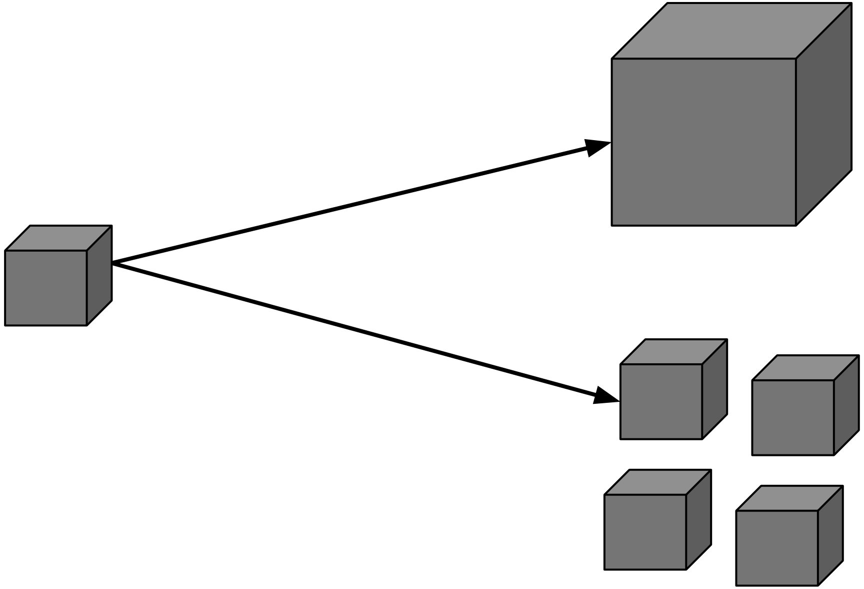
Two approaches to scale your processing: Task Queues and Workflows

Eoin Brazil, PhD, MSc, Team Lead, MongoDB

What happens when your application has one order more ‘use’?



vertical

horizontal

Request - Response

- Everything in one request
- Do it in another request

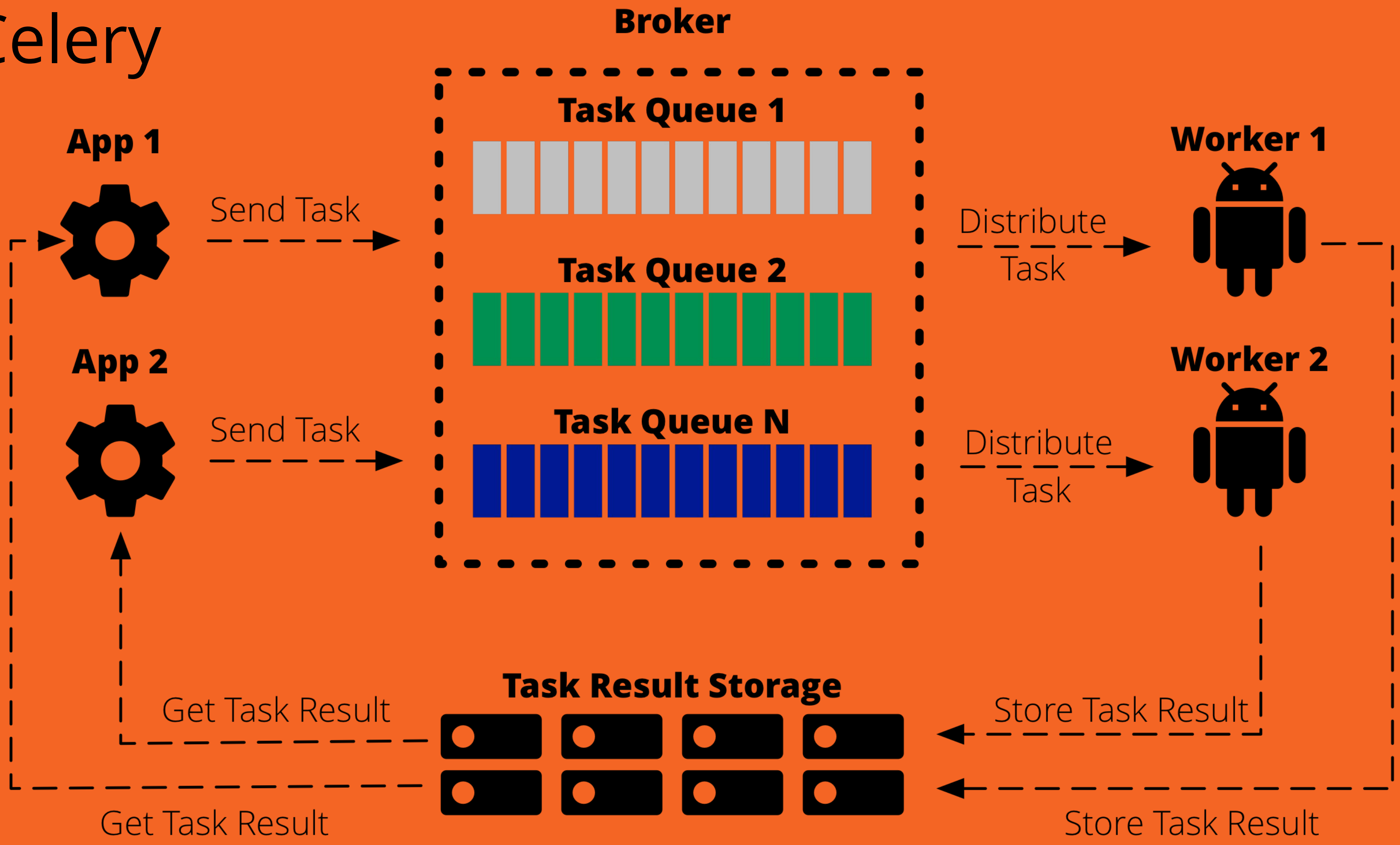
- *Move the request out to a separate process completely*

Queues and Workflows

Asynchronous distributed task queue library, Celery.

A defined sequence of tasks is typically defined as a workflow. Airflow is one such workflow management system.

Celery


Broker
Task Queue 1
Task Queue 2
Task Queue N
App 1
App 2
Send Task
Send Task
Worker 1
Worker 2
Distribute Task
Distribute Task
Task Result Storage
Get Task Result
Get Task Result
Store Task Result
Store Task Result

Tasks

Task

- Exists until acknowledged
- Results can be stored or ignored
- **State** - Pending, Received, Started, Success, Failure, Revoked, Retry

- Definition styles - class or function

Task Definition Examples

```
@app.task
def add(x, y):
    return x + y

add.apply_async((2, 2), link=add.s(16),
expires=60, retry=False)
```

How to call a Task

apply_async(args[, kwargs[, …]])

delay(*args, **kwargs)

calling (__call__)

Link so callback results will be applied to next task as partial argument.

Task Options

ETA and countdown, Expiration
Serialisation - JSON, pickle, YAML and msgpack
Compression - gzip or bzip2

Routing - priority, task_routes

Workflows

Task Workflows

Signatures: Wraps a single task, groups & callbacks.

Primitives: Building blocks to allow you compose more complex tasks or simple workflows.

Task Signatures

Partials: Add args, kargs, or new options

Immutables: Unchangeable signature

Callbacks: Takes parent value
add.apply_async((**2, 2**), link=add.s(**16**))

Task Primitives 1 / 2

Groups - list of task applied in parallel

Chains - links signatures into a chain

Chords - Group/Chain hybrid of header tasks plus body tasks

Task Primitives 2 / 2

Map: Same as built-in, task.map([1, 2]) gives res = [task(1), task(2)].

Starmap: Args*, add.starmap([(2, 2), (4, 4)]) -> res =[task(2,2), task(4,4)]

Chunks: Breaks longer list into parts

Workers

Worker Settings/Options

Concurrency - multiprocessing, Eventlet

Limits - time, rate, max tasks, max memory

Queues, Autoscaling

Scheduling

Do Task X at Time Y or in Z (time units)

Celery beat or RedBeat (Heroku)

In number of seconds as an integer, a timedelta, or a crontab

Custom scheduler

OpenEdx

- Grade updates
- Sending of bulk email
- Generate course structure
- CMS User task emails
- Account / User activation email
- Instructor tasks - update scores, calculate responses, send emails

Airflow

Why Airflow 1 / 2 ?

- Web server that can render UI
- Metadata DB stores models
- Charting
- Workers (Mesos, Celery, Dask, Local, Sequential)
- Hooks (various DB interfaces)
- Operators (a node / action in DAG)

Why Airflow 2 / 2 ?

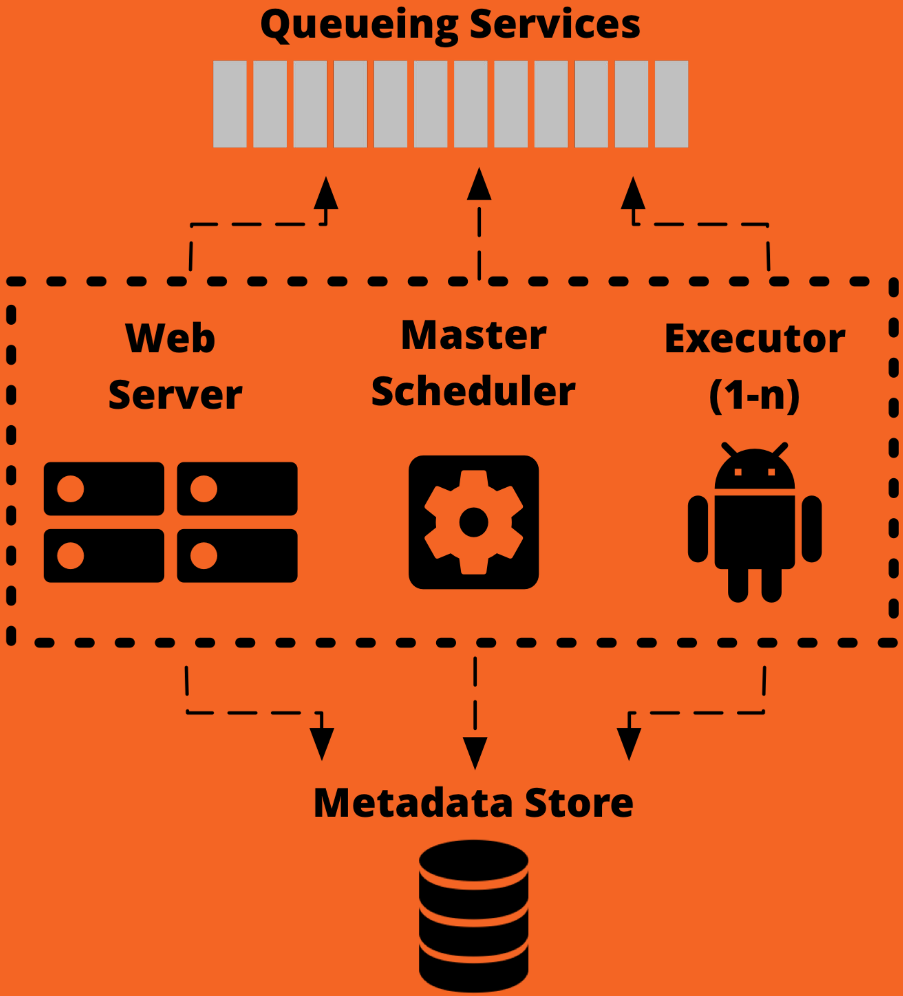
Facilitates more complex workflows, the base unit is the Directed Acyclic Graph (DAG).

Tasks A, B, and C. It could say that A has to run successfully before B can run, but C can run anytime.

Celery and Airflow

*“**CeleryExecutor** is one of the ways you can scale out the number of workers. For this to work, you need to setup a Celery backend (RabbitMQ, Redis, …) and change your airflow.cfg to point the executor parameter to **CeleryExecutor** and provide the related Celery settings.”*

Airflow



Optional Components

Kerberos Ticket Server

Celery Flower

Key Concepts of ‘Work’ in Airflow

DAG: ordering of work

Operator: template of how to do the work

Task: parameterized instance of an operator

Task Instance: a task assigned to DAG and with a state linked to specific run of the DAG

Functionality for complex workflows

- Hooks
- Pools
- Connections
- Queues
- XComs

- Variables
- Branching
- SubDAGs
- Service Level Agreements (SLAs)
- Trigger Rules

When to use which ?

Celery

- RAM / CPU
- MLasS e.g. ores
- Social Media
  - Feeds, Deletions, CrossPost, Spam

Airflow

- ETL Jobs e.g. Astronomer
- Batch jobs e.g. Robinhood
- Complex workflows / jobs

Resources

Documentation and Online User Groups

- Celery
  - http://docs.celeryproject.org/en/latest/userguide
  - https://groups.google.com/forum/#!forum/celery-users

- Airflow
  - https://airflow.incubator.apache.org/index.html
  - https://lists.apache.org/list.html?dev@airflow.apache.org