

IBM Spectrum Symphony RFE 67907 Readme File

About consumer level exclusive

With this feature, you can configure exclusive consumers so that allocations under the same exclusive consumer can share a host, and allocations under different exclusive consumers cannot share a host.

Readme file for: IBM® Spectrum Symphony

Product/Component Release: 7.1.2

Fix ID: sym-7.1.2-build411550-jpmc

Publication date: 15 July 2016

Last modified date: 15 July 2016

1. Scope.....	3
2. Installation	3
1. Prerequisites	3
2. Packages.....	3
3. Installation procedure	3
3. Configuration	4
1. Enable this feature.....	4
2. Start the cluster	4
3. Log on to the cluster management console, and configure exclusive consumers.....	4
4. Usage	5
1. How this feature works.....	5
2. Feature interactions	5
3. Best practices	6
5. Verification	7
6. Uninstalling	7
1. Stop the cluster.....	7
2. Restore the backed-up files	7
3. Start the cluster and enable the application	7
Copyright and trademark information	7

1. Scope

Applicability	
Operating system	RHEL 6 or 7, 64-bit
Product version	IBM Spectrum Symphony 7.1.2

2. Installation

1. Prerequisites

Before applying this fix, you must have IBM Spectrum Symphony 7.1.2 installed.

2. Packages

File name	Description
<code>sym-7.1.2.0_x86_64_build411550.tar.gz</code>	The package that contains this feature.

3. Installation procedure

a. Stop the cluster. Log on to the master host as the cluster administrator and run:

```
$ soamcontrol app disable all  
$ egosh service stop all  
$ egosh ego shutdown all
```

b. Back up the following files on all management hosts:

```
$EGO_TOP/3.4/linux-x86_64/etc/vemkd  
$EGO_TOP/3.4/linux-x86_64/lib/policy/libslot.so  
$EGO_TOP/wlp/usr/servers/gui/apps/ego/3.4/platform/js/ResourcePlanDom.  
js
```

c. Clean up the GUI work directories on all management hosts:

```
$rm -rf $EGO_TOP/gui/work/*  
$rm -rf $EGO_TOP/gui/workarea/*
```

d. Clean up the browser cache on all client hosts.

e. Copy the `sym-7.1.2.0_x86_64_build411550.tar.gz` file to the `$EGO_TOP` directory on all management hosts, and decompress the package.

3. Configuration

1. Enable this feature

Add the following line into the `$EGO_CONFDIR/ego.conf` file:

```
EGO_ENABLE_CONSUMER_LEVEL_EXCLUSIVE=Y
```

2. Start the cluster

Run the following command:

```
egosh ego start all
```

3. Log on to the cluster management console, and configure exclusive consumers

- Open the page for the resource plan (**Resources > Resource Planning (Slot) > Resource Plan**), and select the preferred resource group.
- Expand the **Slot allocation policy** section, and select the **Hybrid policy**, and then **Exclusive > Consumer level**.

Exclusive Consumer will be enabled, as shown in the following screenshot. You can then configure exclusive consumers.

For an exclusive consumer, all of its ancestor consumers must also be exclusive. On the management console, if you select a consumer as exclusive, the console will also automatically select all of its ancestor consumers as exclusive. However, if you deselect a consumer to make it non-exclusive, the system can only deselect all visible child consumers and mark them non-exclusive. As a best practice, select **Expand All** and then deselect the consumer to mark it as non-exclusive.

The screenshot shows the 'Resource Plan' configuration page. The 'Resource Group' is set to 'ComputeHosts'. Under the 'Slot allocation policy' section, the 'Hybrid policy' is selected. Within the 'Hybrid policy' section, the 'Consumer level: each host is only assigned to allocations under the same exclusive consumer' option is selected and highlighted with a green box. Below this, a table displays the configuration for consumers under the 'Ownership' and 'Hybrid' model types. The 'clusterghsui' consumer is expanded, showing 'SymTesting' and 'Symping711' as child consumers. The 'Exclusive Consumer' column has a checked checkbox for 'SymTesting', also highlighted with a green box.

		Model type: Ownership		Model type: Hybrid				
Consumer	Exclusive Consumer	Owned Slots	Consumer Rank	Lend Limit	Borrow Limit	Owned Slots	Reserve	Limit
clusterghsui		16				check all 16		
SymTesting	<input checked="" type="checkbox"/>	0	0			<input checked="" type="checkbox"/> 8		
Symping711	<input type="checkbox"/>	0	0			<input checked="" type="checkbox"/> 4		

Note: Currently, the **Hybrid** model type is supported. Do not configure on the **Ownership** model type.

4. Usage

1. How this feature works

With exclusive consumers, the leaf consumers are divided into several consumer groups. A leaf consumer with an exclusive flag belongs to the consumer group for itself. A leaf consumer without an exclusive flag, if it has one or more exclusive ancestor consumers, belongs to the consumer group for the nearest exclusive ancestor consumer. A leaf consumer without an exclusive flag, if it has no exclusive ancestor consumers, belongs to the consumer group that is owned by the root consumer.

The corresponding allocations for the leaf consumers are divided into several allocation groups by the consumer groups.

When assigning slots to an allocation, the hosts already used by the allocations in the same allocation group will be selected first. Next, the hosts with all its slots are free will be selected. If an allocation cannot be satisfied with these criteria, a reclaim may be triggered. For the selected hosts in each step, there is no order among them in that step.

2. Feature interactions

- CLI

With the CLI, you can see there are still some free slots in the resource group, but new workload cannot run with these slots. This is normal behavior because a host is allocated to an allocation group exclusively, and some of its slots will not be assigned workload to run if there is not enough workload in that allocation group. These slots are free in the output of the CLI, but these slots cannot be used by an allocation from another allocation group.

For example, take the scenario where there is one host with eight slots in a resource group, and there is an allocation that only needs six slots. Running **egosh rg** shows there are two free slots in the resource group, but these free slots cannot be used by an allocation from another allocation group.

- resReq

If the resReq parameter is configured in the application profile, the corresponding allocation may not get the preferred resources if the resources that are already assigned to its allocation group cannot meet the resReq, and if this allocation group cannot get more resources because it is satisfied or over-used.

To ensure that the allocation can get its preferred resources, configure the same resReq value for the allocations in the same allocation group.

- Data-aware scheduling (data affinity)

The data-aware scheduling feature is not impacted by this feature.

- Selective reclaim

Selective reclaim cannot work on a resource group with this feature enabled. Instead, a warning message will be logged in the VEMKD log when the selective reclaim feature and this feature are both enabled.

- EGO_DISABLE_RECLAIM_HYBRID_OWN

The EGO_DISABLE_RECLAIM_HYBRID_OWN parameter is configured in the \$EGO_CONFDIR/ego.conf file. This parameter will not work with this feature when the share ratio is broken.

For example, take a cluster with four hosts, each host with four slots. Consumer A and B are in different allocation groups, and the share ratio for them is 1:1. The own value for consumer A is four.

Scenario 1: consumer A gets all the slots, and then releases some slots, and it gets one slot per host at last, for a total of four slots.

Scenario 2: Consumer B has demands, based on the share ratio, it can get two hosts, so it will reclaim two hosts from consumer A. At this time, although the owned slots of consumer A is four, the own value is also four, but the group for it occupies 16 slots (four slots times four hosts). It will be reclaimed.

For scenario 2, the parameter EGO_DISABLE_RECLAIM_HYBRID_OWN cannot protect consumer A because the share ratio will be considered first.

- Reserve

This parameter will not work with this feature when the share ratio is broken.

This issue may happen when there are several allocations in the same allocation group and this group can borrow slots.

For example, there are two hosts (hostA, hostB) in the cluster, each with eight slots, with a total of 16 slots. There are two consumer (a21 and a22) and they are in the same allocation group. The **Reserve** value for any of them is one, as show in following screenshot:

		24.00						
		Model type: Ownership			Model type: Hybrid			
Consumer	Exclusive Consumer	Owned Slots	Consumer Rank	Lend Limit	Borrow Limit	Owned Slots	Reserve	Limit
clusterghsui		16				check all 16		
a	<input checked="" type="checkbox"/>	0	0			<input checked="" type="checkbox"/> 8	2	
a11	<input checked="" type="checkbox"/>	0	0			<input checked="" type="checkbox"/> 8	2	
a21	<input type="checkbox"/>	0	0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> 4	1	
a22	<input type="checkbox"/>	0	0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> 4	1	

When there is only workload for consumers a21 and a22, they can use the two hosts, so a21 has one task on hostA, and a22 has one task on hostB.

If there is more workload from other consumers, a21 and a22 will have to give out one host (either hostA or hostB) because they can only own one host based the share ratio. Giving out any host will break the reserve value (that is, a21 or a22 will have no slots for several seconds while the reserve value is one). Although the reclaimed node can get its reserved slots from the remaining host later on, the reservation is broken for a while.

3. Best practices

- Place sibling nodes in the same allocation group by setting their parent node with the exclusive flag, or place them in different groups by setting every node with an exclusive flag.

If some sibling nodes are in one allocation group, and others are in another one, when there is a reclaim between sibling nodes, an under-used or satisfied allocation in the over-used group may be reclaimed.

For example, take the scenario where there are three leaf nodes: a11, a12, and a13. The share ratio is 1:1:1; a11 and a12 are in the same allocation group, whereas a13 is in another allocation group. There are three hosts in the cluster. If a11 gets one slot from each host for a total three slots, and a12 gets the remaining slots on the three hosts, there will be no free slots. When there is a demand on a13, a13 should get one host based on its share ratio. When a13 reclaims any hosts, a11 will be reclaimed which is not over-used based on the share ratio.

5. Verification

For exclusive consumers, the allocations under the same exclusive consumer can share a host, and the allocations under different exclusive consumers cannot share a host.

6. Uninstalling

1. Stop the cluster

Log on to the master host as the cluster administrator and run:

```
$ soamcontrol app disable all
$ egosh service stop all
$ egosh ego shutdown all
```

2. Restore the backed-up files

3. Start the cluster and enable the application

Log on to the master host as the cluster administrator and run:

```
$ egosh ego start all
$ soamcontrol app enable <appName>
```

Copyright and trademark information

© Copyright IBM Corporation 1992, 2016.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

IBM Web site pages might contain other proprietary notices and copyright information that should be observed.