



## vDRA Advanced Tuning

---

The configuration values in this document are based on the type and size of DRA deployment.

The deployment can be categorized into two sizes:

- Small Deployment - 2 Directors and 4 workers
- Large Deployment - 8 directors and 10 workers



---

**Note** Recommended values are based on:

- Local Latency:
  - 20 ms for small deployments
  - 25 ms on large deployments
- Remote Latency:
  - 55 ms for small deployments
  - 50 ms for large deployments

In case the latency changes to a higher value, recommended values for all the configurations highlighted in this document needs to be re-characterized again.

---

- [Threading Configuration, on page 2](#)
- [Database Capacity, on page 3](#)
- [Database Alert Expressions, on page 4](#)
- [Database Connection Settings, on page 5](#)
- [Audit Rate Limiter, on page 7](#)
- [Control Plane Tuning Configuration, on page 7](#)
- [IPC Queue Send Thread Tuning Configuration, on page 8](#)

# Threading Configuration

Thread pool configurations are applicable to java processes that run-in diameter-endpoint-s1xx on Director VM and binding-s1xx container on Worker VMs in DRA.

For more information on Threading Configuration, refer to the *CPS vDRA Configuration Guide*.

**Table 1: Thread Pool Names**

Thread Pool Name	Description
bindings	This thread pool is used for IMSI_APN/MSISDN_APN AAR lookup (MOG) and binding audit updates.
bindings.lookup	This thread pool is used only if appsharding.reader thread pool is overloaded. This thread pool is applicable only for AAR VoLTE calls.
bindings.delete	This thread pool is used to query session database and enqueue session and all the binding for deletion.
bindings.response	This thread pool is used to receive database storage response and forwards it as bind store response. This is applicable for CCR-I requests.
msgtimeouts	This thread pool is used to process timed out requests.
qprocessor	This thread pool is used to send routed messages outside on the specific TCP connection.
receivers	Not Applicable
senders	Not Applicable
localcpPublishers	This thread pool is used to publish messages to local control plane.
globalcpPublishers	This thread pool is used to publish messages to global control plane.
cpSubscriberWorker	This thread pool is used to process messages received on local and global control plane.
pcrfQuery	This thread pool is used for PCRF session queries used only for Rx AAR VoLTE calls.
pcrf.restapi	This thread pool is used to send request and receive response to/from pcrf endpoints and used only for RX AAR VoLTE calls.  <b>Note</b> <b>Scale By CPU Core</b> is not applicable for this thread pool.
stackStopHandler	This thread pool is used to gracefully disconnect diameter peers on process shutdown.
appsharding.reader	This thread pool is specific to Rx AAR VoLTE IPv6 binding lookup and processing. This thread pool is created for each shard.

The following tables lists recommended values when single DRA installation TPS is above 200 K.



**Important** The highlighted values must be explicitly configured.

*Table 2: Recommended Values*

Thread Pool Name	Default Values		Recommended Values	
	Threads	Queue Size	Threads	Queue Size
bindings	30	1000	<b>130</b>	<b>30000</b>
bindings. lookup	10	1000	<b>30</b>	<b>5000</b>
bindings. delete	10	1000	<b>100</b>	<b>50000</b>
bindings. response	10	1000	10	1000
msgtimeouts	3	1000	3	1000
qprocessor	30	5000	<b>150</b>	<b>25000</b>
localcpPublishers	15	2000	15	2000
globalcpPublishers	40	3000	40	3000
cpSubscriberWorker	30	10000	30	10000
pcrfQuery	20	400	20	400
pcrf.restapi	10	2000	10	2000
stackStopHandler	50	0	50	0
appsharding. reader	1	1000	<b>2</b>	<b>2000</b>

## Database Capacity



**Note** Rated database capacity has been arrived at after performing capacity tests.

The following table lists the database capacity for small and large deployments.

Table 3: Database Capacity

Deployment	Cluster	No. of Primary Shards (dual DB enabled)	Network Latency (in ms)	Database Capacity (in k)	Database Capacity Per Shard	85% Threshold Value
Small Deployment	Session_IPv6	8	20	30	$30000/8 = 3750$	3200
	IMSI_MSISDN	4	55	15	$15000/4 = 3750$	3200
Large Deployment	Session_IPv6	48	25	180	$180000/48 = 3750$	3200
	IMSI_MSISDN	48	50	145	$145000/48 = 3020$	2600

The following is the test call model used to determine the capacity for each cluster:

1. CCR\_I, CCR\_T and AAR call model for Session-IPv6 database cluster capacity.
2. CCR-I and CCR\_T call model for IMSI-APN/MSISDN-APN database cluster capacity.
3. Database capacity is based on network latency mentioned in the [Table 3: Database Capacity, on page 4](#). If network latency is changed to higher value, then database capacity needs to be re-characterized.

## Database Alert Expressions

Database alerts refer to alerts that can be configured to be triggered if the capacity of the database exceeds beyond a certain threshold.

### IMSI\_MSISDN Cluster

```
alert rule DRA_IMSI_MSISDN_DB_TPS_EXCEEDED
```

```
expression
```

```
"sum(rate(mongo_operation_total{state='primary',type='mongo',op=~'update|query|delete',cluster='IMSI_MSISDN'}[5m]))
> (2500 * sum(mongo_node_state_primary{cluster='IMSI_MSISDN',type='mongo'}))"
```

```
event-host-label instance
```

```
message "{{ $labels.instance }}" Persistence DB TPS exceeded , current value is {{ $value }} !"
```

```
snmp-severity critical
```

```
snmp-clear-message "{{ $labels.instance }}" Persistence DB TPS in control, current value is {{ $value }} !"
```

### Session\_IPv6 Cluster

```
alert rule DRA_SESS_IPV6_DB_TPS_EXCEEDED
```

```
expression
```

```
"sum(rate(mongo_operation_total{state='primary',type='mongo',op=~'update|query|delete',cluster=~'SES_IPV6_*'}[5m]))
> (3200 * sum(mongo_node_state_primary{cluster=~'SES_IPV6_*',type='mongo'}))"
```

```
event-host-label instance
```

```
message "{{ $labels.instance }}" Persistence DB TPS exceeded , current value is {{ $value }} !"
```

```
snmp-severity critical
```

```
snmp-clear-message "{{ $labels.instance }}" Persistence DB TPS in control, current value is {{ $value }}" !"
```

## Database Connection Settings

The following configurations vary based on deployment size (Small or Large) and should be configured accordingly. The worker VM java processes connect to the databases and create two database connection pools based on the database operation:

- Create/Update, and Delete
- Read

For more information on configuration syntax and examples, refer to the *CPS vDRA Operations Guide*.

### binding db-connection-settings

The recommended values listed in the table are for write operations (Create/Update, Delete):



#### Important

The highlighted values must be explicitly configured.

**Table 4: Recommended Value for Write Operations**

Deployment	Bindings	connection-per-host	
		Default Values	Recommended Values
Small Deployment	drasession	10	<b>14</b>
	ipv6	10	10
	imsiapn	10	<b>35</b>
	msisdnapi	10	10
	ipv4	10	10
Large Deployment	drasession	10	10
	ipv6	10	10
	imsiapn	10	<b>20</b>
	msisdnapi	10	<b>20</b>
	ipv4	10	<b>2</b>

**binding db-read-connection-settings**

The recommended values listed in the table are for read operations:



**Important** The highlighted values must be explicitly configured.

**Table 5: Recommended Value for Read Operations**

Deployment	Bindings	connections-per-host-for-read	
		Default Values	Recommended Values
Small Deployment	drasession	5	5
	ipv6	5	5
	imsiapn	5	<b>3</b>
	msisdnapn	5	5
	ipv4	5	5
Large Deployment	drasession	5	<b>20</b>
	ipv6	5	<b>20</b>
	imsiapn	5	<b>10</b>
	msisdnapn	5	<b>10</b>
	ipv4	5	<b>1</b>

**binding cluster-binding-dbs imsiapn-msisdnapn**

The following configuration is applicable only for large deployments. With this configuration IMSI-APN and MSISDN-APN bindings databases use the same connection pool for database transactions. This configuration helps in reducing the overall threads count in the worker VM.

**Table 6: Recommended Values**

Deployment	Recommended Value
Small Deployment	Disable
Large Deployment	Enable

## Audit Rate Limiter

Database Audit is an important functionality in DRA. To ensure that the database does not fill up with unnecessary entries or expired entries are cleared, a house keeping process is always functional. The following parameters should be used based on the deployment size and rated capacity of the system.

**Table 7: Audit Rate Limiter Recommended Values**

Deployment	Rate Limiter	Stale Session Expiry Count	Binding DB Read Preference
Small Deployment	100	6	Nearest
Large Deployment	51	6	Nearest

## Control Plane Tuning Configuration

Director nodes periodically advertise the status of all its peer connections over local and global control plane. All DRA nodes in the network use these peer status messages to keep their peer topology view updated. Peer topology is used to route the messages to appropriate director across different sites. When there are large number of peers connected across different sites, the load on control plane increases. This load increase can cause delay in processing peer status updates.

The following configurations are recommended to handle the increase in the control plane traffic process.

**Table 8: Control Plane Traffic Tuning Commands**

Command	Parameters	Default Values	Recommended Value
<pre>control-plane timers peer-status-update-interval &lt;time-in-ms&gt; peer-expiration-duration &lt;duration-in-ms&gt;</pre> <p><b>Note</b> To reflect the peer expiration duration change, application should be restarted in both director and worker nodes.</p>	<i>time-in-ms,</i> <i>duration-in-ms</i>	2000, 10000	4000, 12000
<pre>control-plane remote-peer-policy mated-system id &lt;system-id&gt;</pre>	<i>system-id</i> (mated pair system ID)	NA	Mated Pair System ID
<pre>control-plane remote-peer-policy global accept diameter-applications [Application Type]</pre>	<i>Application Type</i>	All	Rx

Command	Parameters	Default Values	Recommended Value
control-plane ipc-endpoint update-interval <time-in-milliseconds>	<time-in-milliseconds>	100	1000

## IPC Queue Send Thread Tuning Configuration

DRA maintains an IPC message queue where all the internal messages exchanged between Directors or Workers are stored. There are a set of IPC Queue processor threads that process these messages. During slow network conditions, when messages get delivered to one or a subset of peers, an IPC Queue processor thread that sends the message to these peers can eventually degrade the rate at which the IPC messages drop. These slow down conditions can cause message drops and 3002 errors not only to peers having degraded network but also to other peers.

The following configurations are recommended to handle the slow network peers.

**Table 9: IPC Queue - send thread Tuning Commands**

Deployment	Command	Parameters	Default Values	Recommended Value
Large	dra ipc-send-thread-limit <thread-limit> lock-sla-timeout <time-in-ms>message-throttle-duration <duration-in-ms> timeout-sample-to-throttle <max-samples>	<i>thread-limit</i> <i>time-in-ms</i> <i>duration-in-ms</i> <i>max-samples</i>	50 200 30000 150	50 (1/3rd of IPC threads) 250 30000 150
Small	dra ipc-send-thread-limit <thread-limit> lock-sla-timeout <time-in-ms>message-throttle-duration <duration-in-ms> timeout-sample-to-throttle <max-samples>	<i>thread-limit</i> <i>time-in-ms</i> <i>duration-in-ms</i> <i>max-samples</i>	50 200 30000 150	10 (1/3rd of IPC threads) 50 30000 30



### Note

- To disable the SLA timeout, configure **lock-sla-timeout** to 2000 ms. Disabling SLA timeout leads to ipc message drop in slow network.
- To disable throttling, configure **message-throttle-duration** to 0.

To disable thread throttling(limit), configure it to the value same as IPC Processor threads. Increasing **timeout-sample-to-throttle** reduces the chance of throttling the peer.