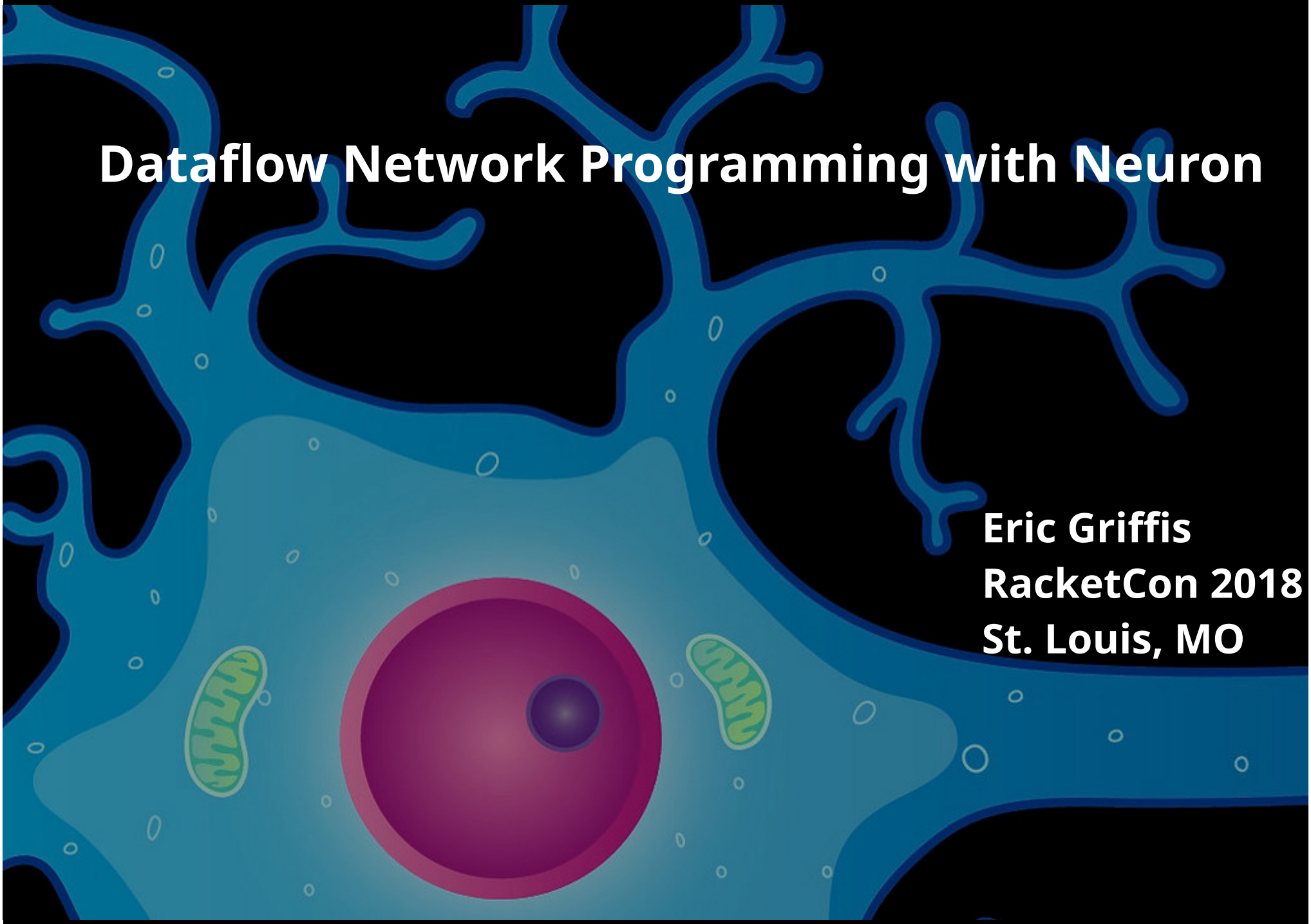
Dataflow Network Programming with Neuron



Eric Griffis

RacketCon 2018

St. Louis, MO

I. Meet Nick

Nick cares about cancer

Curing cancer is expensive

Lakhan Gulati

Follow · January 4 ·

This little baby has cancer and he need money for surgery
Facebook has decided to help by giving 1 Like = 2 dollars . 1 Comment = 4 dollars . 1 Share = 8 dollars

,
Please dont scroll down without typing Amen

Like Comment Share


Come on Facebook...
just 40 more likes and I can save that little girl's life

II. Enter Neuron

The Neuron Framework

concurrency model

messaging API

process construction library

Lightweight Processes

```
(process (λ () (forever (emit (f (take))))))
```

Synchronous and Asymmetric

Simple Exchange

```
(give π v) → (take)

(recv π) ← (emit v)
```

Synchronous and Asymmetric

Mediated Exchange

$$(\texttt{give}\ \pi_{\texttt{fwd}}\ \texttt{v}) \rightarrow (\texttt{forward-to}\ \pi) \rightarrow (\texttt{take})$$

$$(\texttt{recv}\ \pi_{\texttt{fwd}}) \leftarrow (\texttt{forward-from}\ \pi) \leftarrow (\texttt{emit}\ \texttt{v})$$

Serial Messaging Endpoints

Sockets

Codecs

Sockets

- input-output port
- simplified messaging API
- prevent half-open connections

Codecs

- **printer** and **parser** functions
- **encoder** and **decoder** functions

Parsers & Printers

```
parser :: (-> input-port? any/c)
```

read

read-json

```
printer :: (-> any/c output-port? void?)
```

writeln

write-json

Codecs

```
decoder :: (-> parser (-> socket process))
```

```
encoder :: (-> printer (-> socket process))
```

```
codec :: (-> parser printer (-> socket process))
```

UDP Endpoints

```
(define π_udp-src (udp-source read "::" 5000))
```

```
(define π_udp-snk (udp-sink write "somehost" 5000))
```

TCP Endpoints

```
(define π_tcp-cli (tcp-client "somehost" 1234))

(define π_tcp-srv (tcp-server 1234))

(define π_tcp-svc (tcp-service sexp-codec 1234))
```

III. Putting it all Together

Multiplayer Game Server

TCP service

register clients

UDP sinks

broadcast world state

world simulator

refresh world state (default 10 Hz)

Game Server: TCP Service

```
(define π_svc (tcp-service sexp-codec 3000))
```

```
(forever
 (define-values (key msg) (recv π_svc))
 (match msg
   [`(SET ,host ,port)  (set-client key host port)]
   [`(DROP ,host ,port) (drop-client key)]
   [`(MOVE ,Δx ,Δy)     (move-client key Δx Δy)]))
```

Game Server: UDP Sinks

```
(udp-sink writeln host port)
```

Game Server: World Simulator

```
(define π_sim (simulator update-world))

(for ([π_pub (all-clients)])
  (give π_pub the-world))
```

Complete Game Server

```
(define-values (world clients) (values (make-hash) (make-hash)))
(define πsvc (tcp-service sexp-codec 3000))
(define (set-client key host port)
  ...
  (hash-set! clients key (udp-sink writeln host port)))
(define (update-world Δt)
  ...
  (define the-world (hash->list world))
  (for ([πpub (hash-values clients)])
    (give πpub the-world)))
(define πsim (simulator update-world))
(forever
 (define-values (key msg) (recv πsvc))
 (match msg
   [`(SET ,host ,port)  (set-client key host port)]
   [`(DROP ,host ,port) (drop-client key)]
   [`(MOVE ,Δx ,Δy)     (move-client key Δx Δy)]))
```

IV. Summary & Conclusion

Summary

- Neuron is a compositional framework
- Making network software development easy (and fun!)

What Happened to Nick?

Thank You

Eric Griffis

https://dedbox.github.io/

dedbox@gmail.com

@dedbox on twitter and github