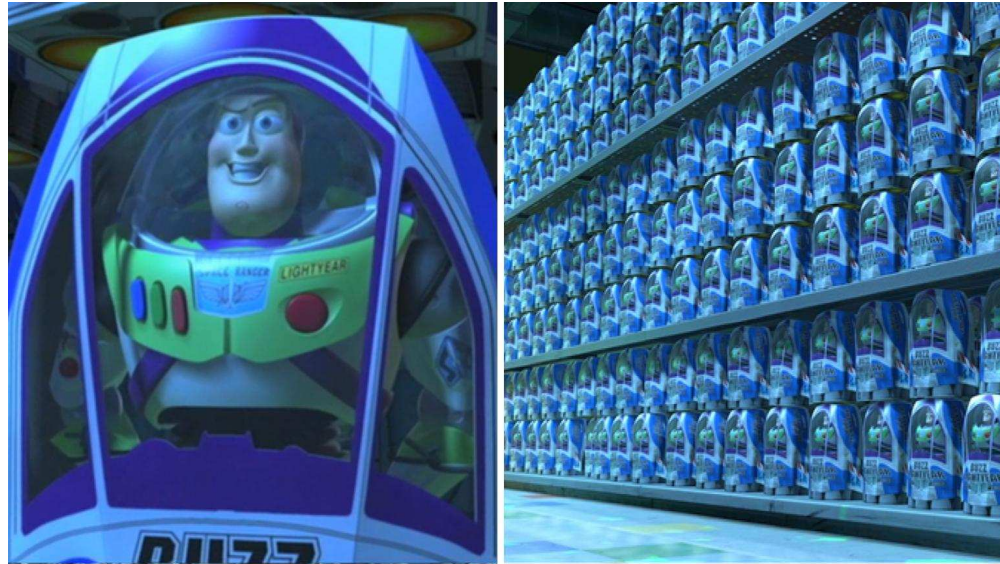


If the Queue Fits

A Novel Task

Let me guess: You need to make HTTP requests to some flaky provider to record numerous points of data for many different customers.



All at Once

Node supports concurrency, so why not?

```
await Promise.all(allTheStuff);
```

”aLI aT oNcE”

Rate limits, resource contention, errors, chaos.



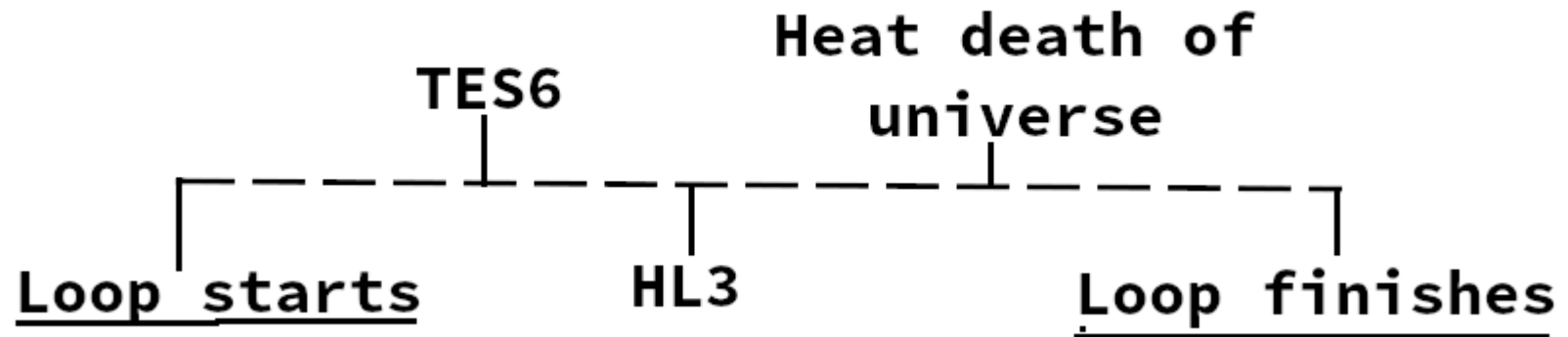
One at a Time

Okay then, so we'll go slower.

```
for (const thingToDo of thingsToDo) {  
  await thingToDo();  
}
```

”oNe aT A tImE”

Slow as molasses.



What Do We Want?

- Concurrency / parallelism
- Crash resilience
- Error handling and retries

RabbitMQ

- Networked message-broker
- Easily solves concurrency and crash concerns
- Number of consumers ~ resource usage



RestOfTheRabbitMQ

- Great for generic messaging
- Many useful features
- Job queue requires more work

How to draw an owl

1.



2.



1. Draw some circles

2. Draw the rest of the fucking owl

RI RetailLink Services

- Used amqplib directly
- Inconsistent error handling
- Hard-to-follow retry logic



Queueball Begins

- RabbitMQ = good, lots of boilerplate = bad
- Consistency can help avoid previous issues
- Stephen vendored it into analytics-scheduler

Example

```
const config = {
  rabbitmqUrl: 'amqp://guest:guest@localhost:5672',
  name: 'queueball-test',
};

const producer = new Producer(config);
await producer.start();

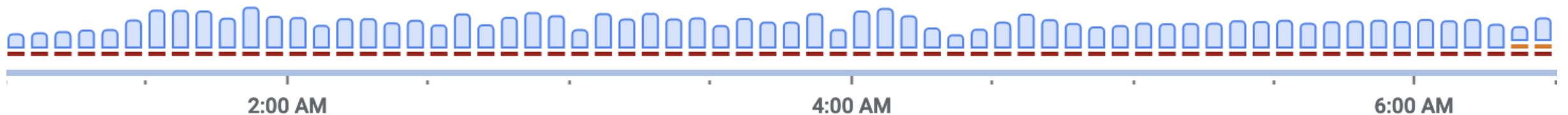
const consumer = new Consumer(config);
consumer.processJob = async ({ payload }) => {
  console.log(payload.message);
};
await consumer.start();

producer.enqueue({ message: 'Hello world!' });
```

Punched in the Face

- Queued jobs are a happy path
- Things rarely go as planned
- Additional techniques can help

 Create metric



Resource Locks

- Lock resource when job starts
- Per user, or even whole job type
- Re-enqueue conflicting jobs
- Redis or Postgres can work

```
await this.sessionsRepo.manager.transaction(async (manager) => {  
  const lockKey = this.generateLockKey(`supplier-session-${supplierId}`);  
  await manager.query('SELECT pg_advisory_xact_lock($1)', [lockKey]);  
  // ...  
}
```

Job Striping

- Space resource contenders apart
- AAABBBCCC -> ABCABCABC

```
const jobPayloadGroups: JobPayload[][] = [];  
  
for (const supplier of suppliers) {  
  const jobPayloads: JobPayload[] = [];  
  jobPayloads.push(...someSupplierJobPayloads);  
  jobPayloads.push(...moreSupplierJobPayloads);  
  jobPayloadGroups.push(jobPayloads);  
}  
  
return zipArrays(...jobPayloadGroups);
```

Structure Tips

- Job generation can be a job
- Prefer smaller, simpler jobs
- Have contextual error messages

Tooling Tips

- Use Kube for scheduling
- Add job params to Sentry errors
- Use trace ID when searching logs

Job

deductionId

6723

id

5ceec178-2ee5-4c0f-99a7-bb69aedcddd9

kind

scrape-purchase-order-pdf

priority

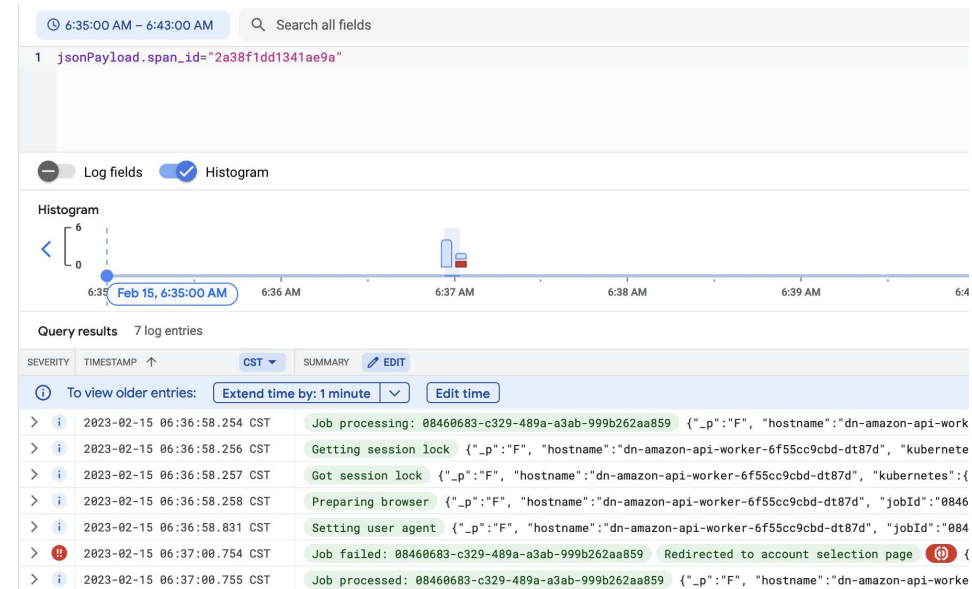
low

purchaseOrderNumber

4PX46K3C

supplierId

22



My Ask

If you use Queueball, let me know how I can make it better.

