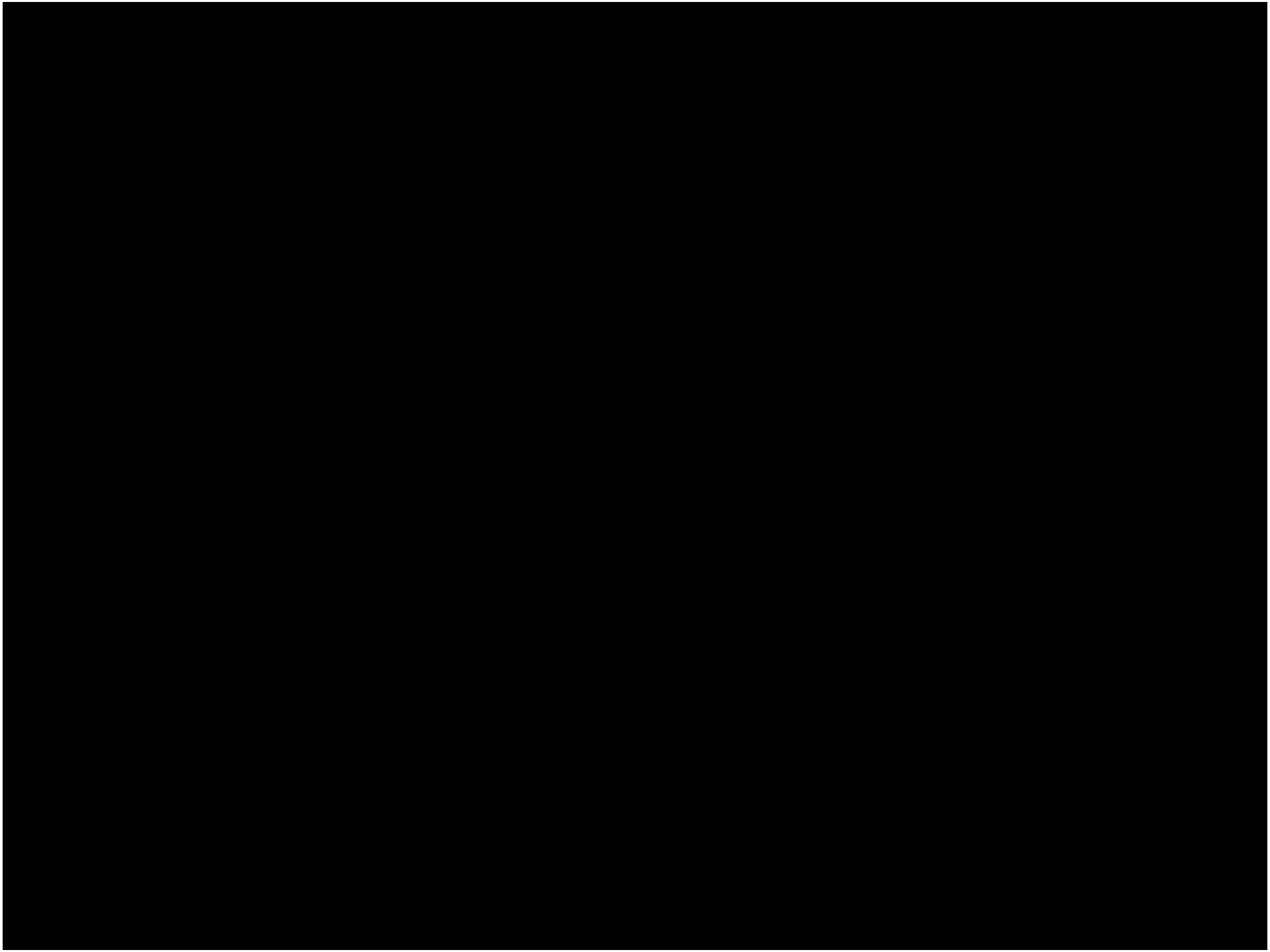


Thanks, Rackspace! ☺

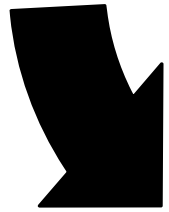
Stress Testing

Brett Jones
David McGrath
Jonathan Weldon
Paul Trampert
Scott Luxenberg

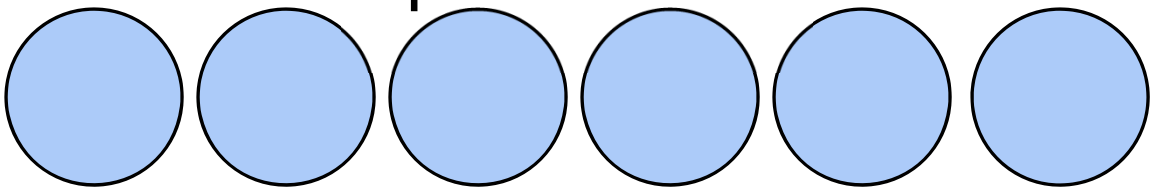
Whatis RabbitMQ?

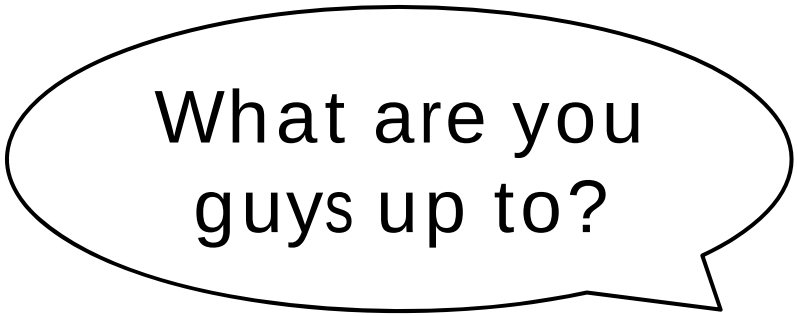
- Application messenger
- Based on publish-subscribe model
  - Content creators (publishers) send content to RabbitMQ messengers
  - Messengers distribute content to content readers (subscribers)

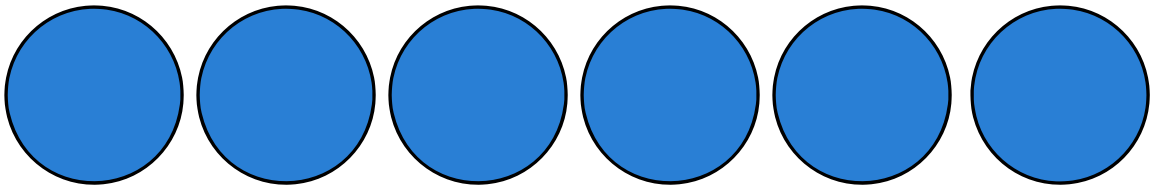
Visualization!



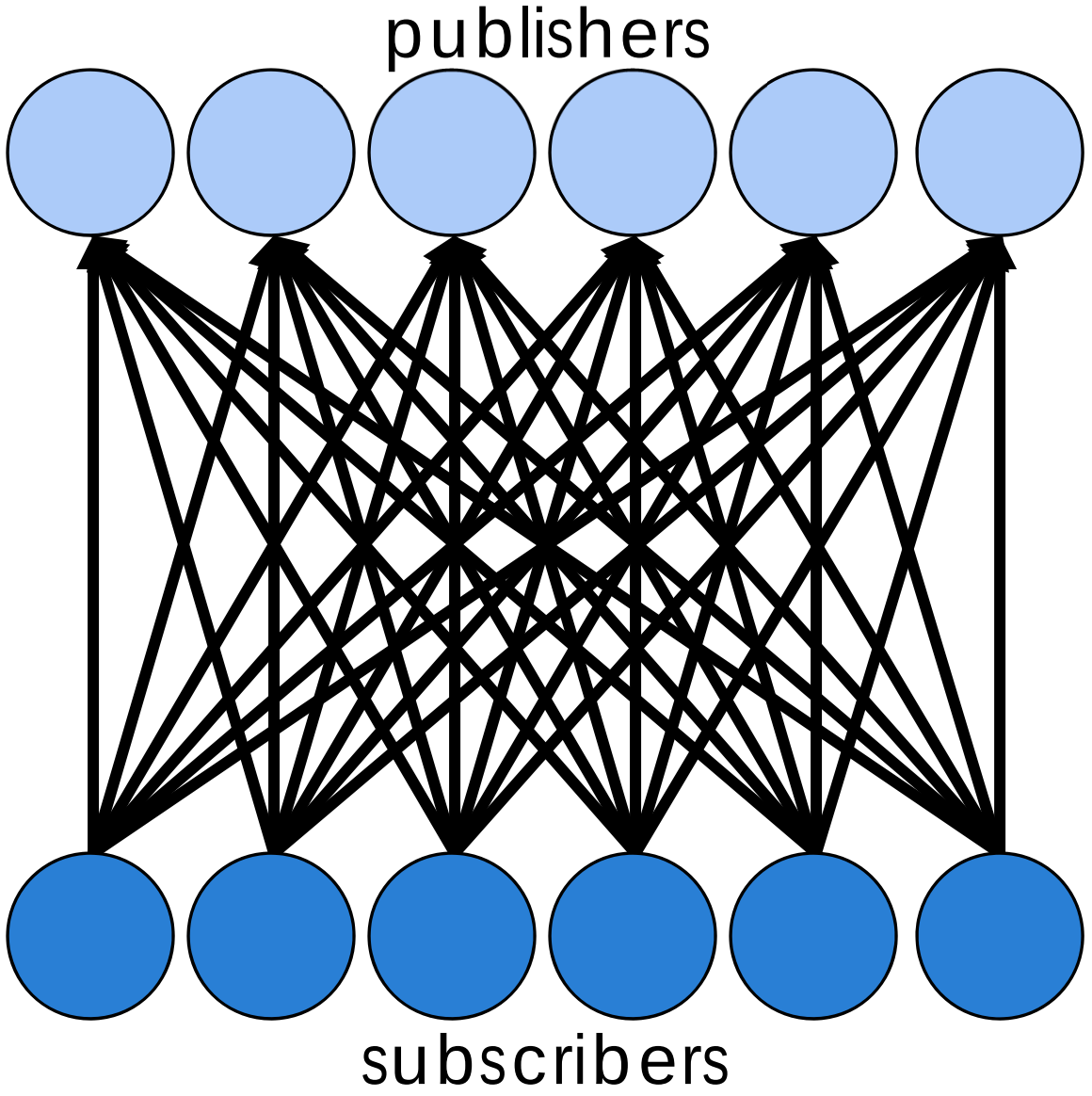
publishers

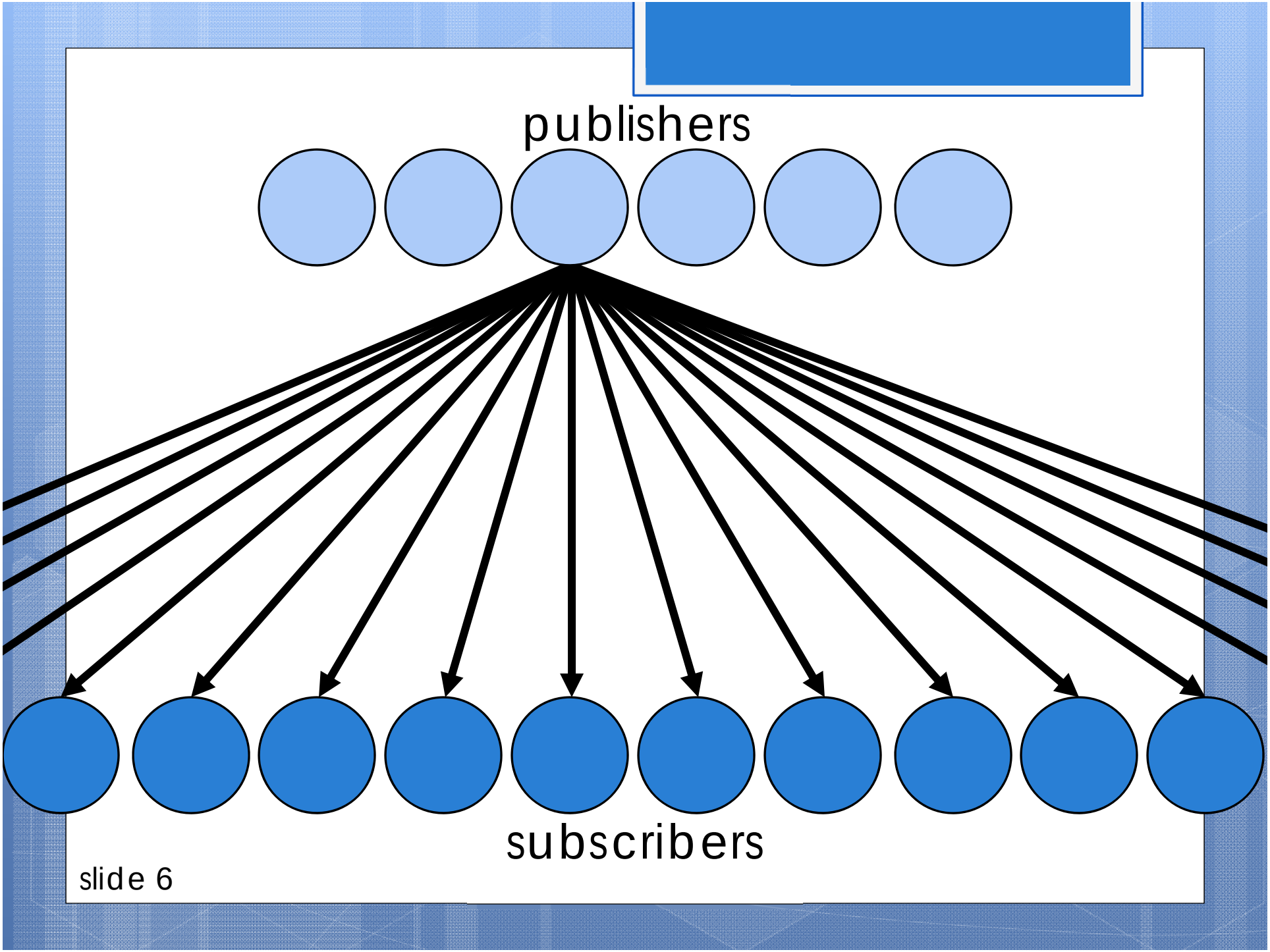


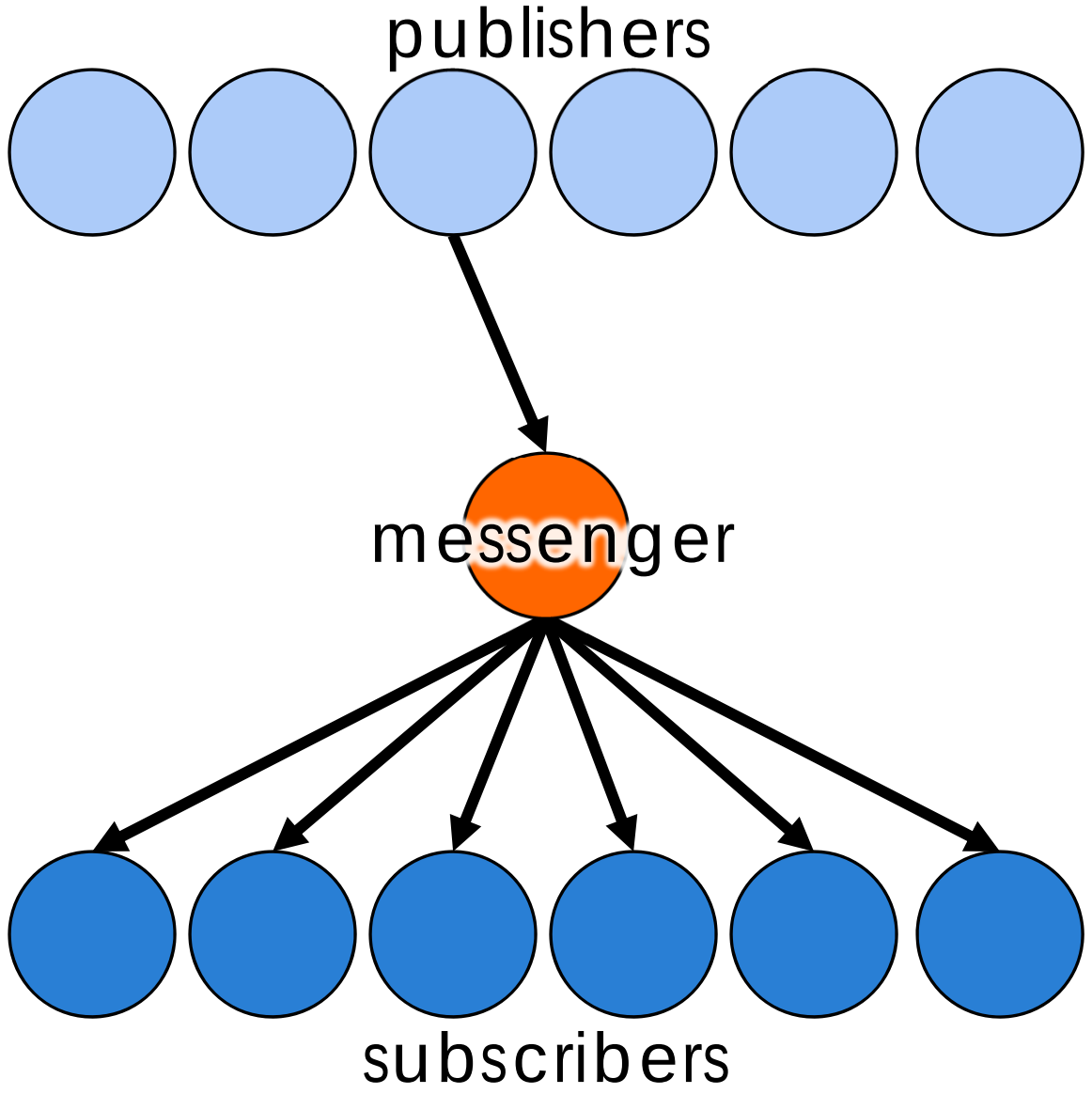




subscribers


publishers
subscribers


publishers
subscribers


publishers
messenger
subscribers

What is RabbitMQ?

- Advantages:
  - No polling
  - Publishers abstracted from subscribers
  - Publishers not overloaded with pushing data

Motivation

- Rackspace looking for scalable messaging architecture
  - Purposed for distributed system messaging backbone within data centers
- Determining the scalability of RabbitMQ can open up more options for its usage in large-scale applications

How well does RabbitMQ scale?

Approach

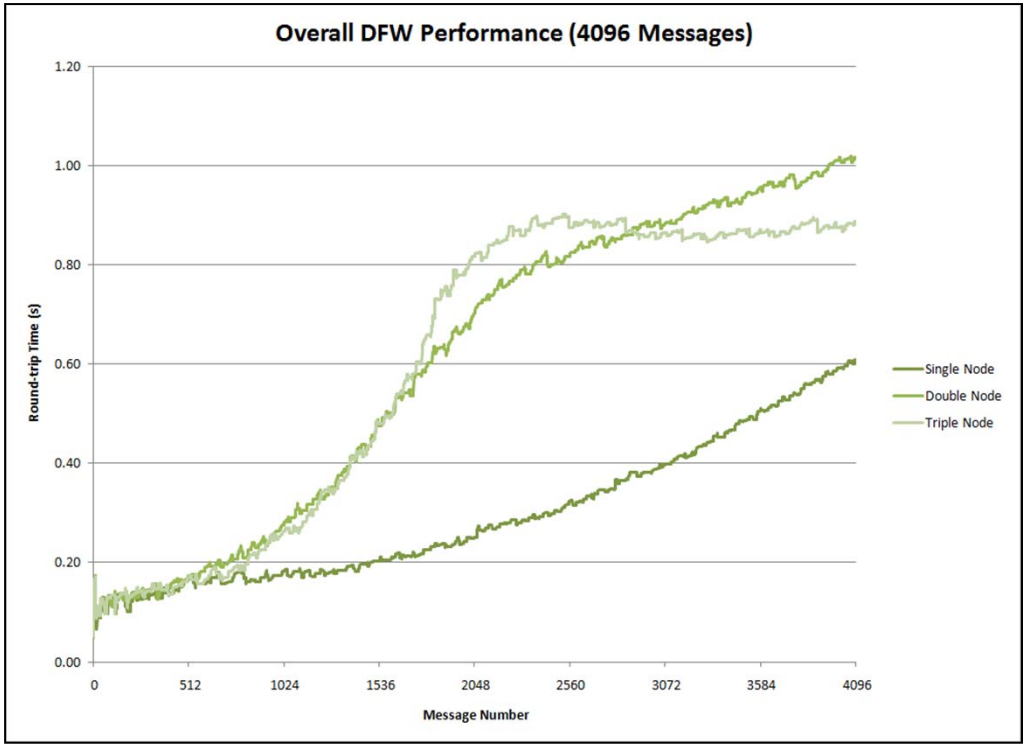
- Write programs that continuously send/recieve messages to/from the RabbitMQ server, while adjusting how many RabbitMQ nodes are acting as one server.
  - We plan to achieve this by making extensive use of virtual machines, as well as using resources provided to us by Rackspace and Virginia Tech.
- Will also adjust the number of publishers and subscribers

Tests

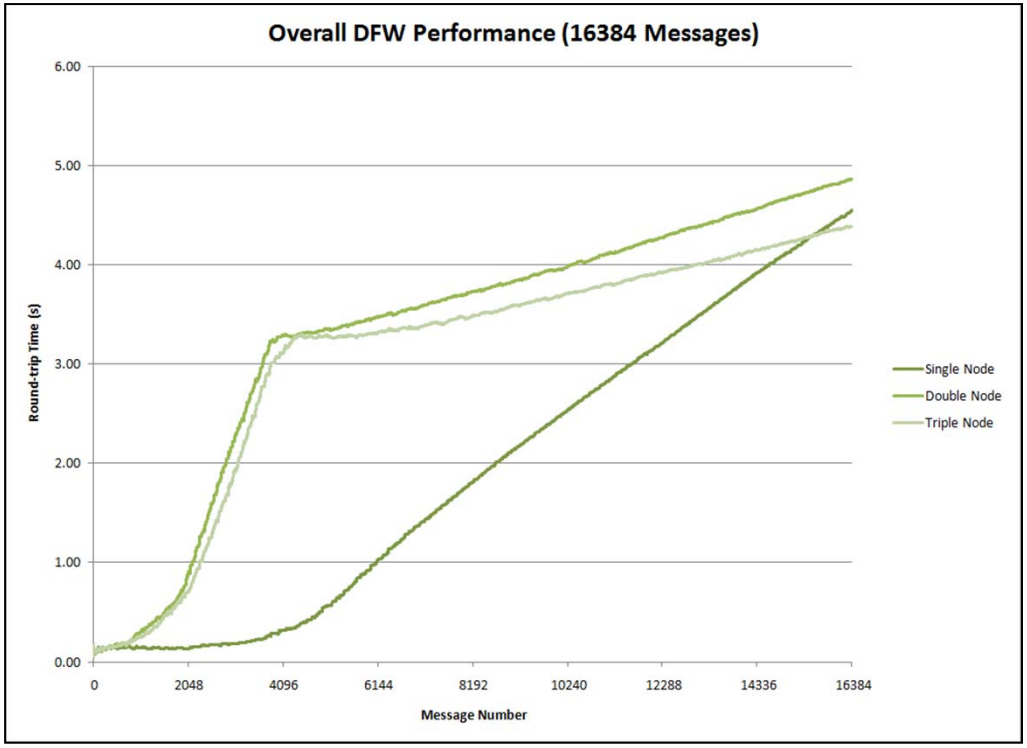
Single Publisher, Single Subscriber

- Process
  - Tested with 1KB messages
  - Graphed round-trip times for each message
  - Used 4K and 16K message sets with clusters of 1-3 nodes in Dallas/Fort Worth data center

Single Publisher, Single Subscriber



Single Publisher, Single Subscriber



Single Publisher, Single Subscriber

- Results
  - Multi-node clusters initially perform worse than single-node clusters, but showed smaller slopes after receiving all messages

Single Publisher, Multiple Subscribers

- Process
  - Tested with 1KB messages
  - Graphed times to receive all messages in 1-4 subscribers
  - Used 8-2K message sets (in powers of 2) with clusters of 1-3 nodes in Dallas/Fort Worth data center

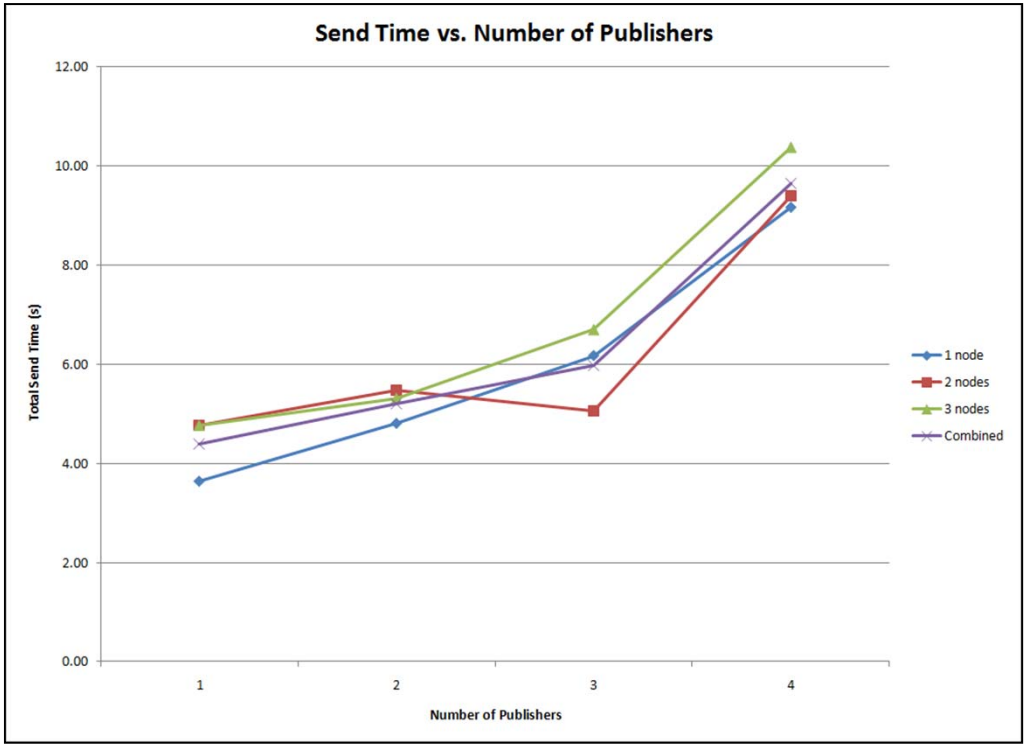
Single Publisher, Multiple Subscribers

- Results
  - Ended up scrapping this test due to complications with uneven message distribution

Multiple Publishers, Single Subscriber

- Process
  - Tested with 1KB messages
  - Graphed times to send all messages from 1-4 publishers
  - Used 16K message sets with clusters of 1-3 nodes in Dallas/Fort Worth data center

Multiple Publishers, Single Subscriber



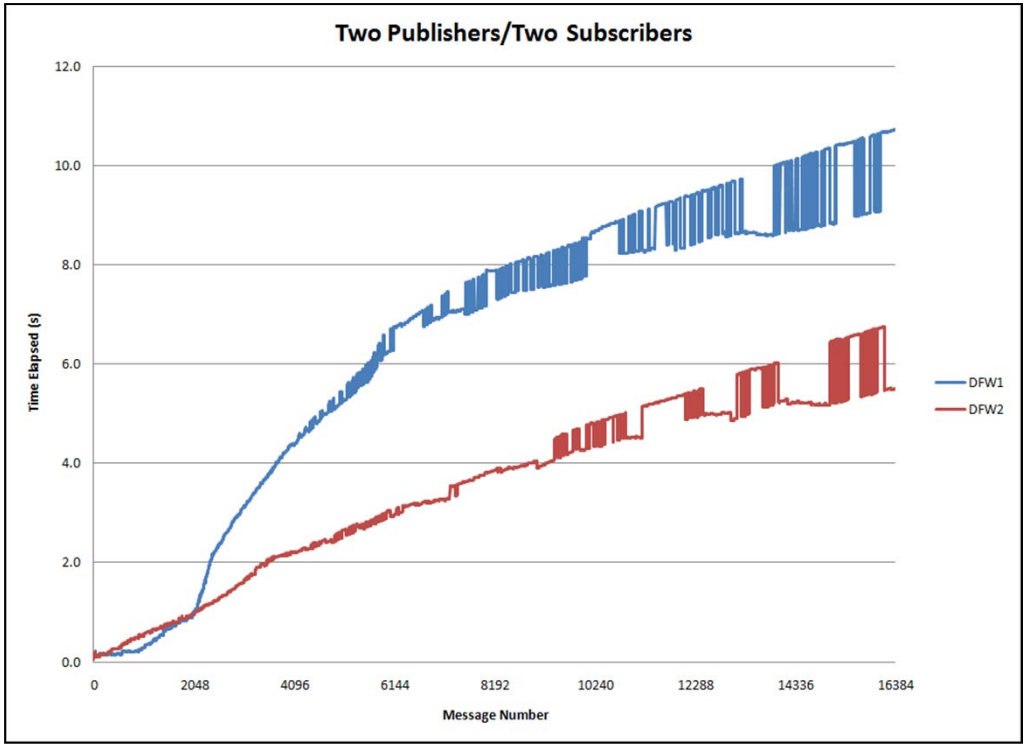
Multiple Publishers, Single Subscriber

- Results
  - Uniform scaling from 1 to 4 publishers
  - No noticeable impact for number of nodes in the cluster

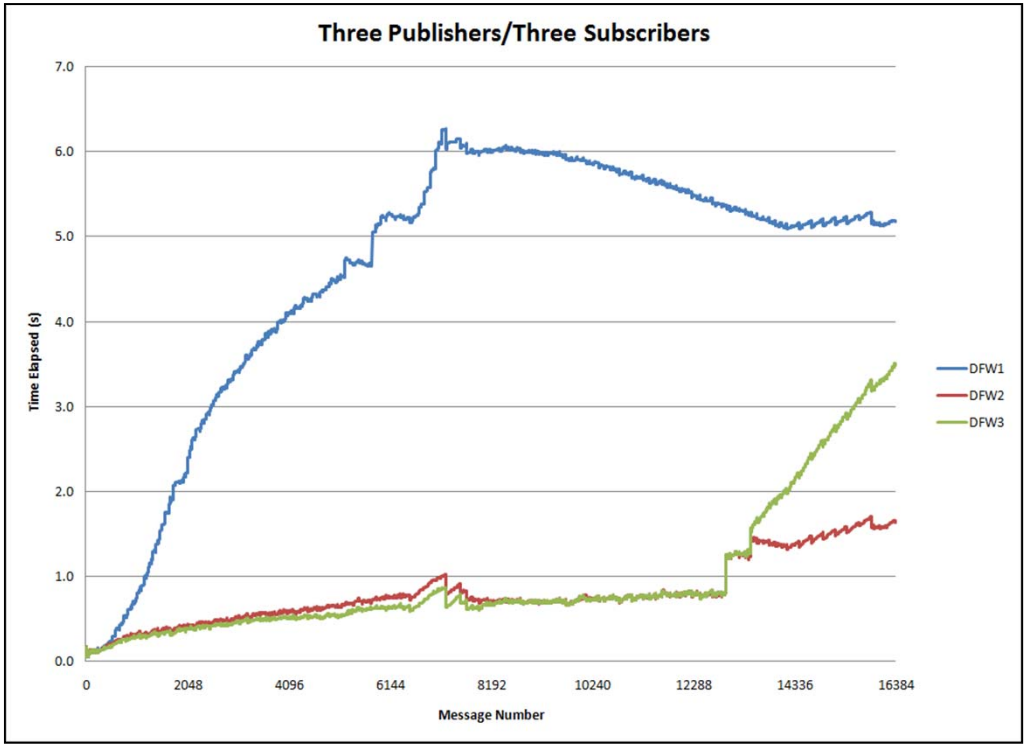
Multiple Publishers, Multiple Subscribers

- Process
  - Tested with 1KB messages
  - Graphed round-trip times for each message
  - Used 16K message sets with clusters of 1-3 nodes in Dallas/Fort Worth data center
  - Each publisher sent messages to a different node

Multiple Publishers, Multiple Subscribers

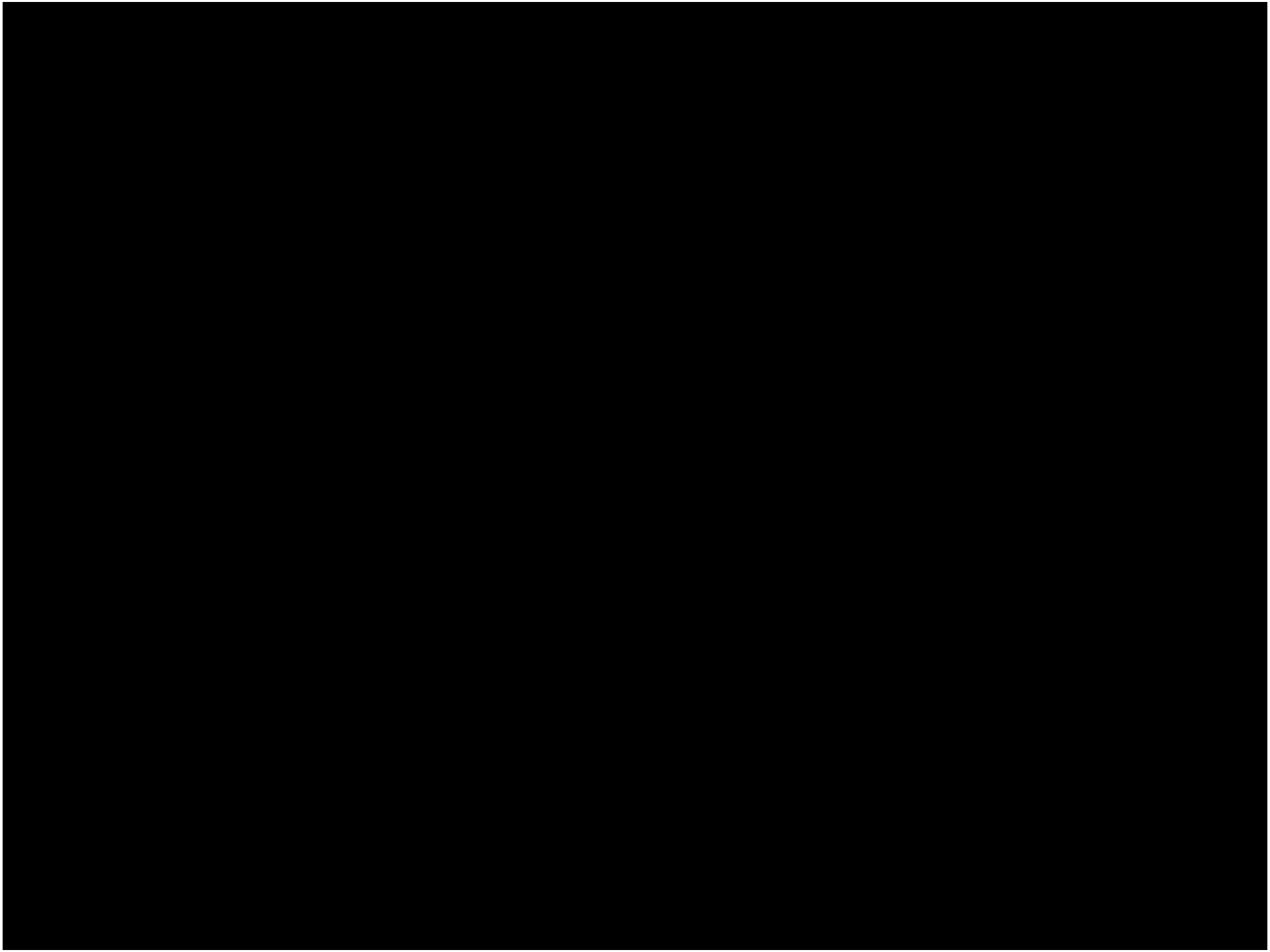


Multiple Publishers, Multiple Subscribers



Multiple Publishers, Multiple Subscribers

- Results
  - Increasing publishers and subscribers puts highest strain on head node
  - Some messages may be received out of order
    - Exchange vs. queue

Demo

Questions?