High Availability

i2 High Availability™ is a solution component designed to provide mission-critical enterprise applications with a software-based, fault-tolerant, nearly 24/7 operational capability. i2 High Availability brings applications close to 100 percent availability by managing multiple copies of an application across multiple hardware servers. Based on the five key design principles of load balancing, fault tolerance, backup and restore, catch-up, and system monitoring, i2 High Availability provides this service with minimal performance impact to a host application. i2 High Availability has been deployed with i2 Demand Fulfillment™ with strong successes across many industries.

Solution Capabilities

Fault Tolerance

Fault tolerance enables uninterrupted transaction processing, in the event of hardware or software failure. i2 High Availability achieves fault tolerance by managing multiple "hot standby" instances of the hosting application. It maintains a number of backup systems so that if a hardware or software component stops working for any reason, another system is available immediately to take over the failed system's processing. It also ensures that these backup systems have current data, so they can take over seamlessly from the failed components. i2 High Availability also has fault tolerance functionality for responding to failures in the network and database. When the network and database come up again, the i2 High Availability components work as usual, without shutting down in between.

Load Balancing

i2 High Availability improves overall system performance by balancing processing requests across the primary i2 High Availability server and the multiple hot standby secondary backup servers. The processing load is balanced in i2 High Availability by differentiating between state-changing and non state-changing requests. A state-changing request (called an order in i2 Demand Fulfillment) is a transaction that changes the state of the hosting application's data. For example, in i2 Demand Fulfillment, a state-changing transaction might consume some quantity of a product in the application.

State-changing transactions typically include transactions that add, delete, modify, or update the application data. A non state-changing request (called an inquiry or quote in i2 Demand Fulfillment) is a transaction that does not change the state of the hosting application's data. In i2 Demand Fulfillment, for example, a non state-changing transaction might ask how much quantity of a given item is available.

Systems typically handle far more non state-changing transactions than state-changing transactions. To balance the processing load, i2 High Availability detects whether the transaction is state-changing or non state-changing. If the transaction is state-changing, i2 High Availability directs it to the primary

Supported Platforms

Server

- AIX 5.3/6.1
- HP 11.23/11.31
- Solaris 10
- Windows 2003

Databases

- DB2 9 Viper 2 9.5
- Oracle 11gR1

server for processing; if the transaction is non state-changing, i2 High Availability directs it to a secondary server for processing. The load of the non state-changing transactions can be further shared by starting more secondary servers.

Backup and Restore

By capturing all transactions in real time and by providing the "restore" feature, i2 High Availability is different from typical backup procedures. For example, to get a current backup, i2 High Availability takes the latest plan and combines it with the transaction history to restore the application. This backup and restore process is user-driven. A user can choose to restore to any point in time. This would be of use, for instance, when an abnormally large order was promised and then later cancelled. The restored state could be driven back to the live environment/test bed for "what-if" analysis.

Certified or Guaranteed Messaging

i2 High Availability guarantees data delivery between different components of the system, a method that is referred to as "certified messaging." The certified messaging functionality in i2 High Availability helps to ensure that transactions are not missed out by any i2 High Availability component. The functionality is critical in maintaining an identical data picture among the primary and secondary servers, and in ensuring that all of the messages received from the client applications are handled correctly and in the correct order.

Zero Downtime Plan Cycle Swap (Catch-Up)

i2 High Availability provides a mechanism to ensure that the hosting application's data is current and up-to-date without making it unavailable. Since i2 High Availability uses a working copy of the hosting application's data that is stored in its memory, the data available to i2 High Availability does not automatically reflect updates to the hosting application's actual data. The hosting application's data may be updated, for example, when new available-to-promise data (ATP data) is sent from a supply planning application en masse. The hosting application's data is updated in i2 High Availability through the catch-up process. In the catch-up process, a new instance of the hosting application is brought up with the new set of data (e.g., the ATP data in i2 Demand Fulfillment).

While this new data picture is being built in, the current i2 High Availability server continues to process incoming requests. When the new instance of the hosting application is ready with the new data, i2 High Availability stops using the previous instance and begins to use the new instance to process transactions. The catch-up process ensures that the system requires no downtime to update the hosting application's data. The catch-up process is necessary in systems where the hosting application's data is updated periodically from sources other than the transactions that i2 High Availability manages. For example, the i2 Demand Fulfillment server might get new data daily or weekly from a supply planning process.

Versioning

When a software version needs to be tested or implemented, i2 High Availability is a great way to support this process. For testing, the test application can be connected to the online cluster of servers with a low priority for fault tolerance. This ensures that the test version is not likely to become live due to a failure elsewhere. Results of this plan can be compared to the primary plan results, ensuring the new version is functioning as planned. This comparison can be done on a plan-versus-plan basis, or to evaluate how two versions handled specific transactions. Similarly, a new version can be implemented in the method described above. In this case, the priority of the new version should be increased so that it can be used as the primary application. In this way, a user can gradually convert servers from one version to another without loss of customer service.

System Architecture

Client Application

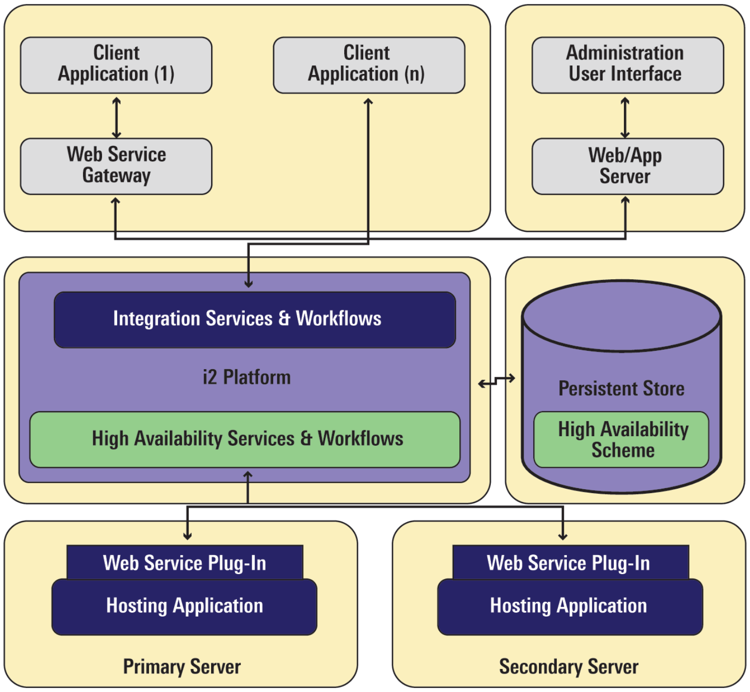
The client application sends state-changing and non state-changing transactions to the hosting application and receives the resulting responses. The client application is typically an order entry system (OES) or order management system (OMS) for i2 Demand Fulfillment, and is sometimes referred to as an order entry system. i2 Demand Fulfillment deployments commonly have i2 Customer Order Fulfillment™, Oracle applications, SAP R/3 or SAP ECC, or IBM Websphere MQSeries based client applications.

The client application may publish web service APIs either directly or via a web service gateway onto i2 High Availability. Gateway applications may have only simple pass-through functionality that manages sending and receiving requests to and from i2 High Availability, or they may perform complex validation and conversion processes on the requests before they send them to the hosting application. Gateways may also be referred to as bridge applications, or as i2 High Availability clients. There are no restrictions on the number of connecting client applications or the number of users on the client application.

Hosting Application

The hosting application is the program that is integrated with i2 High Availability services. The hosting application subscribes and publishes messages via a web service plug-in to take advantage of the i2 High Availability functionality exposed via services. The most common i2 High Availability hosting applications are i2 Demand Fulfillment and i2 Configuration Order Promising.™ However, virtually any application can host i2 High Availability by calling procedures in i2 High Availability's core services and workflows.

i2 High Availability manages a system comprised of a master primary system with one or more hot standbys and a persistent database.



Primary Server

The primary server is the instance of the hosting application that handles the state-changing transactions (e.g., orders, etc.) that is sent by the client application. The primary server also re-sends the transactions to the secondary servers for replication and writes them to the database. If a company doesn't want to run multiple servers or have a catch-up plan swap for a deployment, i2 High Availability can be configured to run with just a single, primary i2 High Availability server.

Secondary Server(s)

The secondary servers are hot standby servers for the primary server; that is, if the primary server fails, one of the (elected) secondary servers can immediately take over normal processing. Secondary servers also minimize the processing load for the primary server by responding to non state-changing transactions (e.g., inquiry). However, to ensure that the secondary server always has up-to-date data, the primary server sends the state-changing transactions that it has processed to the secondary servers. The secondary servers normally run on separate physical machines from the primary server, which provides a measure of fault tolerance and more evenly distributes the processing load among the available hardware. However, multiple servers can be configured to use the same physical hardware.

i2 Platform

The i2 platform is a logical web services-enabled platform used to host i2 High Availability services and workflows in order to route, relay, replicate, recover, and persist data among the i2 High Availability servers and other components in a distributed environment, as well as the integration services and workflows between the client application and the i2 High Availability servers. A single cluster (one primary server with one or more secondary servers) of i2 High Availability may be configured to use multiple instances of the i2 platform if desired. It is also possible to configure a single instance of the i2 platform to support multiple i2 High Availability clusters.

High Availability Schema

The i2 High Availability schema stores transaction history for the current planning cycle. The state-changing requests that i2 High Availability has received in the current planning cycle are stored in the database, providing an up-to-date record of the transactions that have been processed. If necessary, a current data set can be recovered by re-running the transactions stored in the database. Multiple parallel or logical database servers can be configured for a higher level of availability in the event of database server or network failures.

Administration User Interface

i2 High Availability includes a web-based administration user interface (via a web/app server) that monitors and reports statistics on the plan cycle that is currently online, including the number of active servers, server roles, transactions processed, as well as servers scheduled for plan cycle swaps. Critical events such as server failures are reported on the user interface, and it is possible to configure alerts to corporate monitoring tools, as well as recovery actions to restore server availability.

For more information on i2 High Availability, i2 Demand Fulfillment, and other i2 solutions, visit www.i2.com.

The Supply Chain Company®

11701 Luna Road
Dallas, Texas 75234, USA
Phone 1.877.926.9286
Email info@i2.com
Web **www.i2.com**