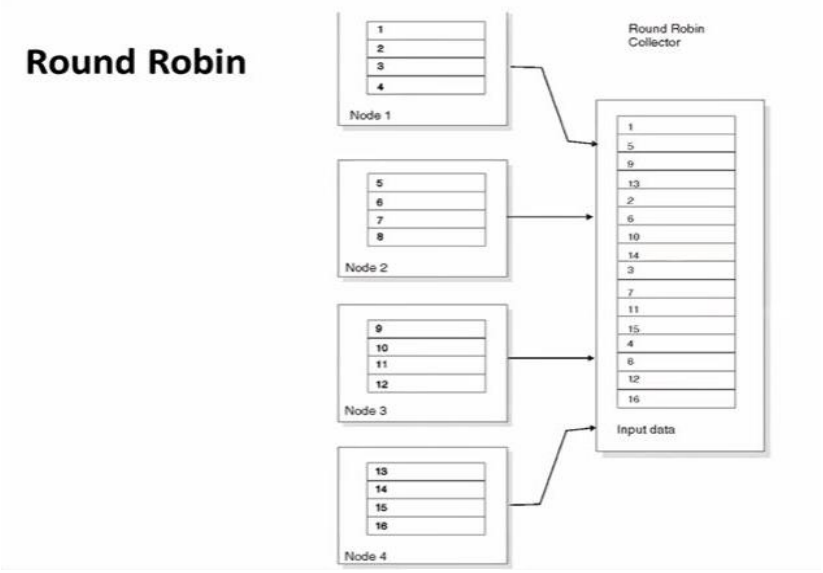
Collecting comes when you are writing data to your final target stage specially when writing to the sequential files. Sequential files do not have parallelism. So for writing the data coming from different streams it must be collected before writing to sequential file

We have some methods for collection of data these are listed below

- Round Robin Collection
- Ordered Collection
- Sorted Merge

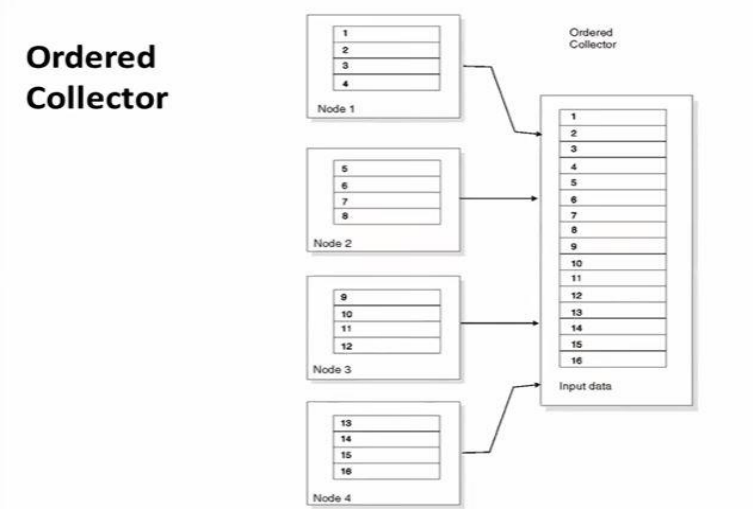
Round Robin Collection

In this collecting technique records are collected from nodes in a round robin fashion. One record from first node is collected then from second node and so on



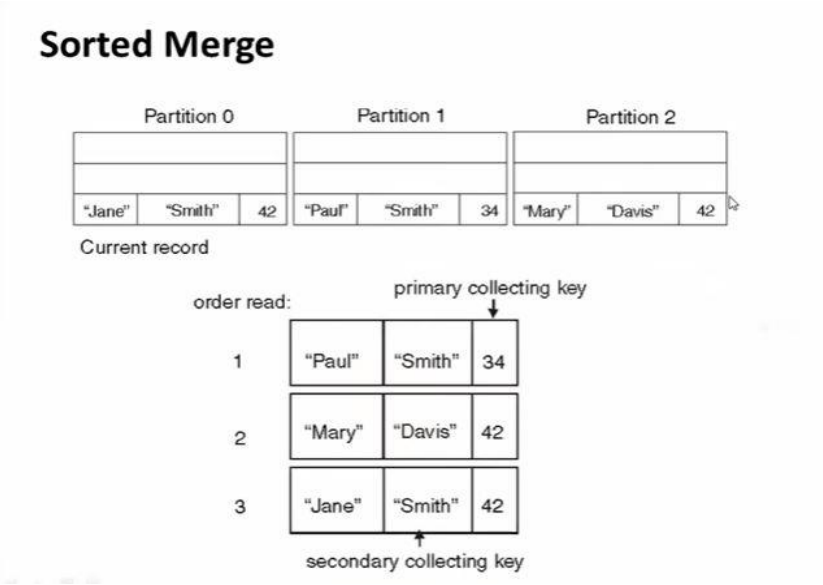
Ordered Collection

In ordered collection first all the records from 1st partition will be collected then all the record from second partition will be collected then third and so on



Sorted Merge Collection

It is a key based collection technique used when we sort the data based on some key column In this collecting technique we have to define a primary and secondary collecting as well



General properties of stages

These are some of the general properties we find in every stage of data stage.

Description

In description user can define short description of what stage is doing.

Execution Mode

We have three kinds of execution modes in data stage

Default

Default mode of execution is parallel.

Parallel

In parallel execution mode job will run in parallel on multiple nodes

Sequential

In sequential execution mode job will run in sequential manner no matter how many jobs have you defined for that job one node will execute job.

Combinability Mode

By default, every data stage job has to be converted into data stage orchestrate language. Orchestrate language is a combination of operators these operators might be similar across stages of a job.

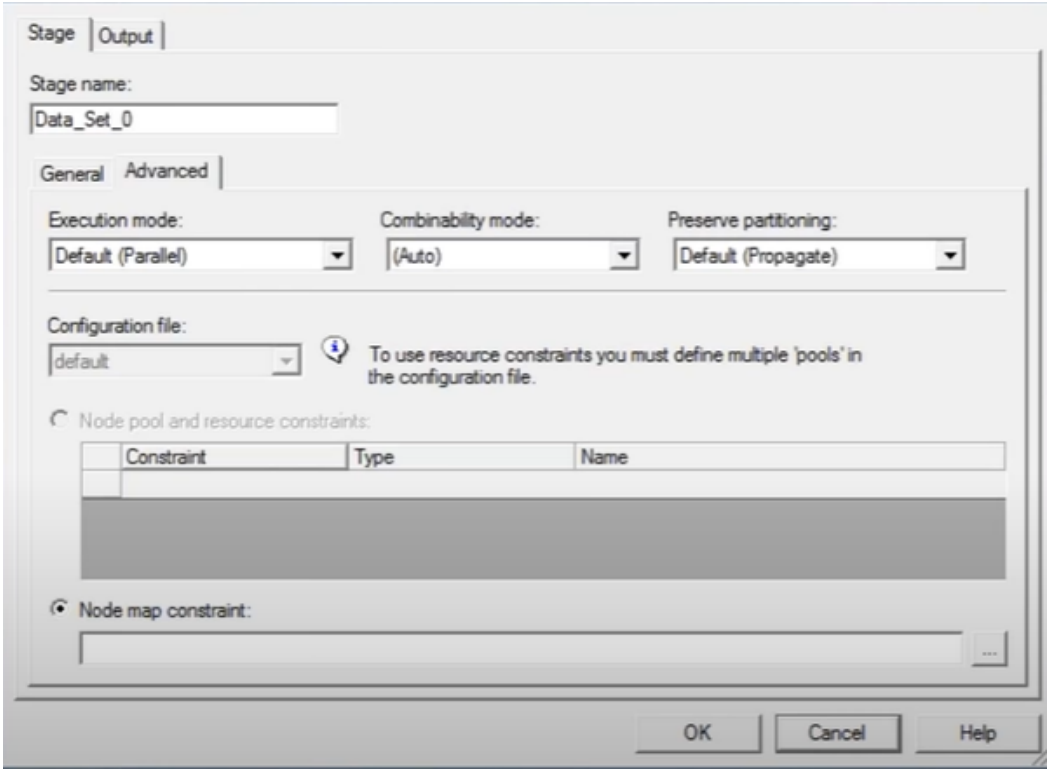
Combinable job means these similar operators are combined

Preserve Partition

Preserve partition means a stage can also request that the next stage in the job preserves whatever partitioning it has implemented.

Configuration files

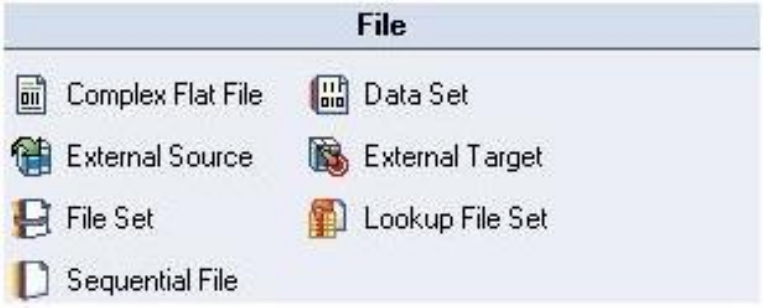
The configuration file defines the resources that data stage will use to execute the job. Most important in number of nodes



File stages in data stage

These are the following important file stages in DataStage.

- Sequential file stage
- Data set stage
- File set stage
- Lookup file set stage
- External source stage
- External Target stage
- Complex Flat File stage



Sequential file stage

The Sequential File stage is a file stage. It allows you to read data from or write data one or more flat files. The stage can have a single input link or a single output link, and a single rejects link.

These are the following main points in sequential file stage

- What are table definitions
- What is row number
- Filter command (sed '1d: $d') and its uses
- For using reject link set reject mode to output and in reject link we catch the records having data type mismatch, metadata mismatch, nullability mismatch etc
- We can also see reject reason
- What is Run time column propagation and why this needs to be enable only if we are reading from a schema file and mostly used when reading from external sources

Data set stage

The Data Set stage is a file stage. It allows you to read data from or write data to a dataset. The stage can have a single input link or a single output link. It can be configured to execute in parallel or sequential mode.

What is a data set? parallel jobs use data sets to manage data within a job. You can think of each link in a job as carrying a data set. The Data Set stage allows you to store data being operated on in a persistent form, which can then be used by other Infosphere DataStage jobs. Data sets are operating system files, each referred to by a control file, which by convention has the suffix .ds. Using data sets wisely can be key to good performance in a set of linked jobs. You can also manage data sets independently of a job using the Data Set Management utility, available from the Infosphere DataStage Designer or Director.

These are the main points for data set stage

- This is data stage native format and to delete any data set we have to use ALT admin command
- Mostly used to store intermediate data
- Has its own format which is understandable by data stage
- It has extension (.ds)

File set stage

The File Set stage is a file stage. It allows you to read data from or write data to a file set. The stage can have a single input link, a single output link, and a single rejects link. It only executes in parallel mode. What is a file set? Infosphere DataStage can generate and name exported files, write them to their destination, and list the files it has generated in a file whose extension is, by convention '.fs'.

The data files and the file that lists them are called a file set. This capability is useful because some operating systems imposea2GB limit on the size of a file and you need to distribute files among nodes to prevent overruns.

Lookup file set stage

The Lookup File Set stage is a file stage. It allows you to create a lookup file set or reference one for a lookup. The stage can have a single input link or a single output link. The output link must be a reference link. The stage can be configured to execute in parallel or sequential mode when used with an input link. When creating Lookup file sets, one file will be created for each partition. The individual files are referenced by a single descriptor file, which by convention has the suffix '**.fs'.**

External source stage

The External Source stage is a file stage. It allows you to read data that is output from one or more source programs. The stage calls the program and passes appropriate arguments. The stage can have a single output link, and a single rejects link. It can be configured to execute in parallel or sequential mode.

External Target stage

The External Target stage is a file stage. It allows you to write data to one or more source programs. The stage can have a single input link and a single rejects link. It can be configured to execute in parallel or sequential mode. There is also an External Source stage, which allows you to read from an external program

Complex Flat File stage

The Complex Flat File (CFF) stage is a file stage. You can use the stage to read a file or write to a file, but you cannot use the same stage to do both. As a source, the CFF stage can have multiple output links and a single reject link. You can read data from one or more complex flat files, including MVS™ data sets with QSAM and VSAM files. You can also read data from files that contain multiple record types.

The source data can contain one or more of the following clauses:

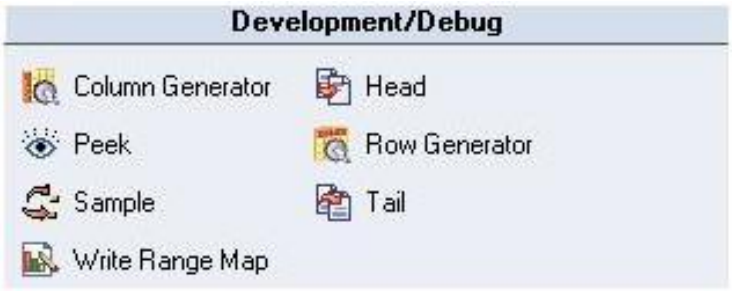
- GROUP
- REDEFINES
- OCCURS
- OCCURS DEPENDING ON

CFF source stages run in parallel mode when they are used to read multiple files, but you can configure the stage to run sequentially if it is reading only one file with a single reader. As a target, the CFF stage can have a single input link and a single reject link. You can write data to one or more complex flat files. You cannot write to MVS data sets or to files that contain multiple record types.

Development / Debug stages in data stage

These are the following important file in data stage.

- Row Generator stage
- Head stage
- Tail stage
- Peek stage
- Write range map stage



Row Generator stage

The Row Generator stage is a Development/Debug stage. It has no input links, and a single output link. The Row Generator stage produces a set of mock data fitting the specified Meta data. This is useful where you want to test your job but have no real data available to process. The Meta data you specify on the output link determines the columns you are generating. For decimal values the Row Generator stage uses float. As a result, the generated values are subject to the approximate nature of floating-point numbers. —Not all of the values in the valid range of a floating-point number are representable. The further a value is from zero, the greater the number of significant digits, the wider the gaps between representable values.

Head stage

Head stage can have a single input link and a single output link. It helps you to get sample data. The Head Stage selects the first N rows from each partition of an input data set and copies the selected rows to an output data set. You determine which rows are copied by setting properties which allow you to specify the number of rows to copy. The partition from which the rows are copied. The location of the rows to copy. The number of rows to skip before the copying operation begins. This stage is helpful in testing and debugging applications with large data sets. For example, the Partition property lets you see data from a single partition to determine if the data is being partitioned as you want it to be. The Skip property lets you access a certain portion of a data set.

Tail stage

Tail stage can have a single input link and a single output link. It helps you to get sample data. The Tail Stage selects the last N records from each partition of an input data set and copies the selected records to an output data set. You determine which records are copied by setting properties which allow you to specify the number of records to copy. The partition from which the records are copied This stage is helpful in testing and debugging applications with large data sets. For example, the Partition property lets you see data from a single partition to determine if the data is being partitioned as you want it to be. The Skip property lets you access a certain portion of a data set.

Peek stage

Peek stage can have a single input link and any number of output links. The Peek stage lets you print record column values either to the job log or to a separate output link as the stage copies records from its input data set to one or more output data sets.

Processing stages in data stage

These are some the following important processing stages in data Stage.

- Aggregator
- Copy
- FTP
- Filter
- Funnel
- Join
- Look up
- Merge
- Modify
- Remove Duplicates
- Change Capture

Aggregator

Aggregator joins data vertically from grouping the incoming data stream and calculating brief about a min, count, max etc.; for each team. The data could be sorted using two methods: pre-sort and hash table.

Copy

This stage copies the input data to single or more output data flows.

FTP

This stage implements the FTP protocol to transfer data to a remote machine

Filter

This stage filters records that don't meet relevant requirements.

Funnel

This stage converts multiple streams into a single one.

Join

This stage combines more than one input according to the values of a key column/s.

Lookup

This stage includes two or more inputs based on values of key column/s. It should have a 1 source, but can have multiple lookup tables

Merge

This stage includes one master input with multiple updates inputs related to the values of a key column/s. The inputs need to be sorted and unmatched secondary entries can be caught using multiple reject links.

Modify

This stage alters the dataset record. Useful for renaming columns, not default data type conversion and null handling.

Remove duplicates

This stage wants input to be single sorted data set.

Slowly changing Dimension

This stage computerizes the revised process of dimension tables where data frequently change.

Sort

This stage sorts the input columns

Transformer

This stage handles validation of data, extracted data and lookups.

Change Capture

This stage catches the before and the after states of 2 data which are inputted and are converted into a single data set that shows the differences made.

Change

This stage will apply changes to the operation to the previous dataset so as to compute after dataset. It retrieves data from the change capture stage.

Difference

This stage executes a record-by-record comparison of 2 input data and outputs the record which is the difference between records.

Checksum

This stage produces checksum from the specific columns in a row and gets added to the stream. Also, able to differentiate between records.

Compare

This stage does a searching comparison on column-column of pre-sorted records. It can have one output and two input links.

Encode

This stage encodes data, such as gzip with encoding command

Decode

This stage decodes the previously encoded data in the previous stage.

External Filter

This stage allows specifying an OS command that filters the processed data.

Generic

This stage permits users to provoke OSH operator from DataStage stage having options.

Pivot Enterprise

This stage is used for horizontal pivoting. It maps or assigns multiple columns in the input row to a single column over multiple output rows.

Surrogate Key Generator

This stage manages key source by generating to a column, the surrogate key.

Switch

This stage matches each input row to an output link on the basis of the value of a selector field. The concept is similar to switch statement in most programming languages.

Compress

This stage combines data set using the GZIP utility (or can be done using LINUX/UNIX command)

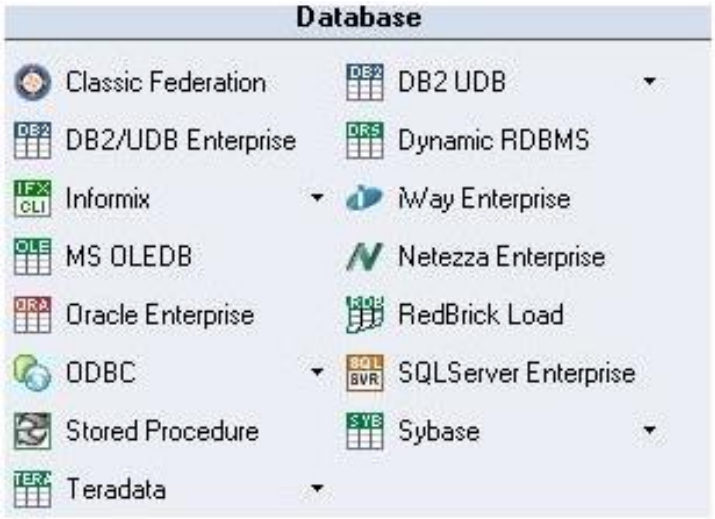
Expand

This stage extracts previous compressed data set into raw binary data.

Database stages in data stage

These are some the following database stages in data Stage.

- Oracle Enterprise
- ODBC Enterprise
- DB2/UDB Enterprise
- Teradata
- SQL Server Enterprise
- Stored procedure



Oracle Enterprise

This stage permits reading data to an Oracle database.

ODBC Enterprise

This stage permits looking data from and writing it to a database called as ODBC source. It is used in the processing of data in Microsoft Access and Excel spreadsheets.

DB2/UDB Enterprise

This stage allows writing, reading data to a DB2 Database.

Teradata

This stage allows writing, reading data to a Teradata data warehouse. Three Teradata stages are present; Teradata connector, Enterprise and Multiload.

SQL Server Enterprise

This stage allows writing, reading data to Microsoft SQLI Server 2005 and 2008 database.

Sybase

This stage allows into Sybase Databases, reading and writing data.

Stored procedure

This stage chains DB2, Oracle, Sybase, Teradata and Microsoft SQL Server.

MS OLEDB

This stage is used to retrieve information from all types of the information repositories, such as an ISAM file, relational Source or a spreadsheet.

Dynamic Relational

This Stage is reading from and writing into a different supported relational DB using interfaces such as Microsoft SQL, DB2, Oracle, Sybase and Informix.

Data quality stages in data stage

These are some the following data quality stages in data Stage.

- Investigate
- Match frequency
- MNS
- WAVES

Investigate

This stage predicts data module of appropriate columns of all records from the source file. Offers word and character investigation methods.

Match frequency

This stage obtains input from a file, a database or processing stages and generates an occurrence distribution report.

MNS

This stage refers to Multinational address standardization

WAVES

This stage refers to worldwide address verification and enhancement system.

Data stage best practices

The following are the points for DataStage best practices:

- Select suitable configurations file (nodes depending on data volume)
- Select buffer memory correctly and select proper partition
- Turn off Run time Column propagation wherever it's not required
- Taking care about sorting of the data.
- Handling null values (use modify instead of transformer)
- Try to decrease the use of transformer. (Use copy, filter, modify)
- Use data-set instead of sequential file in the middle of the vast jobs
- Take maximum 20 stages for a job for best performance.
- Select Join or Lookup or Merge (depending on data volume)
- Stop propagation of unnecessary metadata between the stages.