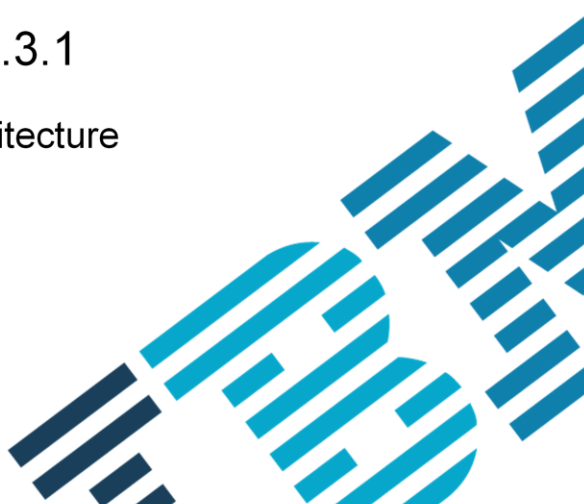


IBM Tivoli Netcool/OMNIbus V7.3.1

Introduction to the multitier system architecture

© 2014 IBM Corporation



IBM Tivoli® Netcool/OMNIbus version 7.3.1 helps with performance and high availability through a multitier system architecture. The architecture has three layers: the collection layer, the aggregation layer, and the display layer. In this training module, you learn about this multitier system architecture and the components in each of the three layers.

Objectives

After you complete this training module, you can accomplish these tasks:

- Recognize the subsystem components in each layer of an IBM Tivoli Netcool/OMNIBus V7.3.1 multitier system architecture
- Plan an IBM Tivoli Netcool/OMNIBus V7.3.1 multitier system architecture implementation
- Describe the function of each multitier system architecture layer
- Follow the flow of event messages in a multitier system architecture

After you complete this training, you can plan an implementation, understand the function of each multitier layer, and check the flow of event messages within this architecture.

The three layers of the multitier system architecture

The IBM Tivoli Netcool/OMNIBus V7.3.1 multitier system architecture has three layers:

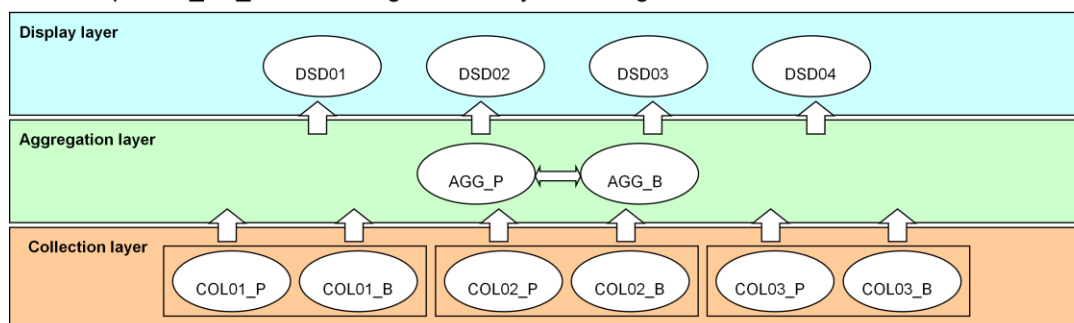
- **Collection layer:** For consolidating and deduplicating events before they are forwarded to the aggregation layer for event processing
- **Aggregation layer:** For managing the incoming events, enriching data, and interacting with users, if required
- **Display layer:** For load balancing user interaction with the event data; or for the displaying of specific subsets of event data to specific users

The multitier system architecture of IBM Tivoli Netcool/OMNIBus version 7.3.1 is the successor to the Event System Framework in version 7.3.0. If you plan to migrate an Event System Framework system to the multitier system architecture, extract any Event System Framework customizations and apply those customizations to the multitier system architecture. On this slide, you see descriptions of the three layers of the architecture.

Block diagram of the display, aggregation, and collection layers

For more information about multitier high availability, go to this website:

http://www-01.ibm.com/support/knowledgecenter/SSSHTQ_7.3.1/com.ibm.netcool_OMNibus.doc_7.3.1/omnibus/wip/install/concept/omn_ins_multitieredhighavailability.html?lang=en



4

Introduction to the multitier system architecture

© 2014 IBM Corporation

In this diagram, the colors orange, green, and blue identify the three system layers. The collection layer is shown in orange. It has one or more ObjectServer pairs, which are connected to the aggregation layer through unidirectional gateways. The aggregation layer is shown in green. It has a dual and resilient ObjectServer pair. This pair is connected through a bidirectional gateway. The display layer is shown in blue. It comes from the aggregation layer by using a unidirectional gateway to each display layer ObjectServer. One or more display ObjectServers are in IBM Tivoli Netcool/OMNibus multitier system architectures. Primary and backup ObjectServers must adhere to the syntax `_P` for primary ObjectServers, and `_B` for backup ObjectServers. For more information about multitier high availability, see the URL shown on the slide.

Collection layer overview

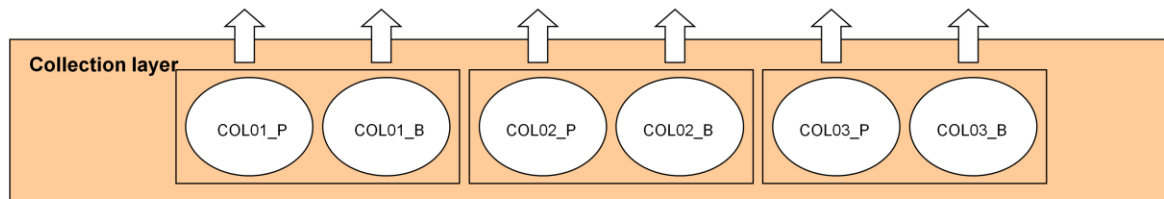
The purpose of the collection layer is to accelerate event processing

- Events are forwarded to and deduplicated at the collection layer
 - Events can be held at the collection layer to reduce aggregation layer loading
 - Minimal event processing is performed at the collection layer
 - Events are expired at the collection layer
- Common collection layer clients are as follows:
 - Probes
 - Monitors
 - IBM Tivoli Monitoring
 - Tivoli Enterprise Console®

The collection layer accelerates event processing. Events are both collected and deduplicated in the collection layer. Although minimal event message processing is performed at the collection layer, event messages can be held at the collection layer to prevent overloading on the upper aggregation layer. The configuration setting that expires event messages is at the collection layer. Some of the more common clients that provide information to the collection layer are probes, monitors, IBM Tivoli Monitoring, and the Tivoli Enterprise Console.

Components of the collection layer

- The collection layer contains pairs of ObjectServers
- A unidirectional gateway is connected to each ObjectServer and is used to forward events to the upper level aggregation layer



The collection layer has pairs of ObjectServers with a unidirectional gateway connected to each pair. The unidirectional gateways are used to forward events to the aggregation layer.

Aggregation layer overview

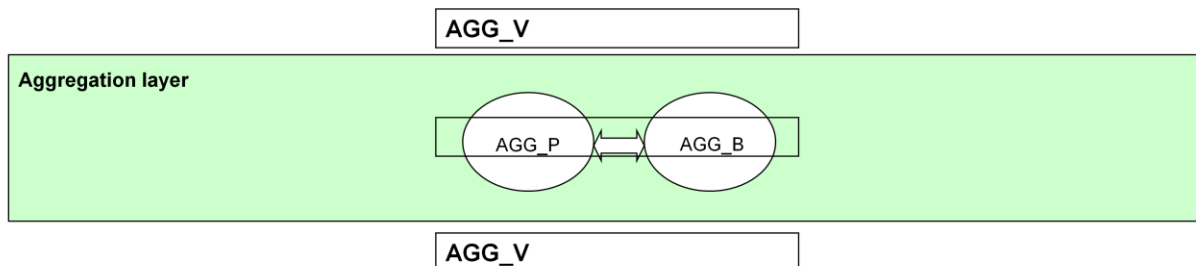
The purpose of the aggregation layer is to correlate and enhance events

- The aggregation layer gets event messages from the collection layer
- Events are processed and enhanced at the aggregation layer by using these components:
 - Triggers
 - Generic Clear triggers
 - Integration triggers
 - Custom triggers
 - Tivoli Netcool/Impact
 - Historical gateways
 - Ticketing gateways

In the aggregation layer, events are correlated and enhanced through various components, as listed on the slide. These components, which come from the collection layer, are used to process and enhance event messages. You must ensure that only one aggregation ObjectServer has active primary-only triggers and that you use the correct ObjectServer nomenclature. You must also ensure that the ObjectServer properties BackupObjectServer and ActingPrimary are set correctly after deployment.

Components of the aggregation layer

The aggregation layer contains a dual resilient ObjectServer pair, which is synchronized by using a bidirectional gateway



The aggregation ObjectServers can be connected to a single virtual ObjectServer, such as AGG_V, to provide automatic failover

For probes, you must use the Server and ServerBackup properties, rather than the virtual ObjectServer, AGG_V. Using these properties ensures ObjectServer failover and failback, when PollServer and NetworkTimeout are set correctly.

Display layer overview

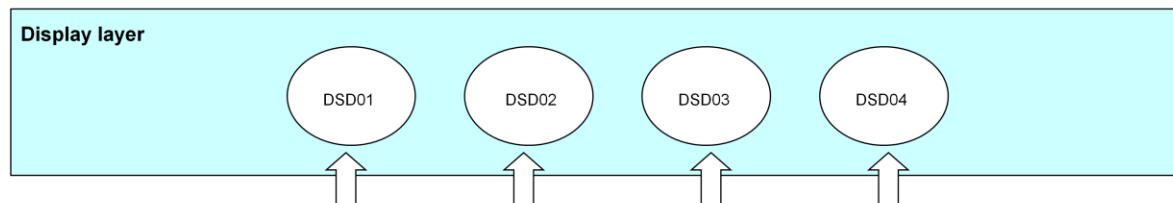
Users can view the processed events in the display layer

- Primary users are IBM Tivoli Netcool/OMNibus web GUI users and IBM Tivoli Netcool/OMNibus event list users
- Other users are as follows:
 - Users of IBM product integrations, such as IBM Tivoli Monitoring or Tivoli Enterprise Console
 - Custom integration users
 - Gateway users, such as Tivoli Service Request Manager® ticket gateways or JDBC gateways
 - Administrators (nco_config)

The display layer provides network event information to its users. The main users are IBM Tivoli Netcool/OMNibus web GUI and event list users. Other users include IBM Tivoli Monitoring users, Tivoli Service Request Manager ticket gateway users, and users of custom network integrations with IBM Tivoli Netcool/OMNibus version 7.3.1.

Components of the display layer

- The display layer contains one or more ObjectServers
- Each display layer ObjectServer is connected to the aggregation layer through unidirectional gateways



The display layer is made from one or more ObjectServers, with each ObjectServer connected to the aggregation layer through a unidirectional gateway.

Multitier system architecture event data flow overview

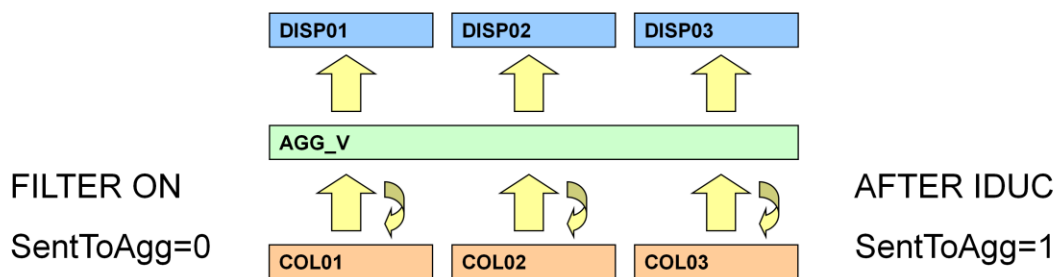
- Events are inserted at the collection layer
 - The ObjectServer field SentToAgg controls which events are forwarded to the aggregation layer from the alerts.status table
- All events from the aggregation layer are forwarded to the display layer, and by default, are set as follows:
 - The display layer is in a dual-write configuration
 - The display layer contains a copy of the events in the aggregation layer
- By default, the content of these three dynamic tables is forwarded up the layers through the ObjectServer gateways:
 - alerts.status
 - alerts.journal
 - alerts.details
- An event is defined by the Identifier, ServerName, and ServerSerial parameters
 - The Identifier must be unique within an ObjectServer
 - The ServerName and ServerSerial must be consistent throughout the life of the event within the multitier system

Events are first inserted at the collection layer. They are then forwarded from the collection layer to the aggregation layer by aggregation layer gateway ObjectServers. This stage of event data flow is controlled by the ObjectServer field named SentToAgg. From the aggregation layer, events are forwarded by default to the display layer. The content of three dynamic tables, alerts.status, alerts.journal, and alerts.details, is forwarded up through the layers of the architecture. Events are defined by three parameters: Identifier, ServerName, and ServerSerial. Each identifier must be unique within an ObjectServer. For an individual event, its ServerName and ServerSerial remain the same throughout the life of that event.

Multitier system architecture event data flow diagram

The following fields control the display and expiration of events:

- **SentToAgg**: New events (value = 0) distinguished from existing events (value = 1)
- **CollectionExpireTime**: Controls the time of event expiration at the collection layer



This diagram highlights event display deduplication. New events have the SentToAgg flag set to zero. These events are forwarded from the collection layer to the aggregation layer for processing. Events and updates can flow freely to the display layer. The SentToAgg flag must be set to zero for these events and updates to flow up from the aggregation layer. An administrator can then control which data is forwarded after the events are inserted. Events are expired at the collection layer by using the field CollectionExpireTime. This field must be set to a value, such as 300 or 600. The purpose of CollectionExpireTime is to prevent unnecessary reinserts into the aggregation layer and ensure that unique events are passed to ticket and historical gateways correctly. The value of CollectionExpireTime is set to a value that best represents the desired lifetime of an event in the system.

Summary

Now that you completed this training module, you can accomplish these tasks:

- Recognize the subsystem components in each layer of an IBM Tivoli Netcool/OMNIBus V7.3.1 multitier system architecture
- Plan an IBM Tivoli Netcool/OMNIBus V7.3.1 multitier system architecture implementation
- Describe the function of each multitier system architecture layer
- Follow the flow of event messages in a multitier system architecture

Now that you completed this training, you can describe the function of each IBM Tivoli Netcool/OMNIBus multitier system architecture layer. You can recognize the subsystem components, plan an implementation, and check the flow of event messages in the system architecture.

Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, Service Request Manager, Tivoli, and Tivoli Enterprise Console are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2014. All rights reserved.