

# IBM Platform Symphony RFC 4420 Readme File

## About SSM balance allocation between hosts feature

This package contains the **SSM balance allocation between hosts** feature.

The **SSM balance allocation between hosts** feature is based on Pre-start feature and MinService policy. With the pre-start feature, the SSM requests slots and starts SI's as soon as SSM is enabled, and all SI's must be ready before submitting a session for this feature. When the MinService policy allocates the slots to the sessions, it will first select the host that has the most number of idle slots in its list before tasks are dispatched to it.

**Readme file for:** IBM® Platform Symphony

**Product/Component Release:** 6.1.0.1

**Update Name:** Fix pack

**Fix ID:** sym-6.1.0.1-build229959-nudt

**Publication date:** 17th Mar 2014

**Last modified date:** 13th Mar 2014

1.	Scope.....	3
2.	Installation.....	4
	1. Prerequisites.....	4
	2. Packages.....	4
	3. Installation procedure: Symphony production cluster on Linux x86_64 RHEL 6.x.....	4
3.	Configuration.....	5
	1. Prerequisites.....	5
	2. Related Settings.....	5
4.	Usage.....	6
	1. How these features work.....	6
	2. Examples.....	7
5.	Note.....	8
6.	Copyright and trademark information.....	9

# 1. Scope

Applicability	
Operating system	Linux x86_64 RHEL 6.3
Symphony version	Symphony 6.1.0.1
Limitations	
<Limitation>	<ul style="list-style-type: none"><li>• Only supports MinService policy</li><li>• Does not support non-prestart mode</li><li>• Does not support DAS feature.</li></ul>

## 2. Installation

### 1. Prerequisites

1. Symphony 6.1.0.1 is installed.

### 2. Packages

File name	Description
symphony_build229959_linux2.6-glibc2.3-x86_64.tar.gz	The binary package that contains this new feature for Linux x86_64 RHEL 6.x.
checksum.md5	Contains the MD5 checksum for symphony_build229959_linux2.6-glibc2.3-x86_64.tar.gz

### 3. Installation procedure: Symphony production cluster on Linux x86\_64 RHEL 6.x

1. Download "symphony\_build229959\_linux2.6-glibc2.3-x86\_64.tar.gz".
2. Shut down all applications using the following commands:  

```
[lsfadmin@mghost ~]$soamcontrol app disable all -f
```
3. (OPTIONAL) Back up SOAM binaries. Use this step if you want to be able to roll back the change.  
Go to `SOAM_HOME/6.1/linux2.6-glibc2.3-x86_64/etc/` directory and back them up.
4. Copy `symphony_build229959_linux2.6-glibc2.3-x86_64.tar.gz` to `/tmp/RFC4420` directory and decompress the package.
5. Copy `/tmp/RFC4420/ssm` to `SOAM_HOME/6.1/linux2.6-glibc2.3-x86_64/etc/`.

### 3. Configuration

#### 1. Prerequisites

Symphony 6.1.0.1 is installed.

#### 2. Related Settings

Add environment variable to SSM section of application profile to enable the feature:

***SOAM\_SSM\_BALANCE\_ALLOCATION\_BETWEEN\_HOSTS = Y/N***

***N/n***: the feature is off

***Y/y***: SSM will dispatch the task to the compute host with the most idle slots percentage. As a result, the tasks are balanced in the cluster.

For example:

```
<SSM startUpTimeout="60" shutDownTimeout="300">
.....
<osTypes>
  <osType name="X86_64" .....>
    <env name="SOAM_SSM_BALANCE_ALLOCATION_BETWEEN_HOSTS">Y</env>
  </osType>
</osTypes>
.....
</SSM>
```

If the environment variable is not defined, the feature is off.

## 4. Usage

### 1. How these features work

The **SSM balance allocation between hosts** feature is based on the Pre-start feature and MinService policy. With the pre-start feature, the SSM requests slots and starts SI's as soon as the SSM is enabled, and all SI's must be ready before submitting a session for this feature. When the MinService policy allocates the slots to the sessions, it will first select the host that has the most number of idle slots in its list before tasks are dispatched to it.

#### Example 1 to demonstrate tasks balanced between hosts:

##### Pre-condition:

- Pre-start feature is enabled
- MinService policy is used
- 4 slots in total (2 hosts with 2 slots each)

##### Scenario:

- Submit a session with 2 running tasks

##### Post-condition:

- As shown on the left of Figure 1, the two tasks are dispatched to host1 when this feature is disabled; and host2 is idle
- Meanwhile, the tasks are balanced between hosts when this feature is enabled; as shown in right picture of figure 1.

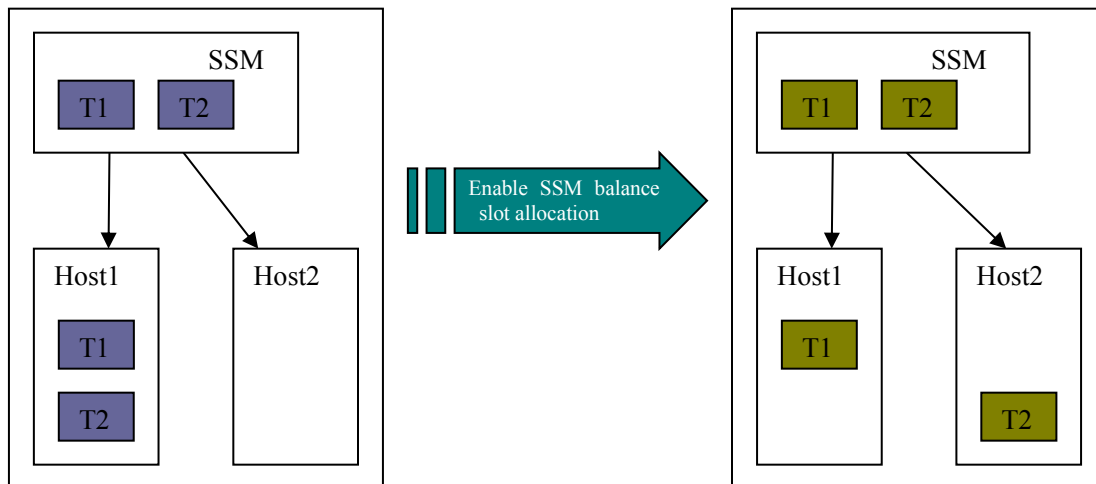


Figure 1: SSM balance slot allocation example

#### Total slots in each host:

When SSM dispatches tasks, the total slot number of the compute host is considered. To measure the extent of idle slots on the host, a percentage named slot idle percentage is introduced (defined in Table 1: slots idle percentage definition). Based on slot idle percentage, the SSM will first select the host that has the highest slots idle percentage to dispatch tasks to. If the slots idle percentage is the same for two hosts, the SSM will select one host randomly.

$$\text{slots idle percentage} = \text{idle slots} / \text{total slots}$$

Table 1: slots idle percentage definition

#### Example 2 to demonstrate tasks balanced between hosts with different total slots:

##### Pre-condition:

- *Pre-start* feature is enabled
- *MinService* policy is used
- 2 hosts; *Host1* has 8 slots, *Host2* has 2 slots

Scenario:

- Submit a session with 5 running tasks

Post-condition:

- As shown on the left of Figure 2, when this feature is disabled, the 5 tasks are dispatched to *Host1*. *Host2* is idle.
- When this feature is enabled, the tasks are balanced between hosts based on *idle slot percentage*; as shown on the right of Figure 2 (the *idle slot percentage* of *Host1* and *Host2* are 50%).

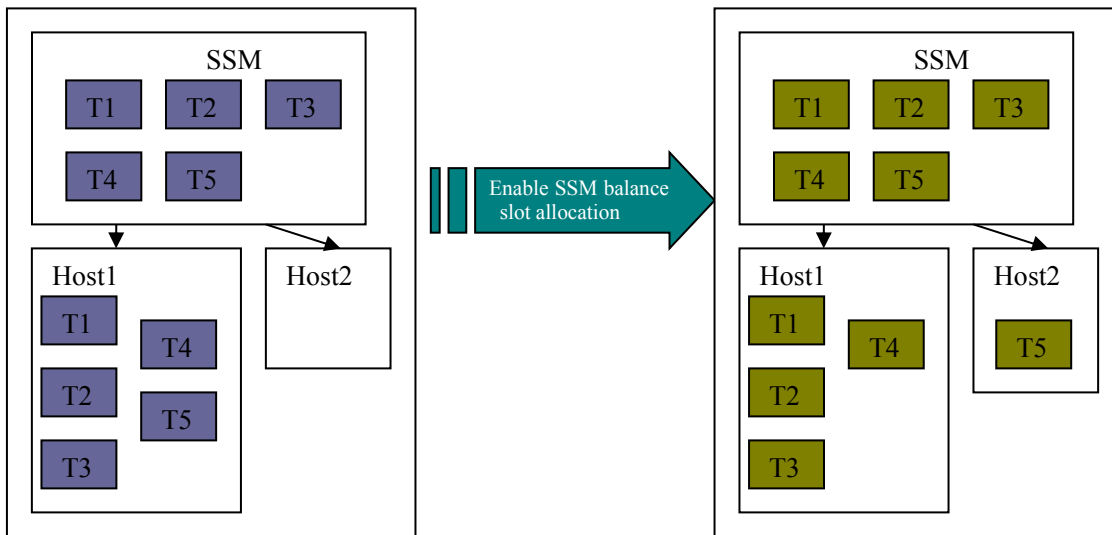


Figure 2: Balanced allocation with different number of slots

**Note:**

The Example 2 is only for application level balance. It just means if the session's minService is 5, SSM will assign 4 SIMs on Host 1 and 1 SIM on Host 2 to the session.

## 2. Examples

### Case 1: Tasks balanced between hosts with the same total number of slots

**Summary:**

1. With this feature enabled, the tasks will be balanced between hosts.

**Pre-condition:**

1. Pre-start feature is enabled.
2. MinService policy is used.
3. 4 slots in total (2 hosts with 2 slots each).

**Scenario:**

1. Submit a session with 2 running tasks.

**Post-condition:**

1. The tasks are balanced between hosts: 1 task for each host.
2. The session completed successfully.

## 5. Note

1. Pre-start feature must be enabled and all slots must be pre-started.
  2. The slots number on each host must be equal.
  3. The session number multiply the minService for each session must be equal the total slots number in the cluster.
  3. When session give up their resources, such as reclaim, preemption, or some tasks finish and session return the resource, the feature will not re-balance the rest running tasks.
  4. About the session level balance, before all the minService slots are bound to the session, the balance result may be imprecise. After all slots are bound, the balance result is precise.
- Because we can't dispatch a task to resources that are not yet "bound", if SIMs on a host(Host1) happen to "bind" faster than SIMs on another host(Host2) and the tasks are already ready to be dispatched, then we will end up with unbalanced workload across the hosts, the tasks may be dispatched to Host 1 first and then Host 2.

## **6. Copyright and trademark information**

© Copyright IBM Corporation 1992, 2014.

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.