EclairJS = Node.js + Apache Spark

Doron Rosenberg
IBM Emerging Technologies

Two Emerging Trends

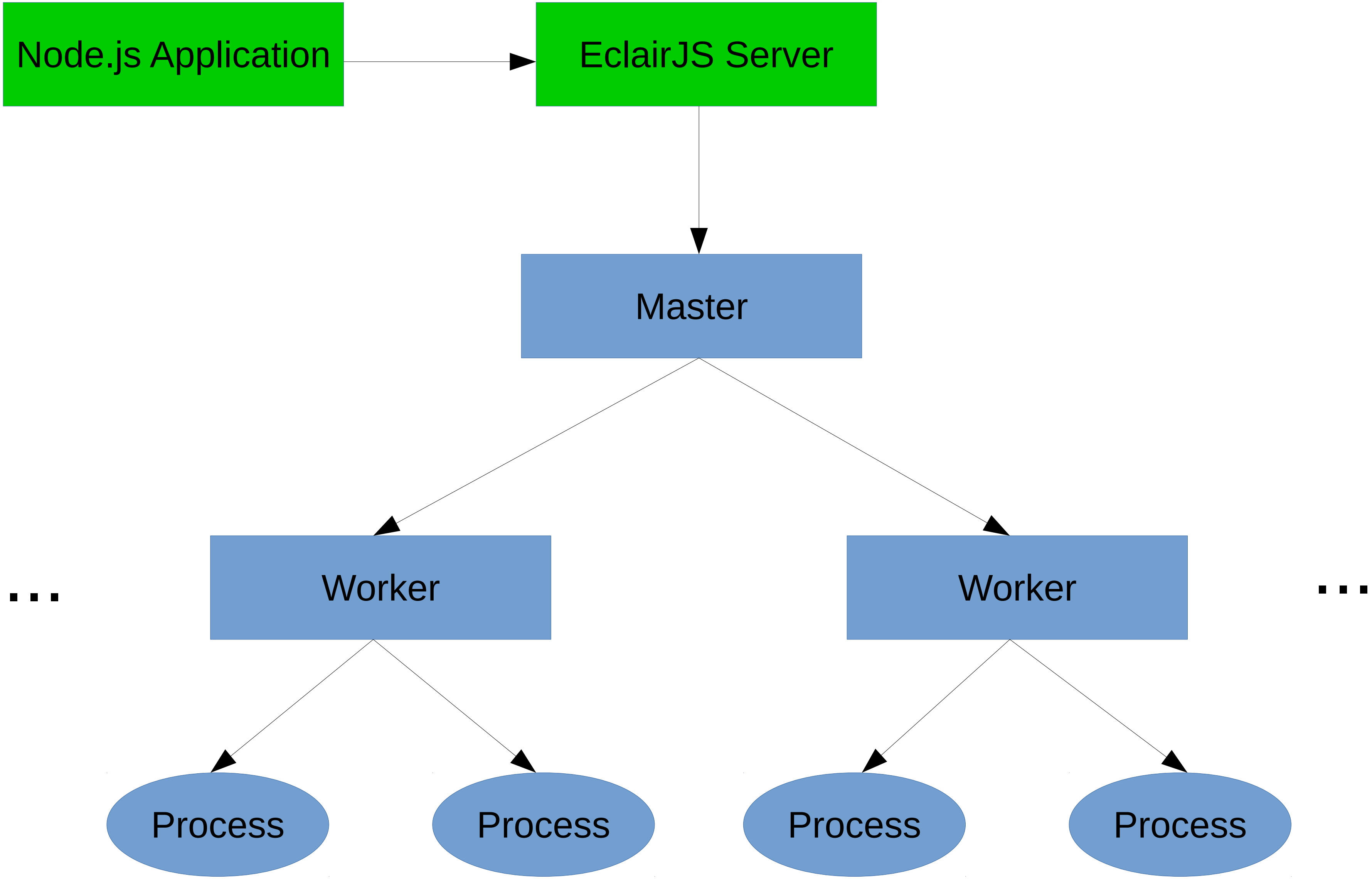
- Node is growing fast and being used by companies of all sizes across a variety of industries
- Companies are generating more and more data (logs, transactions, user actions, etc)
- Very little of all data is ever analysed
- Companies want to gain insight from all this data
- “Big Data” - too much data for one machine to handle, need to distribute data and work load, distributed computing analytics

Apache Spark

- Fast and general engine for large-scale data processing
- Came out of Berkley Amplab in 2009, contributed to Apache, focused on performance
- Common API for static & streaming data, SQL, Machine Learning, GraphX
- Master node controls multiple worker nodes, with each running multiple worker processes.
- Highly scalable and fault tolerant
- Local mode for easy development
- Run on the JVM, supports Java/Scala/R/Python

EclairJS

- http://eclairjs.org, http://github.com/EclairJS
- Provides Node.js developers access to Apache Spark, Open Source, Apache 2 licensed
- Composed of two parts
  - Client, a npm module
  - Server, which sits in front of Apache Spark and enables native Javascript support in Apache Spark, based on Apache Toree
- Bring “Big Data” closer to the application developer, reduce barrier to entry
- Node is inherently async, strong fit for dealing with long running Apache Spark workloads.


Node.js Application
EclairJS Server
Master
...
Worker
Worker
...
Process
Process
Process
Process

Basic Example

```
var eclairjs = require('eclairjs');
var spark = new eclairjs();

var sc = new spark.SparkContext("local[*]", "Spark Example");

var data = sc.parallelize([1.10, 2.2, 3.3, 4.4]);

var doubleddata = data.map(function(num) {     // Runs on Spark workers
    return num * 2;
});

doubleddata.collect().then(function(results) {
    sc.stop();
    console.log(“Results: “, results);
})
```

Word Count

```
var textFile = sparkContext.textFile('foo.txt');

var words = textFile.flatMap(function(line) {
  return line.split(" ");
});

var wordsWithCount = words.mapToPair(function(word, Tuple2) {
  return new Tuple2(word, 1);
}, [spark.Tuple2]);

var reduced = wordsWithCount.reduceByKey(function(value1, value2) {
  return value1 + value2;
});

reduced.collect().then(function(results) {
  console.log('Word Count:', results);
  sc.stop();
});
```

Demo

Store Machine Learning Demo

https://github.com/EclairJS/eclairjs-examples

Project Status

- Support for Spark 2.0 and 1.6
- Performance – slower than Java, faster than Python (benchmark)
- https://github.com/EclairJS/eclairjs (new, merge of two previous repos)
- Examples for SQL, Streaming, Machine Learning (all of them!)
- No GraphX

Try It!

```
docker pull eclairjs/minimal-gateway
docker run -p 8888:8888 eclairjs/minimal-gateway
```

Includes:

- Apache Spark
- EclairJS Server
- Example data

In Closing

- EclairJS allows Node.js application developers to work directly with Apache Spark, using 100% Javascript
- Open Source, in active development on Github for the past year
- Looking for contributors and use cases!

Thank You

eclairjs.org

github.com/EclairJS/eclairjs

groups.google.com/forum/#!forum/eclairjs

eclairjs.slack.com

doronr at us.ibm.com