

Clustering in JReport

JReport Server Live provides reliable, high performance reporting able to meet the stringent application requirements of enterprises across industries like financial services, government, technology, telecommunications, etc. JReport Clustering technology is designed to provide two benefits. First, unlimited scalability so that as the use of JReport increases, capacity will never run out due to software limitations. And secondly, by having no single point of failure so that any computer in the cluster can go offline and JReport Server Live will still be able to provide reporting services with zero downtime, enhancing the overall reliability of the system.

Clustering Overview

There can be anywhere from two nodes to hundreds of nodes (clustered servers) that play the same role in a clustered server. Each server is capable of performing every function in the cluster so no special servers are required to manage other servers.

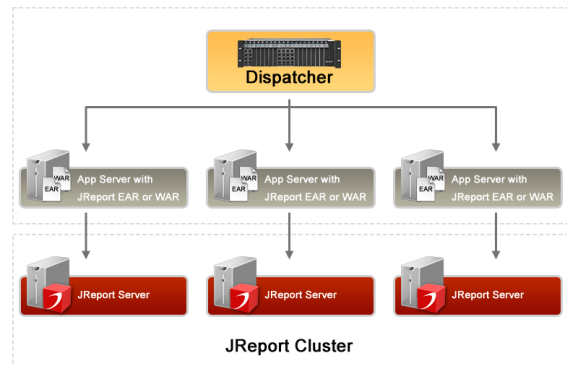
	Business Tasks			Administrative Tasks			
	Run Reports	Submit Scheduled Tasks	Load-Balancing	Failover	Load-Balancing Administration	Security Administration	Resource Administration
Clustered Server	Y	Y	Y	Y	Y	Y	Y

Sample Clustered Server Task Capabilities

JReport Server Cluster

A JReport Server Cluster is a distributed cluster in which a group of servers work together to provide cluster-wide shared resources, security, schedules and version services that manage multiple versions of resources. In a JReport Server Cluster, all clustered servers play exactly the same role. New servers can be added to the existing cluster or removed from the cluster at any time. Dynamic load balancing ensures that all servers in the cluster are utilized efficiently and the JReport Monitor allows administrators to manage utilization for the entire cluster and add and remove resources as needed. Notifications can be set up to ensure administrators are contacted if any servers encounter problems.

A common architecture is to have a dispatcher that distributes user requests from the web to one or more application servers. As illustrated in the JReport Server Cluster infrastructure diagram below, each application server instance contains one JReport Server Live.



With JReport, the maximum number of servers is unlimited and is based on anticipated loads and required protection from system failures. Often with multi-CPU and multi-Core systems, better overall throughput can be achieved by having several nodes on a single physical server. With too few nodes some resources will be under utilized and too many nodes will result in thrashing and lower total throughput. Only by testing in the specific environment will the number of nodes yielding the highest performance be found.

Cluster Management and Monitoring

Choosing a DBMS

By default, JReport includes the Apache Derby DBMS for the server data such as resources, users, groups, roles, etc. Each installed node creates its own database. In order to use a JReport Cluster, all nodes must use the same database. It is a simple task to point each node in the cluster to a single Derby network DBMS. However,

Derby does not have failover allowing a single point of failure in the DBMS itself. If a reliable DBMS such as MySQL or Oracle is already working within the application, this can also be used for the server database rather than maintaining a separate one for JReport.

Administering Security and Resources

In a distributed cluster, all administrative tasks can be accomplished from any single node, including security administration. After logging on to the cluster as an administrator, users, groups, realms, permissions, and ACLs can all either be created, removed or edited depending on what is needed in the application.

Notification of Server Down

JReport Cluster includes a feature to send notification emails to a specified address when the server is down so that if a member server crashes or is disconnected from the cluster, the disruption can be addressed immediately.

Monitoring Clustered Servers

JReport Monitor is a standalone web-based application used for monitoring the overall performance of JReport Server Live. JReport Monitor inspects the status of JReport Server Live, shows server performance statistics in graph/text mode, provides tools for administering Server Live, and creates profiling reports for performance and statistics.

JReport Server Cluster Main Features

Cluster Scheduler Lease

All nodes in a JReport Server Cluster have the ability to act as a scheduler. This prevents having a single point of failure for running scheduled tasks. However, depending on the reporting workload, only a subset of the servers may be “Active” as a scheduler, up to a configurable limit in order to provide better overall throughput of the system.

There are two additional parameters that can be set, one to determine the amount of time that the lease holder will continue to compete for scheduled tasks to run, (default of 300 seconds) and the other to determine the amount of time between when non-lease nodes check whether a lease is available to pick up (default of 30 seconds).

Load Detection

There is a JReport Server Live residing in each node of a JReport Server Cluster. The number of concurrent reports running on every JReport Server Live is one factor that most affects load balancing. In order to avoid heavy loads, every member server in the cluster sends the number of concurrently running reports on it to the other cluster nodes. Thus if the server is already running the maximum number of reports allowed, the scheduler will send the request to a different available server.

Load Balancing

As for scheduled tasks in a cluster environment, JReport Server Live provides a load balancing mechanism enabling the server to work more effectively. There are many benefits of deploying load balancing in a JReport Server Cluster some of which include, automatically allocating tasks to suitable servers based on their current load and performance capabilities and making sure that all of the servers in the cluster are fully utilized.

For example, every node in a clustered server is authorized to be a scheduler, and among the schedulers, those with a lease are considered "active" schedulers. When the time of a scheduled task arrives, active schedulers compete for the task and the winning scheduler then triggers the task to run. When dispatching tasks, the server with the winning active scheduler will select a server according to a load balancing algorithm and allocate the task to it. Load balancing in a JReport Server Cluster will also automatically re-balance the network load when a server is added or removed.

Built-in Load Balancing Algorithms

JReport Server Cluster supports several algorithms for load balancing among servers. Configurable algorithms for load balancing are:

Min-load: The server with the active scheduler will select the server with the least number of currently running reports. If the local server is one of the qualified servers, it will be given higher priority.

Round Robin: The server with the active scheduler will select each server in sequence one by one until each has been allocated a report to run then will repeat the cycle. When a new scheduler gets a lease, it will pick up the round robin cycle starting from the point it left off when it last held the lease. This is the default setting for JReport Server Live.

Weighted Min-load: The active scheduler will select the server with the least weighted current reports. Report weight is tabulated by dividing the number of currently running reports by the performance weight. If the local server is one of the qualified servers, it will be given higher priority. If the performance weight is not set, the algorithm will default to 1, working the same as min-load since every server will appear to have the same capability.

Random: The server that holds the active scheduler will select the server randomly.

Performance weight is a positive floating point number that can be set to each server in a cluster on any clustered server as a whole. The weight panel in JReport helps measure the performance of a typical report on each node of the cluster. The higher performance weight set to a clustered server, the higher chance it will get selected by the server holding the active scheduler during load balancing.

Failover

Status of the clustered servers can be checked with JReport Monitor so that failure of any member server does not remain unnoticed as JReport Monitor checks the server status periodically via RMI. If a member server is down for any reason, JReport Cluster will automatically remove it from the active clustered server list.

Effect on Load Balancing

When JReport Cluster detects a failed member server it will remove the member server from the active server list and will no longer schedule reporting tasks to that server. Load balancing will proceed, distributing between the remaining active servers.

Effect on Incomplete Tasks

When JReport Cluster detects a failed clustered server, it will check the shared table for the list of incomplete tasks, and automatically reassign all incomplete tasks to other active servers using the load balancer.

Effect on Completed Tasks

JReport supports report level recovery, not session level recovery meaning that once a report task is completed, it will be written to temporary storage for redirection to the requester. Failure after that will not be recovered.

Distributed Resource Storage

In a pure distributed cluster the resource files are not stored in a central place. Each server has its own directories to be able to hold all types of resources including catalogs, templates, dashboards and report results. The sharing of resources among servers is achieved via a copying mechanism. Having more copies of resources reduces the possibility of resources being lost if a server goes down. However, making too many copies consumes system resources by making copies that are never used.

There are two different copying scenarios in JReport Server Cluster. One is that when a resource is saved, JReport will copy it to a number of other randomly selected servers within the cluster based on number of copies the cluster is configured to make. The other is on-demand. When a resource does not exist on the local server, JReport will search among the other servers for the resource and then copy it to the local server.

Several types of resources support distribution including history resources which are resources saved into the versioning system, realm resources, which are cached report results, and memory storage related resources. The number of copies can be set in each type of resource.

The best number to use for each resource depends on two factors, how many points of simultaneous failure need to be tolerated and how often resources (that are not local) are requested by users. For example, 2 points of failure may need to be covered, meaning that 3 copies would be required. However, if there are many on-demand requests, tolerating more points of failure will provide better performance since users will not need to wait for resources, as it is more likely the resource will be available on the local server.

Summary

JReport Server Cluster has all the capabilities and flexibility needed by enterprise applications to provide scalability and reliability to match all other components of the application such as application servers and DBMS systems. Using JReport will ensure that users can always access the data that they need and will never reach a bottleneck or failure point in their application. Unlike other tools, JReport guarantees that if the DBMS and the application are running, reporting will be

available 100% of the time, ensuring that users can depend on their BI tool to have the same reliability built into it as the rest of the application.

More Information

[For More Information about JReport Clustering Contact Us Now](#)

Technical Requirements

Hardware/OS Requirements:

- Intel P4 Xeon 3.0 GHz
- Windows, Unix, Linux, z/Linux
- 2GB of RAM minimum
- 1GB of available hard disk space
- Java VM JDK 6 or above

Web Browser Support:

- Internet Explorer 9 and above
- Google Chrome 23 and above
- Firefox 20 and above



Copyright © 2015 Jinfonet Software, Inc.

Jinfonet Software
2275 Research Blvd
Rockville, MD 20850

Tel: +1 240-477-1000
Web: www.jinfonet.com