THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

University of Waikato
Network Research Group

openLI

Combining RabbitMQ with OpenLI

OpenLI Training: Chapter Twenty One

Shane Alcock
University of Waikato
New Zealand
shane.alcock@waikato.ac.nz

Adding Resiliency with RabbitMQ

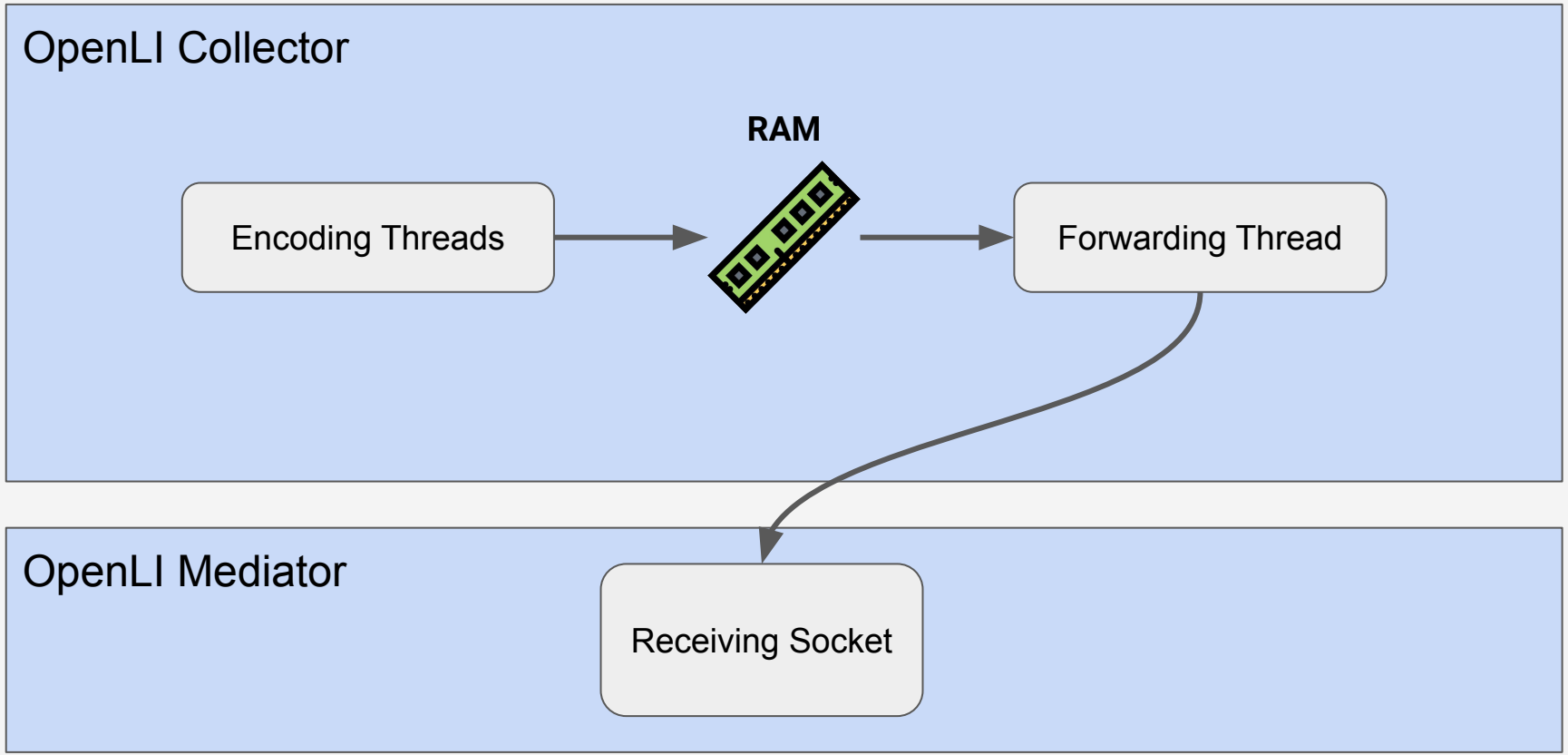
- Intercepted records are buffered in collector memory

- If the mediator is unreachable:
  - Memory consumption grows
  - If not resolved, collector process will be killed

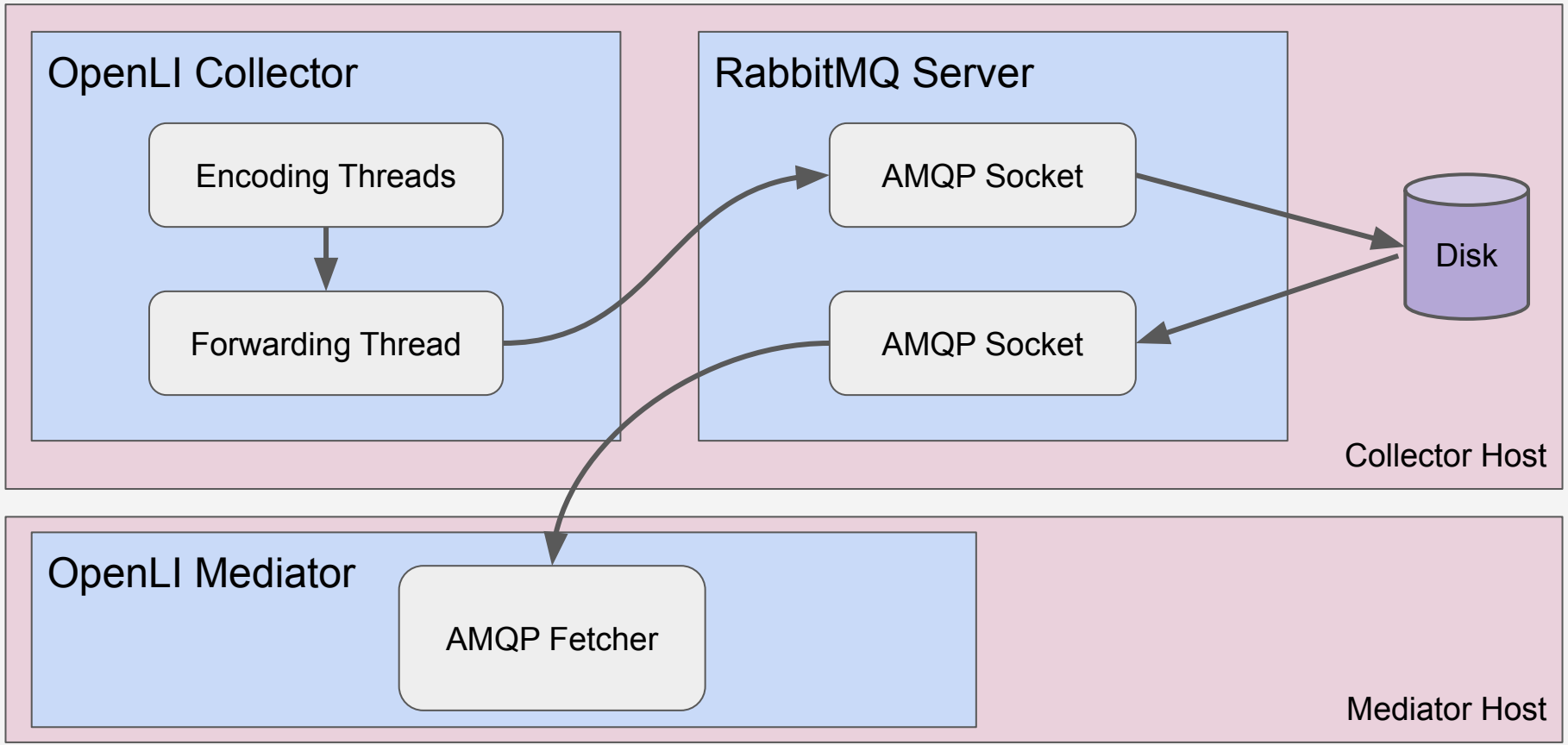
Adding Resiliency with RabbitMQ

- Using RabbitMQ, records can be buffered to disk instead
  - More time to recover the mediator
  - Can deploy collectors with less memory

Adding Resiliency with RabbitMQ



Adding Resiliency with RabbitMQ



Installing RabbitMQ

- RabbitMQ server is installed alongside openli-collector
  - Already installed on the collector component in the lab too

Starting RabbitMQ

- By default, the RabbitMQ server on the collector is stopped

```
/home/openli-coll# service rabbitmq-server start

 * Starting RabbitMQ Messaging Server rabbitmq-server                  [ OK ]
```

Adding a RabbitMQ User

- RabbitMQ is authenticated, so we need to create an account
  - First, we need a virtual host, i.e. a context

```
/home/openli-coll# rabbitmqctl add_vhost “OpenLI”

Adding vhost "OpenLI" ...
```

Adding a RabbitMQ User

- Create a user and password for the vhost
  - Username should match the CN of your SSL certificate
    - Otherwise you will not be able to use SSL Authentication
    - For the lab, the CN is “openli.nz”
  - In this example, my password is going to be “security”

```
/home/openli-coll# rabbitmqctl add_user "openli.nz" "security"

Adding user "openli.nz" ...
```

Adding a RabbitMQ User

- Grant global permissions for our new user on the vhost

```
/home/openli-coll# rabbitmqctl set_permissions -p “OpenLI” “openli.nz” “.*” “.*” “.*”

Setting permissions for user "openli.nz" in vhost "OpenLI" ...
```

Enabling SSL Authentication

- First, enable the SSL Auth plugin

```
/home/openli-coll# rabbitmq-plugins enable rabbitmq_auth_mechanism_ssl

Enabling plugins on node rabbit@7b0172927950:
rabbitmq_auth_mechanism_ssl
The following plugins have been configured:
  rabbitmq_auth_mechanism_ssl
Applying plugin configuration to rabbit@7b0172927950...
The following plugins have been enabled:
  rabbitmq_auth_mechanism_ssl

started 1 plugins.
```

Enabling SSL Authentication

- Edit the RabbitMQ configuration file using a text editor
  - /etc/rabbitmq/rabbitmq.conf
  - This file probably won't exist if your RabbitMQ install is new

```
/home/openli-coll# vim /etc/rabbitmq/rabbitmq.conf
```

Enabling SSL Authentication

- Add the following to the configuration file
  - This configuration is specific for the training lab
  - Adapt as necessary for your deployment

```
listeners.ssl.default = 5671

ssl_options.cacertfile = /etc/openli/ssl/ca-crt.pem
ssl_options.certfile = /etc/openli/ssl/collector-crt.pem
ssl_options.keyfile = /etc/openli/ssl/collector-key.pem
ssl_options.verify = verify_peer
ssl_options.fail_if_no_peer_cert = true

ssl_cert_login_from = common_name

auth_mechanisms.1 = PLAIN
auth_mechanisms.2 = AMQPLAIN
auth_mechanisms.3 = EXTERNAL
```

Enabling SSL Authentication

- Add the following to the configuration file
  - This configuration is specific for the training lab
  - Adapt as necessary for your deployment

```
listeners.ssl.default = 5671

ssl_options.cacertfile = /etc/openli/ssl/ca-crt.pem
ssl_options.certfile = /etc/openli/ssl/collector-crt.pem
ssl_options.keyfile = /etc/openli/ssl/collector-key.pem
ssl_options.verify = verify_peer
ssl_options.fail_if_no_peer_cert = true

ssl_cert_login_from = common_name

auth_mechanisms.1 = PLAIN
auth_mechanisms.2 = AMQPLAIN
auth_mechanisms.3 = EXTERNAL
```

Enabling SSL Authentication

- Add the following to the configuration file
  - This configuration is specific for the training lab
  - Adapt as necessary for your deployment

```
listeners.ssl.default = 5671

ssl_options.cacertfile = /etc/openli/ssl/ca-crt.pem
ssl_options.certfile = /etc/openli/ssl/collector-crt.pem
ssl_options.keyfile = /etc/openli/ssl/collector-key.pem
ssl_options.verify = verify_peer
ssl_options.fail_if_no_peer_cert = true

ssl_cert_login_from = common_name

auth_mechanisms.1 = PLAIN
auth_mechanisms.2 = AMQPLAIN
auth_mechanisms.3 = EXTERNAL
```

Enabling SSL Authentication

- To save some time, the lab container has file already prepared
  - Just copy it into the appropriate directory

```
/home/openli-coll# cp rabbitmq.conf /etc/rabbitmq/
```

Enabling SSL Authentication

- Stop and restart the rabbitmq-server service on the collector
  - Once again, we provide a script for the lab container
  - A real deployment will use the systemd restart command

```
# IN THE LAB

/home/openli-coll# stop_rabbitmq.sh

/home/openli-coll# service rabbitmq-server start


# IN A REAL DEPLOYMENT

/home/openli# service rabbitmq-server restart
```

Updating Collector Configuration

- Enable RabbitMQ within the OpenLI collector
  - Edit the configuration and restart the collector service

```
/home/openli-coll# vim /etc/openli/collector-config.yaml

...

RMQenabled: true

RMQname: “openli.nz”

RMQpass: “security”

RMQport: 5671

...
```

Updating Collector Configuration

- Enable RabbitMQ within the OpenLI collector
  - Edit the configuration and restart the collector service

```
/home/openli-coll# vim /etc/openli/collector-config.yaml

...

RMQenabled: true

RMQname: “openli.nz”

RMQpass: “security”

RMQport: 5671

...
```

Updating Collector Configuration

- Enable RabbitMQ within the OpenLI collector
  - Edit the configuration and restart the collector service

```
/home/openli-coll# vim /etc/openli/collector-config.yaml

...

RMQenabled: true

RMQname: "openli.nz"

RMQpass: "security"

RMQport: 5671

...
```

Updating Collector Configuration

- Enable RabbitMQ within the OpenLI collector
  - Edit the configuration and restart the collector service

```
/home/openli-coll# vim /etc/openli/collector-config.yaml

...

RMQenabled: true

RMQname: "openli.nz"

RMQpass: "security"

RMQport: 5671

...
```

Updating Collector Configuration

- Enable RabbitMQ within the OpenLI collector
  - Edit the configuration and restart the collector service

```
/home/openli-coll# vim /etc/openli/collector-config.yaml

...

RMQenabled: true

RMQname: “openli.nz”

RMQpass: “security”

RMQport: 5671

...
```

Updating Collector Configuration

- Enable RabbitMQ within the OpenLI collector
  - Edit the configuration and restart the collector service

```
/home/openli-coll# stop_collector.sh

/home/openli-coll# service openli-collector start

/home/openli-coll# less /var/log/openli/collector.log

...

openlicollector[6532]: OpenLI: Connected to RMQ instance

...
```

Updating Mediator Configuration

- Enable RabbitMQ within the OpenLI mediator
  - Edit the configuration and restart the mediator service

```
/home/openli-med# vim /etc/openli/mediator-config.yaml

...

RMQenabled: true
RMQname: "openli.nz"
RMQpass: "security"
RMQport: 5671

RMQSSL: true

RMQheartbeatfreq: 30

...
```

Updating Mediator Configuration

- Enable RabbitMQ within the OpenLI mediator
  - Edit the configuration and restart the mediator service

```
/home/openli-med# vim /etc/openli/mediator-config.yaml

...

RMQenabled: true
RMQname: "openli.nz"
RMQpass: "security"
RMQport: 5671

RMQSSL: true

RMQheartbeatfreq: 30

...
```

Updating Mediator Configuration

- Enable RabbitMQ within the OpenLI mediator
  - Edit the configuration and restart the mediator service

```
/home/openli-med# vim /etc/openli/mediator-config.yaml

...

RMQenabled: true
RMQname: "openli.nz"
RMQpass: "security"
RMQport: 5671

RMQSSL: true

RMQheartbeatfreq: 30

...
```

Updating Mediator Configuration

- Enable RabbitMQ within the OpenLI mediator
  - Edit the configuration and restart the mediator service

```
/home/openli-med# stop_mediator.sh

/home/openli-med# service openli-mediator start

/home/openli-med# less /var/log/openli/mediator.log

...

openlimediator[2312]: OpenLI Mediator: attempting to connect to RMQ using SSL on port 5671
openlimediator[2312]: OpenLI Mediator: RMQ Registered consumer ID1

...
```

Testing

- Make sure everything still works
  - Repeat some of the earlier intercept tests using the pcaps

- If something fails:
  - Check OpenLI component logs for errors
  - Check /var/log/rabbitmq on the collector
  - Carefully check configuration files for typos

More Documentation

https://github.com/wanduow/openli/wiki/Using-RabbitMQ-for-disk-backed-buffers-in-OpenLI

Next Time

- The OpenLI training is complete!
  - More advanced lessons may be available in the future
  - For now, you should have the skills you need
  - More documentation at https://github.com/wanduow/openli/wiki

- Any further questions or comments?
  - openli-support@waikato.ac.nz