InfoSphere DataStage Grid Solution

Julius Lerm
IBM Information Management
What is Grid Computing?

Grid Computing doesn’t mean the same thing to all people.

• Definitions include:
  • Using Idle machines on the internet
  • Using Idle desktop machines within the company
  • Using any server that’s not currently in use
    • Regardless of OS, physical Location, CPU speed ...
  • Anything running more than 1 Linux box in any manner
  • Anything running on any computer when you don’t care which computer it runs on
What is Grid Computing with InfoSphere DataStage?

- Low cost solution that provides high throughput processing
  - InfoSphere DataStage Grid Toolkit
- Allows for maximum resource allocation flexibility to one or more project teams
- Available for DataStage, QualityStage, and Information Analyzer
- Enables both grid distribution methods simultaneously
  - assigning jobs to specific servers in the grid
  - assigning a single parallel job to run across multiple servers
- Platforms:
  - RedHat or SuSE (Intel/AMD)
  - AIX/Power
What Is Driving Rapid Customer Adoption of Data Integration Grid?

**Better decisions based on better data yields ROI**

- Grid-based integration makes it possible for companies to process and analyze larger data volumes, create a consolidated view of data, and put the right data into the enterprise data warehouse and other critical enterprise applications
  - More sources of data, more data from each source, better matching, real-time versus batch
- Better data yields:
  - Better business decisions
  - Enhanced customer relationships
  - More cross selling and upselling
  - New services delivered to customers

**Reduced Data Integration Costs**

- Reduced administration and operating costs – centralization of staff
- Reduced data integration project costs – lower cost per project delivered by data integration center of excellence versus siloed projects
- Reduced hardware costs
Benefits of Grid Computing

• Low cost hardware
• High-throughput processing
• Significant ROI (Return on Investment) for data management solutions
• Supports a high-availability (HA) solution
  • Resource manager monitors availability of hardware at startup / job deployment time
• SLA (Service Level Agreement)
  • Consistent runtimes
  • Isolates concurrent job executions
• Shared resource pool
  • Not typical silo-ed environment
  • Hardware shared across multiple environments and departments
Why Grid?

*Improve the return on infrastructure investments!*

- Help improve infrastructure price/performance
- Improve the utilization of computing resources
- Help provide unlimited scalability and offer capacity on demand
- Optimize the allocation of resources to applications
- Help reduce complexity, consolidate servers, storage and data centers
- Provide a highly available environment
- Help eliminate single points of failure
- Optimize use of available processing resources
- Ensures that application tasks complete within stable predictable time frame → improving SLA performance
“Silo-ed” architecture & proliferation of SMP servers:

- Higher capital costs through limited pooling of IT assets across silos
- Higher operational costs
- Limited responsiveness due to more manual scheduling and provisioning
- Inherently more vulnerable to failure
- No ability to exploit available capacity when other teams are idle
After Grid

DataStage Multi-Process Grid Framework

ProfileStage Project 1 → Node 1
DataStage Project 2 → Node 2
QualityStage Project 3 → Node 3
DataStage Project 4 → Node 4
IBM Software Project N → Node N

“Virtualized” infrastructure:

- Creates a virtual data integration collaboration environment
- Virtualizes application services execution
- Dynamically fulfills requests over a virtual pool of system resources (nodes)
- Offers an adaptive, self-managed operating environment that guarantees high availability
- Delivers maximum available capacity to anyone participating in the grid
InfoSphere DataStage Grid Definition

It’s an InfoSphere DataStage Cluster that supports the dynamic creation of Config Files

- All the requirements for a Cluster apply
- Adds
  - Resource Manager
  - Grid Toolkit (GTK)
Cluster Requirements

- Shared Storage
- NIS/LDAP
- Networking
- Passwordless SSH
- HA Solution
- DB Connectivity
- Users
- Review System Requirements

Platform Requirements Usually Implemented by Client’s own sysadmins and storage/networking personnel
Sample IS Grid

Public Network

Private Grid Network

External DBs
IS Client Tools

IIS Services (Domain) Layer (WAS) 4-Cores
IIS Engine Layer (DB) Engine 4-Cores
Compute Node Standby for IIS Services and Engine 4-Cores

Linux-HA Failover for IIS Domain
Linux-HA Failover for IIS Engine

Fiber Channel Connections

SAN

Disk Array
XMETA DB /opt /home
Databases Scratch Staging

IIS Metadata Layer (XmetaDB) Managed by DBAs

Storage shared across all nodes via Cluster File System, with exception of Scratch (there should be at least one scratch file system dedicated to each compute node)

Dedicated to Information Server Grid
Resource Management

- Tracks resources (nodes) based on which jobs are already running, which servers are down
- Queues jobs when no resources are available
- Provides a list of nodes that are assigned for a job
- Extensive advanced features
  - We leverage a subset of the features
- Manager node where tasks are scheduled and resources allocated
  - Usually happens on the head node
- Compute nodes have agent processes that communicate back to the manager
- Jobs (scripts or executables) are started on compute node, not head node
LoadLeveler Classes

• LoadL_admin file:

```plaintext
dsbatch:
  type = class  # class for medium jobs
  priority = 50  # ClassSysprio
  max_total_tasks = 50
  class_comment = "Class for DataStage batch jobs"

dsrealtime:
  type = class  # class for medium jobs
  priority = 50  # ClassSysprio
  max_total_tasks = 50
  class_comment = "Class for DataStage batch jobs"
```

• LoadL_config.local:

```plaintext
CLASS = dsbatch(4) dsrealtime(3)
```
Grid Enablement Toolkit

• **What does it do?**
  • Prebuilt integration with resource managers
  • Coordinates activities between the parallel framework and the resource manager
  • Creates the parallel configuration file to drive the dynamic assignment of compute resources
  • Logging (interaction w/ RM, usage details)
1) Start DS job

2) Request a number of resources

3) Return list of hostnames

4) Create dynamic parallel config file

5) Activate parallel job

6) The parallel job runs on the hostnames returned by LoadLeveler
## Grid Environment Variables In Administrator

### Environment variables

The following categorized environment variables are defined in this project. Either set a default value for an existing environment variable or add a new environment variable to the user defined category.

<table>
<thead>
<tr>
<th>Categories</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td></td>
</tr>
<tr>
<td>Customize</td>
<td></td>
</tr>
<tr>
<td>Parallel</td>
<td></td>
</tr>
<tr>
<td>Operator Specific</td>
<td></td>
</tr>
<tr>
<td>Reporting</td>
<td></td>
</tr>
<tr>
<td>Compiler</td>
<td></td>
</tr>
<tr>
<td>Grid</td>
<td></td>
</tr>
<tr>
<td>User Defined</td>
<td></td>
</tr>
</tbody>
</table>

### Categories:

- **General**
- **Customize**
- **Parallel**
- **Operator Specific**
- **Reporting**
- **Compiler**
- **Grid**
- **User Defined**

### Details:

<table>
<thead>
<tr>
<th>Name</th>
<th>Prompt</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>APT_GRID_COMPUTENODES</td>
<td>COMPUTENODES - Number of Compute Nodes</td>
<td>1</td>
</tr>
<tr>
<td>APT_GRID_CONFIG</td>
<td>APT_GRID_CONFIG - Use this configuration template file vs</td>
<td></td>
</tr>
<tr>
<td>APT_GRID_ENABLE</td>
<td>ENABLE - Use DYNAMIC_CONFIG_SCRIPT (NO/YES)</td>
<td>NO</td>
</tr>
<tr>
<td>APT_GRID_IDENTIFIER</td>
<td>IDENTIFIER - name to appear within Resource Manager for</td>
<td></td>
</tr>
<tr>
<td>APT_GRID_OPTS</td>
<td>APT_GRID_OPTS - Specific Grid engine command flags</td>
<td></td>
</tr>
<tr>
<td>APT_GRID_PARTITIONS</td>
<td>PARTITIONS - Number of partitions per Node (compute)</td>
<td>1</td>
</tr>
<tr>
<td>APT_GRID_QUEUE</td>
<td>QUEUE - Resource Manager Queue name</td>
<td></td>
</tr>
<tr>
<td>APT_GRID_SCRIPTPOST</td>
<td>SCRIPTPOST - Script to run after running job, before release</td>
<td></td>
</tr>
<tr>
<td>APT_GRID_SCRIPTPRE</td>
<td>SCRIPTPRE - Script to run after resource manager assigns r</td>
<td></td>
</tr>
<tr>
<td>APT_GRID_SEQFILE_HOST</td>
<td>SEQFILE_HOST - First Host name identified by Resource Mgr</td>
<td></td>
</tr>
<tr>
<td>APT_GRID_SEQFILE_HOST2</td>
<td>SEQFILE_HOST2 - Second Host name identified by Resource Mgr</td>
<td></td>
</tr>
<tr>
<td>APT_GRID_STAT_CMD</td>
<td>STAT_CMD - vmstat or sar command syntax to run on first node</td>
<td></td>
</tr>
</tbody>
</table>
Grid Toolkit Environment Variables

- **APT_GRID_ENABLE**
  - YES: Current osh will intercept the run script to create a new configuration file
  - NO: Use the existing configuration file

- **APT_GRID_QUEUE**
  - Name of the Resource Manager queue the job will be submitted to

- **APT_GRID_COMPUTE_NODES**
  - The number of compute nodes required for the job
  - Used to request the number of compute nodes in the dynamically created configuration file
  - A compute node is a server that can be used for processing
    - Not e.g. dedicated for IO or DB2
  - Default value is 1

- **APT_GRID_PARTITIONS**
  - Used to create multiple partitions for each compute node
  - Default value is 1
TotalPartitionsRule=cores
APT_GRID COMPUTENODES <= # slots for selected queue
APT_GRID PARTITIONS always set to 1
Dynamic Config file

Static Config File
Grid Config Template

```
{  
  node "Conductor"
  {  
    fastname "host1.acg.iips"
    pools "conductor"
    resource disk "/opt/IBM/InformationServer/Server/Datasets" {pools ""}
    resource scratchdisk "/tmp" {pools ""}
  }
  node "$$Compute"
  {  
    fastname "$$fastname"
    pools ""
    resource disk "/opt/IBM/InformationServer/Server/Datasets" {pools ""}
    resource scratchdisk "/tmp" {pools ""}
  }
}~
```
node "Conductor"
{
    fastname "host1.acg.ips"
pools "" resource disk "/opt/IBM/InformationServer/Server/Datasets" {pools ""}
resource scratchdisk "/tmp" {pools ""}
}
node "Compute1"
{
    fastname "host2.acg.ips"
pools "" resource disk "/opt/IBM/InformationServer/Server/Datasets" {pools ""}
resource scratchdisk "/tmp" {pools ""}
}
node "Compute2"
{
    fastname "host2.acg.ips"
pools "" resource disk "/opt/IBM/InformationServer/Server/Datasets" {pools ""}
resource scratchdisk "/tmp" {pools ""}
}
node "Compute3"
{
    fastname "host2.acg.ips"
pools "" resource disk "/opt/IBM/InformationServer/Server/Datasets" {pools ""}
resource scratchdisk "/tmp" {pools ""}
}

APT_GRID_COMPUTENODES=3
Information Server (IS) Grid Solution Offering

A total solution approach to implement an Information Server Grid system based on proven methods & best practices

Features
This offering consists of the following phases:

- Education & Project Planning
  - Common understanding of IS Grid
- Architecture & Design
  - Customer responsible for building the Grid Infrastructure
- IS Grid Implementation
  - Install & configure the IS Grid Framework
  - Test & Validate
  - Address Administration Hand-Off
  - Monitor during 24 hours

Benefits
- Leverage repeatable, proven processes and standard collaterals to reduce costs and project risks
- Accelerate the time to value and return on investment with the knowledge and best practices brought by our Information Server Grid experts
- Gain knowledge transfer and mentoring from our experts

Deliverables
- Environment Prerequisite Checklist
- Grid Planning & Architecture Document
- Project Plan
- System Configuration Guide
- Standard Collateral
  - Build your own Grid Toolkit
  - Grid Enablement Toolkit
  - Basic High Availability

Backed by world class industry and product experts in deploying InfoSphere software
Project Approach

- **Education & Project Planning Workshop (1 week)**
  - Educate/obtain a common understanding about a grid environment
  - Start to create a customer environment prerequisite checklist

- **Architecture & Design Workshop (15 days spread in 3 to 4 weeks)**
  - Interactive discussion/collaboration with the customer to finalize the architecture of the grid environment
  - Finalize the customer environment prerequisite checklist

- **Implementation of the Infrastructure by Customer or GTS**
  - Following the customer environment pre-requisite checklist (System Requirements)

- **IS Grid Implementation (3 weeks)**
  - Review/check that the customer environment is ready
  - Install & Configure Information Server Grid Framework based on the defined architecture
  - Migrate existing parallel DataStage jobs & sequences (max. 50) to the new environment (optional, as needed)
  - Two days monitoring period

**One SOW that includes all three Phases**
Deliverables

- **Education & Project Planning Workshop**
  - Understanding all infrastructure implication to operate a GRID Environment (Standard Collateral’s)
    - Build your own Grid Toolkit
    - Grid Enablement Toolkit
    - Basic High Availability
    - Project Plan

- **Architecture Design**
  - Environment Prerequisite Checklist
  - Grid Planning & Architecture

- **IS Grid Implementation**
  - Grid Toolkit Software
  - System Configuration Guide
  - Validated Test
# Education Prerequisite

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Audience</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>DX444</td>
<td>DataStage Essentials</td>
<td>Learn about DataStage V8.1 in its IBM Information Server environment. Learn how to build DataStage parallel jobs that read and write data to and from a variety of data stores including sequential files, data sets, and relational tables. Also, learn how to build parallel jobs that process data in a variety of ways: business transformations, data filtering, data combining, data generation, sorting, and aggregating.</td>
<td>4 days</td>
</tr>
</tbody>
</table>
Other Reference

- **GTS Grid Offerings:**
- **GTS - Implementation Services for SAN storage software**
- **GTS - Implementation Services for Network Attached Storage systems**
- **Grid Redbook:**
- **White Paper:**
- **Tivoli Dynamic Workload Broker:**