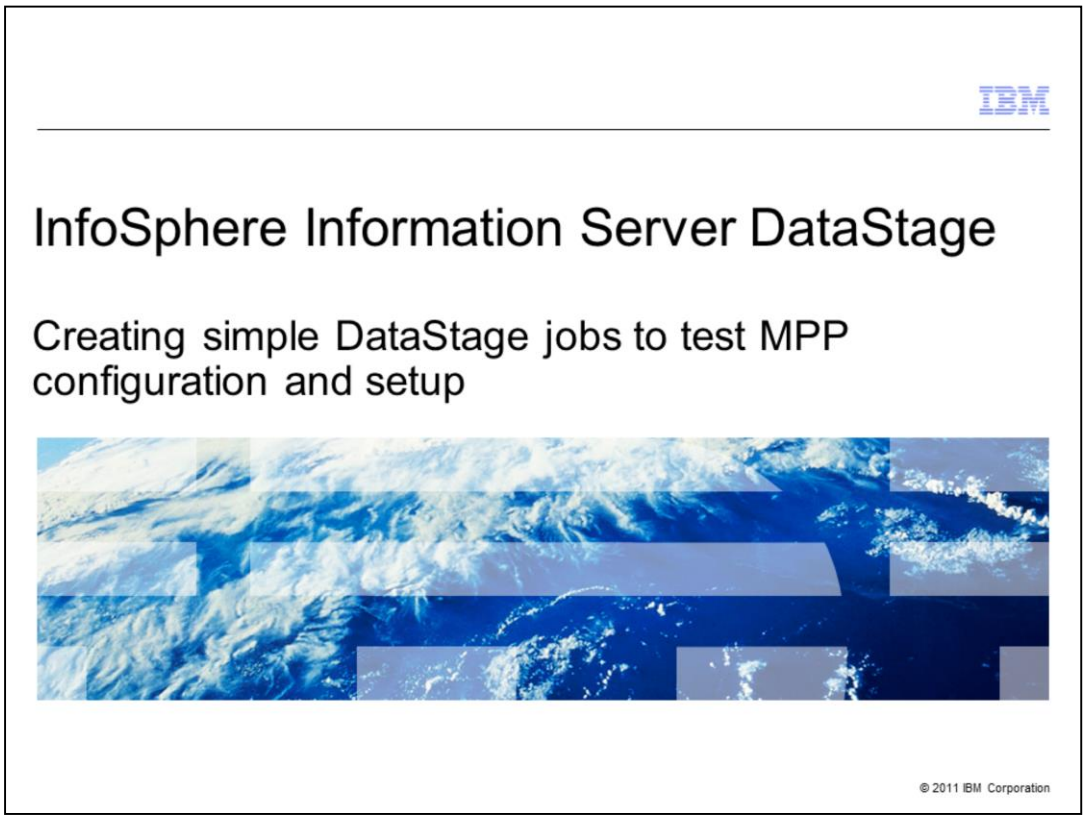


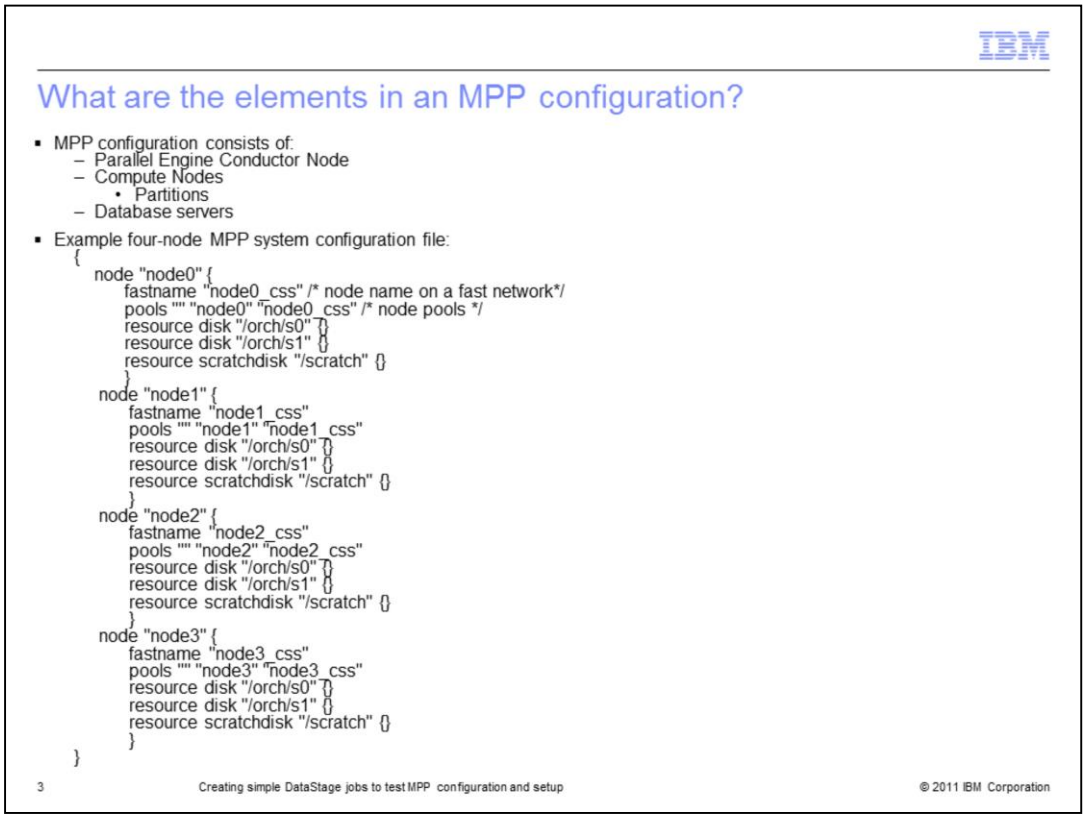
This presentation will discuss how to test your MPP configuration for DataStage® 7 and 8 by using seven simple DataStage jobs.

The objectives of this presentation are to understand what elements make up a MPP configuration with InfoSphere™ Information Server and how to test that the configuration is set up correctly with seven simple DataStage jobs.

This document will not discuss how to set up the MPP environment. For information on this, look to the Information Server documentation in the Information Center. For Information Server 8.5, the provided URL has information on how to add computers to share engine processing. More information can be found within the IBM InfoSphere Information Server Planning, Installation, and Configuration Guide.

The assumption is that the MPP configuration is complete and simple jobs can be created to ensure that the configuration is correct.

This presentation also expects you to retain some mastery in administrating InfoSphere Information Server and have some development knowledge of Parallel jobs.



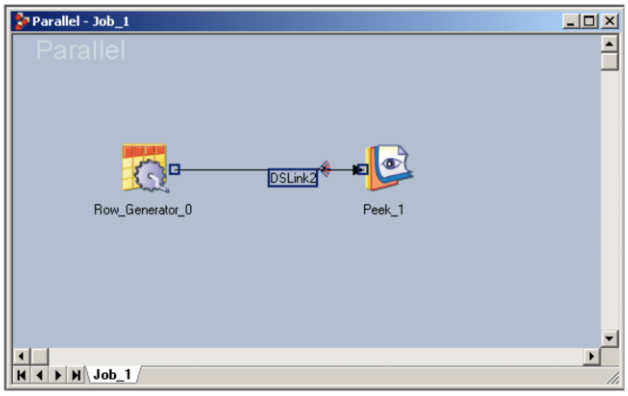
At a high level, InfoSphere Information Server with a MPP configuration must have certain elements. These elements include the Parallel Engine Conductor node, one or more compute nodes with one or more partitions, and a database server accessible from the conductor and compute nodes.

Each host runs its own image of the operating system and retains its own processors, disk, I/O resources and memory.

The configuration file displayed on this slide is for a four node configuration. Pay attention to how the use of fastname indicates the location of the server the jobs are assigned to. fastname must be the output of the “uname –n” command on that particular server. In a MPP environment, each node will have its own disks, therefore its own /orch/s0, /orch/s1 and /scratch. It is not necessary to have the same resource disk or scratch disk paths between nodes.

Testing MPP configuration – Job 1

- Test ability to propagate jobs on multiple nodes
- Check basic connectivity and configuration
- RowGenerator to Peek job



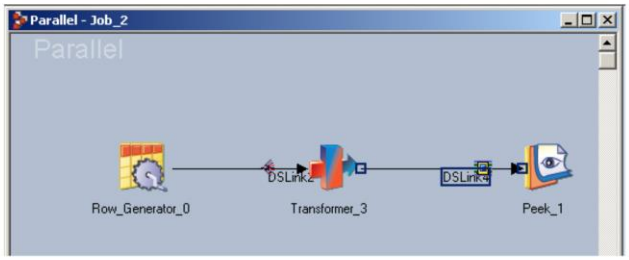
4 Creating simple DataStage jobs to test MPP configuration and setup

This first example job will test the ability to propagate jobs on multiple remote nodes. This will confirm that RSH/SSH privileges are properly set and that the peek stage can start on all servers.

To create this job, use a row generator stage to a peek stage. Generate a nominal number of rows for a couple columns to test this connectivity. Configure the RowGenerator stage to run in parallel. For this job and all subsequent jobs in this presentation, be sure that $APT_CONFIG_FILE is set to point to a configuration file that is set up similar to the example displayed in the previous slide. Also, when running this job and all other jobs in this presentation, set the environment variable $APT_DUMP_SCORE to true on the job properties to confirm that all nodes are being used.

Testing MPP configuration– Job 2

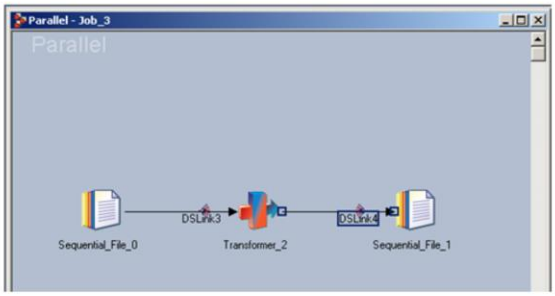
- Project folder is visible on all servers
- Project is in same location on all servers
- Transformer code properly propagated to remote nodes
- RowGenerator => Transformer => Peek
- Relevant Technote: https://www-304.ibm.com/support/docview.wss?uid=swg21512049



The second job will make sure that the project folder is available on all of the nodes and that the propagation of the transformer code is properly configured. This job consists of a RowGenerator stage linked to a Transformer stage which in turn is linked to a Peek stage. Once again, it is not required to generate a large number of rows or have a large number of columns in this job. Transformer stage is a user generated operator that does not already exist on all nodes, unlike all other operators. It will be auto-propagated if properly configured. The technote referenced on this slide discusses the software install requirements when using DataStage on a MPP.

Testing MPP configuration– Job 3

- Check I/O by adding file system onto job
- Sequential File => Transformer => Sequential File
- Relevant Technotes:
  https://www-304.ibm.com/support/docview.wss?uid=swg21457724
- https://www-304.ibm.com/support/docview.wss?uid=swg21496848

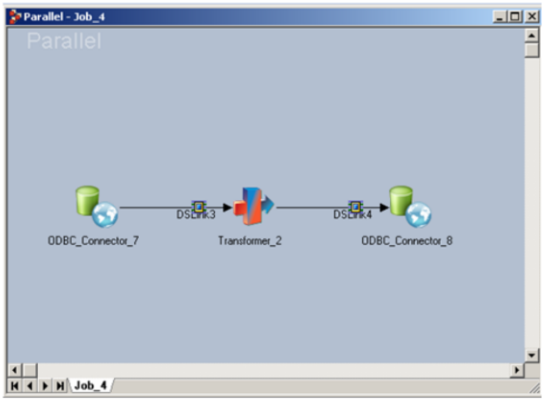


The third job will test the I/O or the handling of the file system on the MPP setup. This job will use a sequential file stage that links into a parallel transformer stage which in turn outputs to a sequential file stage. It is good practice to use a sequential file that represents the data that will be used in the actual jobs.

The technotes referenced on this slide discusses common disk problems such as disk space, ulimit, and how to detect and correct these issues.

Testing MPP configuration– Job 4

- Check Database connectivity
- Database => Transformer => Database



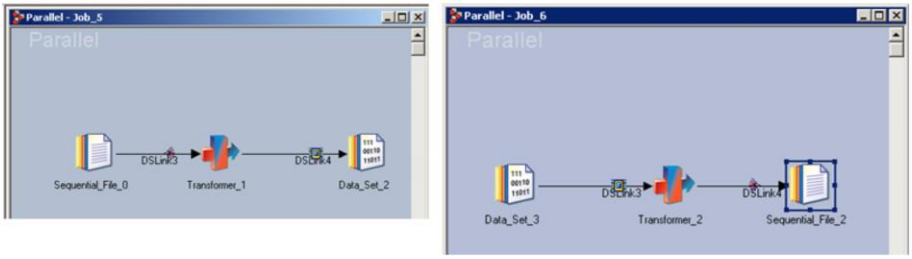
7 Creating simple DataStage jobs to test MPP configuration and setup

The fourth job will confirm that DataStage is able to connect to source and target databases from all of the nodes. It will also ensure that the client version of the databases is available on all of the authorized nodes and is functioning correctly.

For simplicity, use the ODBC connector stage as shown in the job displayed on this slide and put in the connection information for the databases that the jobs will be connecting to. Use the ODBC Connector stage for the database source. Link to a Transformer stage which in turn will link to another ODBC Connector stage pointing to the target database. It may be necessary to use other database-specific stages in lieu of the ODBC connector stage to further test connectivity.

Testing MPP configuration– Jobs 5 and 6

- Check resource space
- Write
  - Sequential File => Transformer => Data Set
- Read
  - Data Set => Transformer => Sequential File
- Relevant Technote:
  https://www-304.ibm.com/support/docview.wss?uid=swg21393359



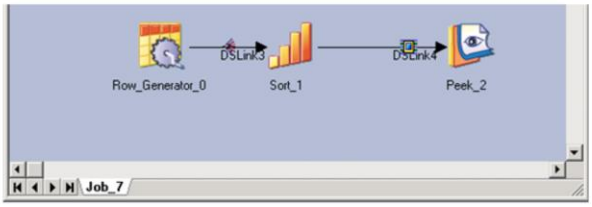
8 Creating simple DataStage jobs to test MPP configuration and setup

The fifth and sixth jobs will confirm that there is sufficient resource space for both reading and writing. For the data in job 5, use a Sequential File that represents the amount of data that might be processed by actual production jobs. Create a job that uses a Sequential File stage that links to a Transformer stage that in turn, outputs to a Data Set stage. This will test the resources space when writing a dataset.

For job 6, use the previously created Data Set as input to a transformer and then write it out to a Sequential File Stage. This will test the resource space for the read. It will also test the ability to read from the dataset even if the job is running on different nodes. The technote referenced on this slide discusses some of the ways to examine resources to determine if enough space has been allocated.

Testing MPP configuration – Job 7

- Job will check
  - Scratch Space
  - Swap space
  - Memory
- RowGenerator stage to Sort stage to Peek stage job
- Relevant Technote:
  https://www-304.ibm.com/support/docview.wss?uid=swg21444852



9 Creating simple DataStage jobs to test MPP configuration and setup

The seventh job will confirm that there is sufficient scratch space on each of the nodes. This job will check this by generating an exceptionally high number of records and using the sort operator to rearrange the records. Ideally, the job should generate a similar volume data that will be processed in a normal load.

By adding the sort, the job will be using memory, swap and scratch space. To create the job, use a RowGenerator stage that generates several rows that represents about 20% over the expected job load. Ideally, the columns should also be similar to what will be used in the actual jobs. The RowGenerator stage links to a Sort stage which in turn outputs to a Peek stage.

Only checking scratch space without running this type of job can give a false indication that there is sufficient disk space. Review the technote referenced on this slide for information on the errors that can be received and how to correct any issues.

Trademarks, disclaimer, and copyright information