

Projet :
Systèmes distribués pour le traitement de données

COURET Nathanael, HAGENBOURGER Maxime, CHENAL Vincent,
Rakotoarisoa Jérémy, SERGENT Julien, TABOULOT Clément,
MATTON Valentin, VALLADIER Victor et LAFOREST Victore

Lundi 21 Novembre 2016



Table des matières

1	RabbitMQ	3
1.1	Présentation générale	3
1.2	Fonctionnalités	3
1.3	Principes de fonctionnement	3

1 RabbitMQ

1.1 Présentation générale

RabbitMQ est un middleware de Pivotal facilitant l'échange de messages entre applications présentes sur un système informatique : message broker software. RabbitMQ implémente le protocole AMQP (conçu par un consortium international) qui standardise les échanges de messages. Écrit en Erlang, RabbitMQ est open-source et un support payant est disponible. Des bibliothèques sont disponibles pour l'écrasante majorité des langages utilisés, est compatible avec les principaux systèmes d'exploitation et bénéficie d'une large communauté d'utilisateurs. Il est utilisé par de nombreuses entreprises pour répondre à leurs besoins en échange de messages, telles que Instagram, The New York Times, Nokia, SoundCloud...

1.2 Fonctionnalités

RabbitMQ propose les fonctionnalités suivantes :

- Le transport de messages ;
- La communication asynchrone : L'émetteur d'un message et le récepteur du message n'ont pas besoin d'être activés en même temps. La file d'attente reçoit le message de l'application émettrice et le stocke jusqu'à ce que l'application réceptrice vienne lire le message ;
- Le routage : Les messages peuvent être routés entre machines ;
- Le clustering : Plusieurs serveurs RabbitMQ peuvent former un cluster ;
- La persistance des messages : Les messages d'une file peuvent être sauvegardés sur un support physique ; fiabilité : Chaque envoi ou réception par une application fait l'objet d'un accusé de réception. Couplé avec la persistance, ce mécanisme permet de garantir qu'aucun message ne sera perdu dans son transfert entre les applications ;
- La diversité des protocoles supportés : AMQP, STOMP, MQTT, AMQP, HTTP ;
- La diversité des clients : Java, Python, .NET, Ruby, PHP, etc... (plus de 20).

1.3 Principes de fonctionnement

RabbitMQ propose les différents cas d'utilisation :

- Point à point : Une seule application productrice et une seule consommatrice. Le message est retiré de la file d'attente lorsque le consommateur l'a lu ;
- Publish-subscribe : Les applications consommatrices des messages s'abonnent à un channel correspondant à une catégorie de messages qu'elles veulent recevoir. Tous les messages du channel restent disponibles tant que tous les abonnés ne les ont pas lus ;
- Work queue : Une application remplit une file de job à exécuter. Des serveurs viennent récupérer ces tâches pour les réaliser.
- Routing : Les messages sont distribués dans les queues en fonction de leur type.