

Strange Loop

"It's better to be safe than sorry" is such crap. You know what's better than being safe? Being AWESOME. - Jeff Atwood

Can you smell what the loop is cooking?

Sept 23-25, 2012

Welcome!

Strange Loop is finally here. We're glad you could make it!

Connect

There are lots of opportunities to stay informed about what's happening at the conference. Staying on top of official conference changes and what others are doing is essential to making the most of your experience.

Some tools for you:

- **Twitter** - follow the Twitter hashtag **#strangeloop** or watch the official conference Twitter feed at **@strangeloop_stl** for up to the minute info.
- **Guidebook** - the Guidebook mobile app and guide are the official place for Strange Loop information on your mobile device. Download the app (<http://guidebook.com/g/sl12/>), then grab the guide for Strange Loop. The app contains schedule, maps, sponsors, a tweet stream, and more.



- **Blog** - for longer items, watch the conference blog at <http://thestrangeloop.com/news>.
- **Email** - every day of the conference an email of the day's events will go out to attendees. Check your spam folders if you're not seeing it!

Thanks

Thanks for attending Strange Loop. If you have any questions or suggestions during the conference or after, please feel free to tweet @strangeloop_stl or email info@thestrangeloop.com. If we can fix it, we will.

The Strange Loop team: Alex Miller, Mario Aquino, Ryan Senior, and Nick Cowan

Want More?

You will be able to find Strange Loop 2012 videos on InfoQ (<http://infoq.com>) in the months following the conference. Many thanks to InfoQ for making the filming, editing, and hosting of the videos possible. Slides will be available as soon as possible on the Strange Loop github site (<https://github.com/strangeloop/strangeloop2012>).

Check these out!

If Strange Loop just doesn't happen often enough for you, check out these other fine conferences from the makers of Strange Loop:



Clojure/West

Clojure/West (<http://closurewest.org>) is a west coast Clojure conference with an emphasis on new Clojure developers and its growing use as a high-leverage tool for working developers. The second edition of Clojure / West will take place in Portland, Oregon tentatively March 18-20th, 2013.



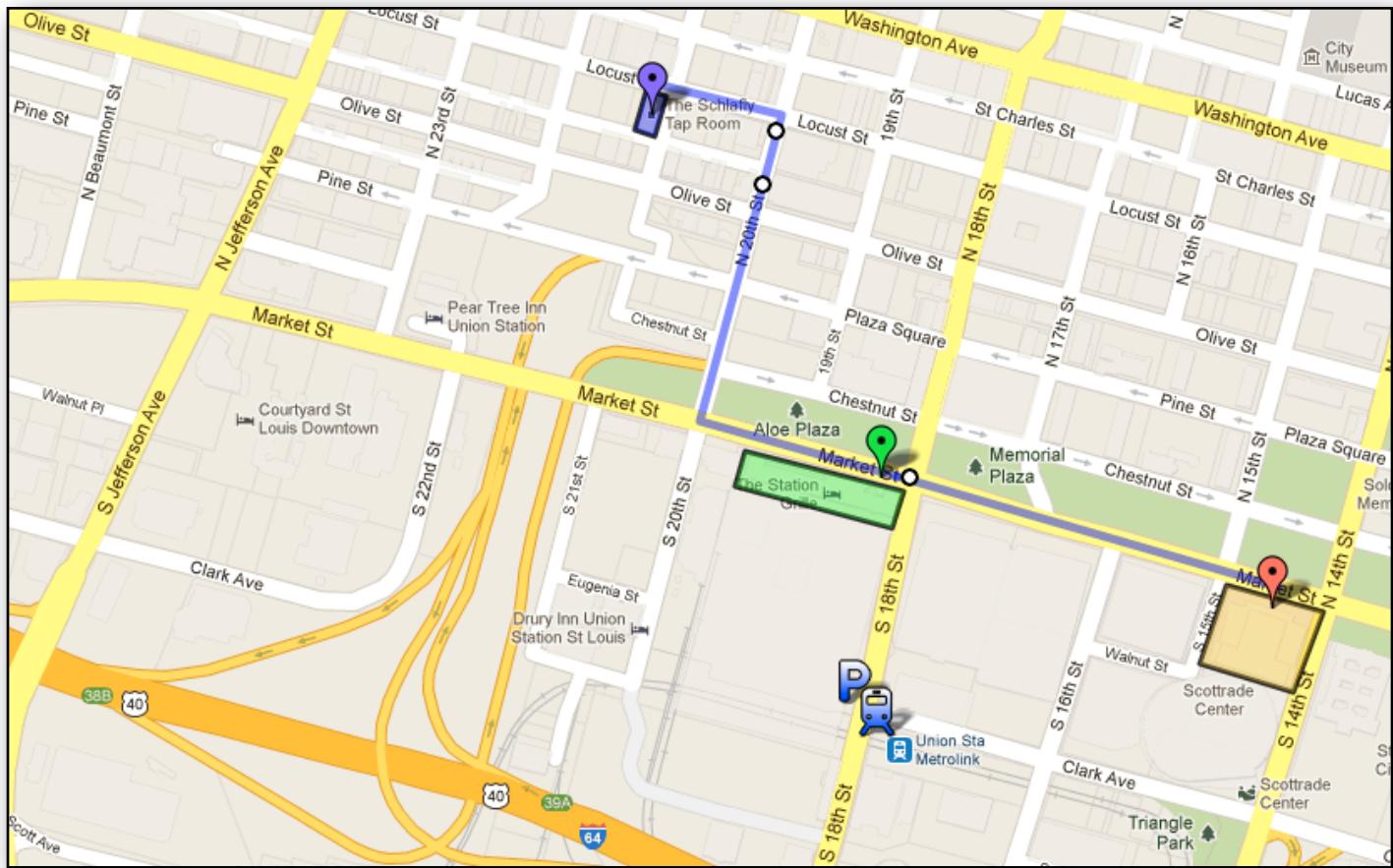
guidebook



Lambda Jam (<http://lambdajam.com>) is a new traveling conference for functional programmers. This conference that includes not just sessions but significant time for practice and learning with others. Lambda Jam will make its debut in Chicago, tentatively July 8-10, 2013.

Venue Map

[Interactive version](#)



Union Station Marriott

1820 Market St.
St. Louis, MO 63103

<http://unionstationmarriott.com>

Peabody Opera House

1400 Market St.
St. Louis, MO 63103

<http://peabodyoperahouse.com>

Schlafly Tap Room

2100 Locust St.
St. Louis, MO 63103

<http://schlafly.com/tap-room/>



Union Station Marriott

1820 Market St - Sunday preconf, Monday night unsessions



Schlafly Tap Room

2100 Locust St - Sunday night pre-party



Peabody Opera House

1400 Market St - Mon/Tues conference venue



Union Station Metrolink Station

Light rail from the airport



Union Station Parking Lot

Day and overnight parking \$7 for attendees



Walking directions to 1400 Market Street, St. Louis, MO

0.3 mi - about 6 mins



Walking directions to 2100 Locust Street, St. Louis, MO

0.4 mi - about 9 mins



Schlafly Tap Room



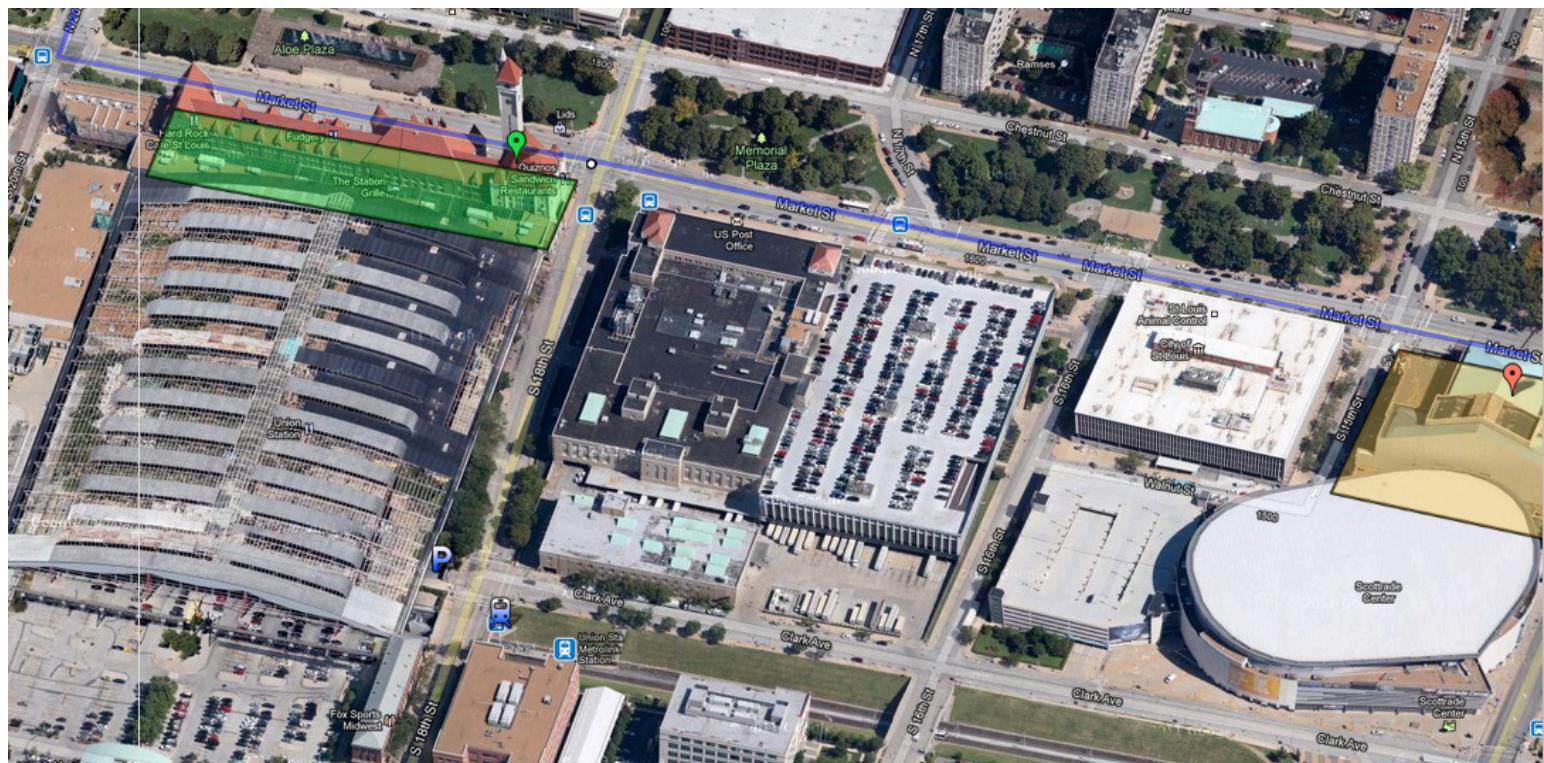
Union Station Marriott



Peabody Opera House

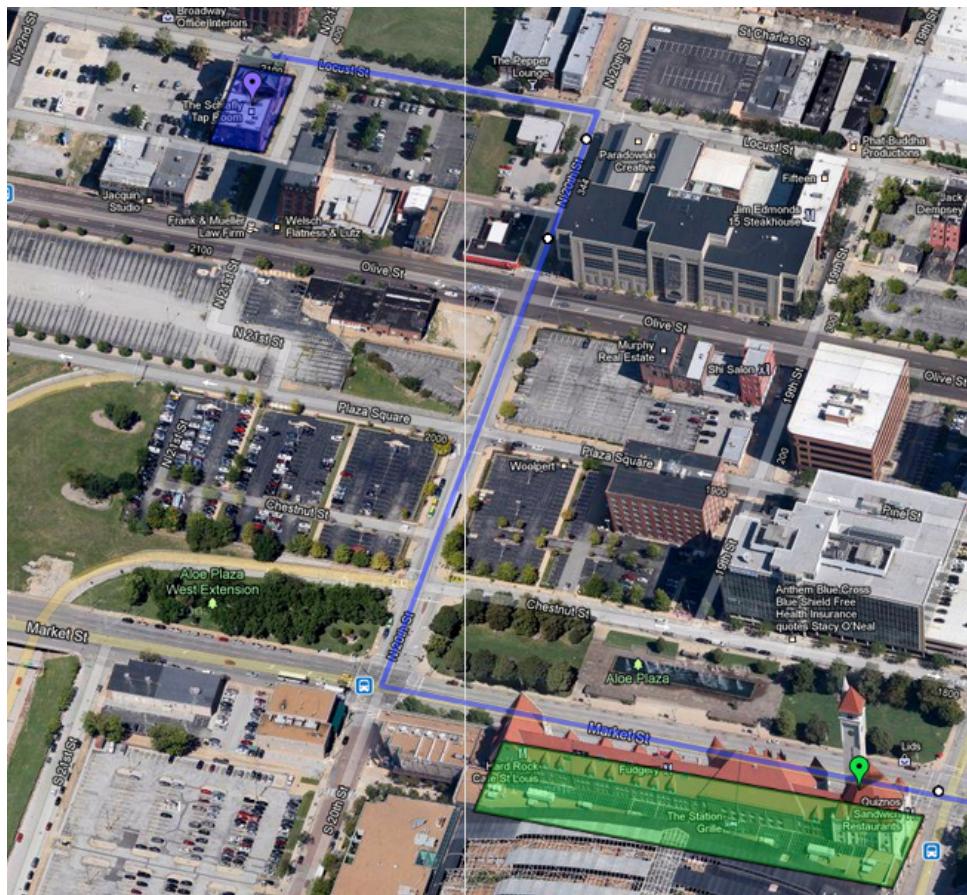
Union Station to Peabody Opera House

Go east from Union Station Marriott to the Peabody - look for the big stone bears!



Union Station to Schlafly Tap Room

Go north from Union Station up 20th St, then left on Locust to the Tap Room. Shuttles will be available Sunday 6-10 pm.



Sunday, Sept 23rd - Emerging Languages Camp

(prepaid registration required)

	Regency Ballroom A/B		Regency Ballroom A/B
9:30-10:00 AM	Symbiotic Languages: Transpiling into JavaScript - Jeremy Ashkenas	12:40-1:20 PM	The Reemergence of Datalog - Michael Fugus
10:00-10:30	Bandicoot: code reuse for the relational model - Ostap Cherkashin, Julius Chrobak	1:20-2:00	Roy - Brian McKenna
10:30-11:00	Elm: Making the Web Functional - Evan Czaplicki	2:00-2:40	Julia: A Fast Dynamic Language For Technical Computing - Stefan Karpinski
11:00-11:30	Plan: a new dialect of Lisp - David Kendal	2:40-3:20	Rust - David Herman
11:30-12:00	Clever, Classless and Free? - Håkan Råberg	3:20-3:50	Snack Break (Foyer)
12:00-12:40	Lunch in Regency C (Provided)		
		3:50-4:30 PM	Regency Ballroom A/B
		4:30-5:10	Grace: an open source educational OO language - James Noble
		5:10-5:50	Elixir: Modern Programming for the Erlang VM - Jose Valim
			Visi: Cultured & Distributed - David Pollak

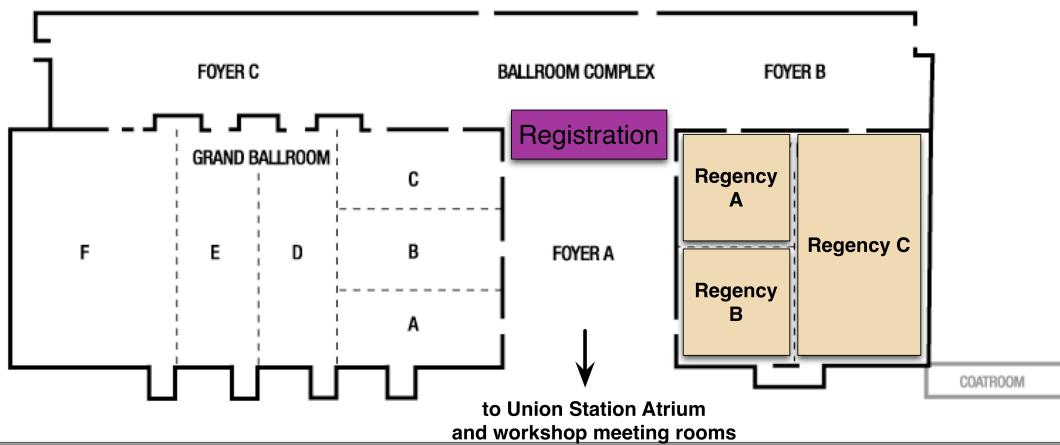
Sunday, Sept 23rd - Workshops

(prepaid registration required)

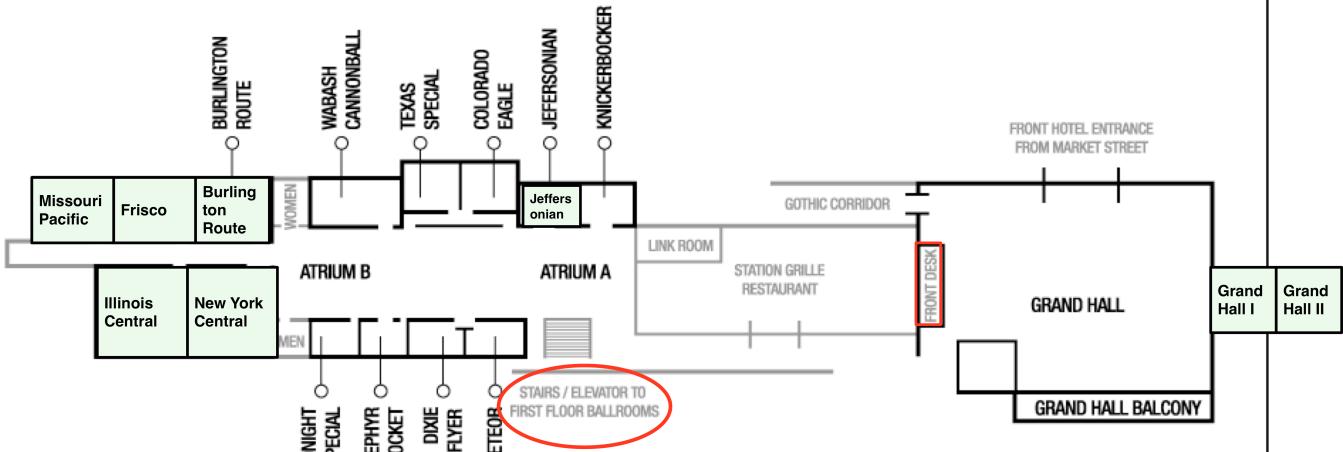
	Regency C						
11:30-12:30	Lunch (if pre-paid)						
12:30-3:00	Illinois Central Professional Javascript and Coffeescript development - Peter Bell	New York Central Introduction to Apache Hadoop - Tom Wheeler	Missouri Pacific Google Go: Fast, Lean, Scalable and Fun - Eleanor McHugh	Frisco GPU Programming Crash Course - Kashif Rasul	Burlington Route Applying Presentation Patterns: Sharpen Your Talks - Neal Ford	Jeffersonian Vim masterclass - Drew Neil	Grand Hall Meeting 1 Intro to Clojure - Stuart Halloway
3:00-3:30	Snack break Foyer						
3:30-6:00	Building next generation web apps using Backbone - Peter Bell	Machine Learning for Hackers - Drew Conway	Datomic: A New Database System - Stuart Halloway	Concurrent Programming Using The Disruptor - Trisha Gee	Using HBase - Jonathan Hsieh	Scalding for Data Analysis in Hadoop Systems - Dean Wampler	Making Your Website Mobile Ready - Scott Davis
5:30-10:00	Schlafly Tap Room All Conference Attendees	Shuttles will be available between Union Station and the Schlafly Tap Room from 6-10 pm.					

Union Station Marriott

FIRST FLOOR GARDEN SECTION



HEAD HOUSE CONFERENCE CENTER



Monday, Sept 24th - Peabody Opera House

	Union Station Marriott						
8:00 - 8:45	Breakfast						
8:45-9:00	Theater	Pepsi Encore	Midland States	Peabody Lounge	Centene Room	Kiel Club	
9:00-9:50	In Memory Databases - The Future is Now - Michael Stonebreaker						
10:00-10:50	Pontificating Quantification - Aaron Bedra, Daniel Spiewak	Monad examples for normal people in Python and Clojure - Dustin Getz	Understanding Indexing - Zardosht Kasheff	Concurrency in Android - G. Blake Meike	Get a Leg Up with Twitter Bootstrap - Howard Lewis Ship	Hallway track	
11:00-11:50	Functional Design Patterns - Stuart Sierra	Software Architecture using ZeroMQ - Peters Hintjens	Famous Unsolved Codes: Kryptos - Eltonka Dunin	Concurrency in iOS - Jeff Kelley	Renormalize - The sequel to SQL - Jack Orenstein, Ori Herrnstadt		
12:20-12:50	A Whole New World - Gary Bernhardt	What may not be expected in a country of eternal light - Noel Weichbrodt	Deconstructing P vs. NP (or why I hate sudoku) - Daniel Spiewak	As We May Do -- Augmented Reality and Computer Vision - Neil Milsted		Lunch 11:50 - 1:00 p.m.	
1:00-1:50	A Type-Driven Approach to Functional Design - Michael Feathers	Relational Programming in miniKanren	Postgres Demystified - Daniel Friedman, William Byrd	Designing voice navigated apps - Robert van Loghem	This is Your Workflow on Catnip - Bodil Stokke	Hallway track	
	The Higher-Order Rubyist - Robert Pitts			Zipkin: a distributed tracing framework - Johan Oskarsson	Doppio: Building a JVM in the Browser - Jez Ng, CJ Carey, Jonny Leahey		
2:00-2:50	ClojureScript: Better Semantics at Low, Low Prices! - David Nolen	A Taxonomy of Scala - Jamie Allen	Executing Queries on a Sharded Database - Neha Narula	Building an Impenetrable ZooKeeper - Kathleen Ting	Erlang Web Development with Yaws - Steve Vinoski		
2:50-3:30	Snacks - foyer sponsored by 						
3:30-4:20	Types vs. Tests: An Epic Battle? - Amanda Laucher, Paul Snively	Graph: composable production systems in Clojure - Jason Wolfe	Data Structures: The Code That Isn't There - Scott Vokes	Information Rich Programming with F# 3.0 - Donna Malayeri	Lessons from Erlang: Principles of Reliable Systems - Garrett Smith		
4:30-5:20	The Database as a Value - Rich Hickey	Real World Redis - David Czarnecki	River Trail - Parallel Programming in JavaScript - Stephan Herhut	F# for Trading - Phillip Trefford	Engineering Elegance: The Secrets of Square's Stack - Bob Lee		
5:30-6:20	Pushing the Limits of Web Browsers - Lars Bak						

Monday night unsessions

Monday night we have several rooms available for a series of unsessions suggested and chosen by you!

Where: Union Station Marriott

- Regency A
- Regency B
- Regency C
- New York Central
- Illinois Central

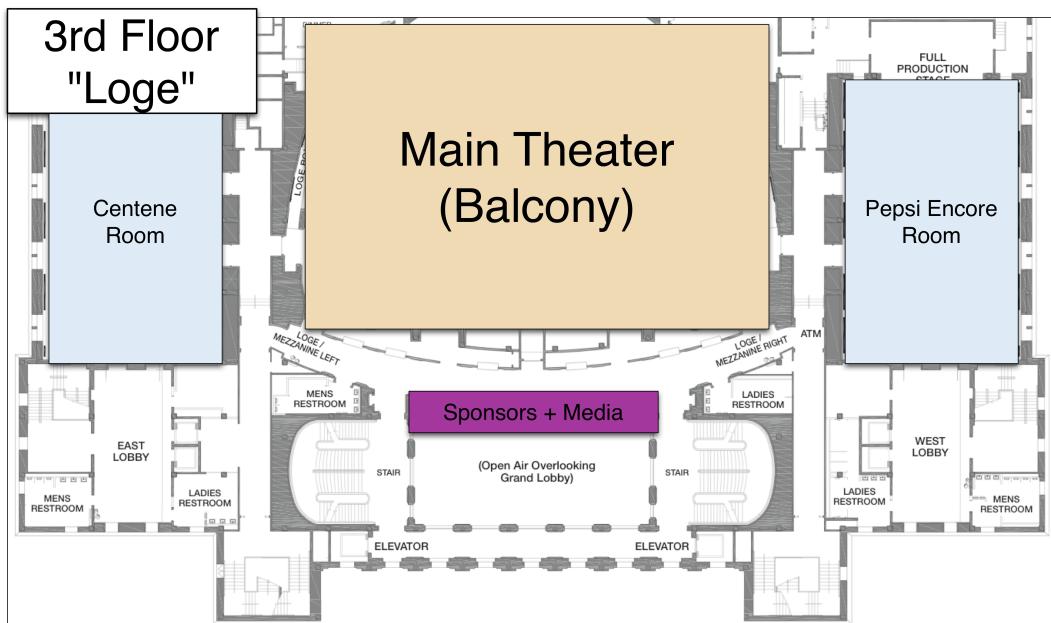
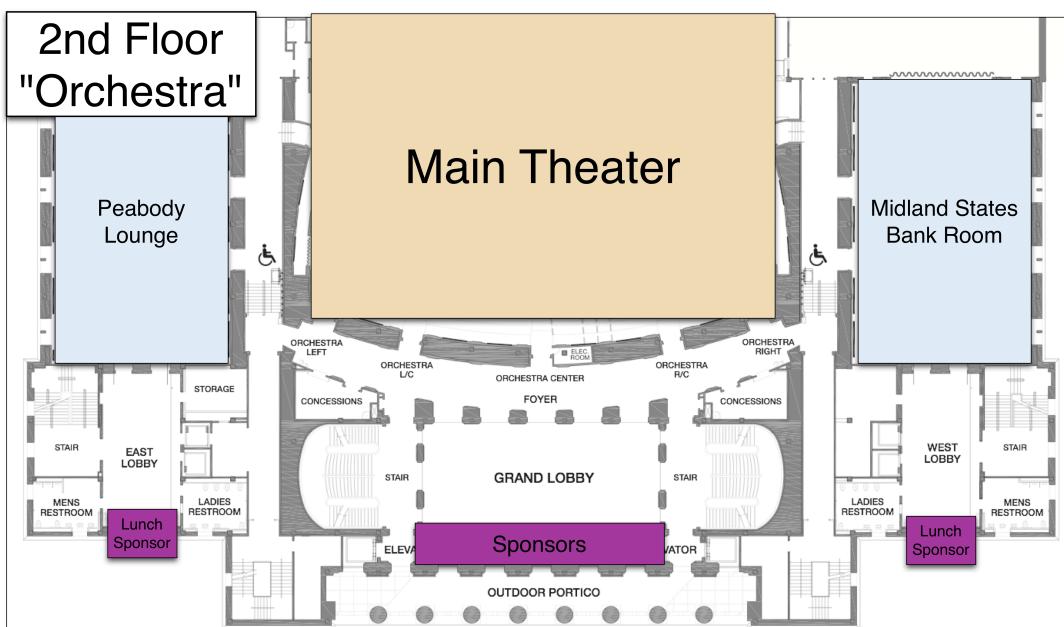
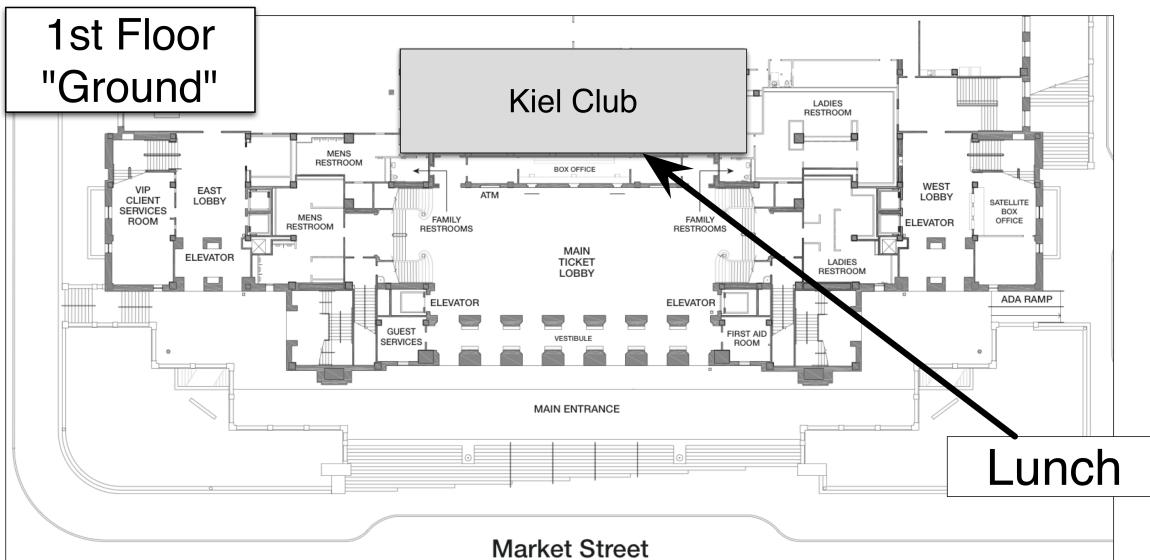
When: Monday, Sept 24, 7-10 pm

For more information on the schedule, please check out <https://github.com/strangeloop/strangeloop2012/wiki/Unsession-Schedule>

Tuesday, Sept 25th - Peabody Opera House

	Union Station Marriott					
8:00 - 9:00	Breakfast					
9:00-9:50	Theater	Pepsi Encore	Midland States	Peabody Lounge	Centene Room	Kiel Club
10:00-10:50	Computing Like the Brain - Jeff Hawkins					
11:00-11:50	Y Not? - Adventures in Functional Programming - Jim Weirich	Behind the mirror - Chris Granger	H2O: open source BigDatabase for interactive analytics - Cliff Click	Making the Web Faster at Google and Beyond - Ilya Grigorik	Making Hadoop Real Time with Scala and GridGain - Nikita Ivanov	Hallway track
12:10-12:50	Runaway complexity in Big Data... and a plan to stop it - Nathan Marz	The Racket Way - Matthew Flatt	Compiling Scala to LLVM - Geoff Reedy	Apache Cassandra Anti Patterns - Matthew Dennis	Cross-Browser Testing with BrowserStack - Scott González	
1:00-1:50	Humanity 2.0 - Matthew Taylor	Computer architecture of the 1960's - Carlton Mills	How Plato and Aristotle invented modern programming - Matt Butcher	Getting Physical: Networked Hardware with Node.js - Ted Hayes	Lunch 11:50 - 1:00 p.m.	
2:00-2:50	Eventually Consistent Data Structures - Sean Cribbs	Guess lazily! Making a program guess and guess well - Oleg Kiselyov	Clojure + Datomic + Storm = Your Personal Digital Secretary - Amit Rathore	Android App Anatomy - Eric Burke	Stop Using Native HTML5 - Kyle Simpson	Hallway track
2:50-3:30	The Audubon Society for Partial Failures - Cliff Moon	Up up and Out: Scaling software with Akka 2 - Jonas Boner	Wolfram's data analysis platform - Taliesin Beynon	ql.io: Putting Node.js to Work for Mobile - Subbu Allamaraju	AngularJS - HTML redesigned for web apps - Misko Hevery	
3:30-4:20	Expressing abstraction - Abstracting expression - Ola Bini	Go: code that grows with grace - Andrew Gerrand	Designing Fault Tolerant Distributed Applications - Scott Andreas	Building visual, data-driven UIs with ClojureScript - Kevin Lynagh	Ozma, an extension of Scala with Oz concurrency - Sébastien Doeraene	
4:30-5:20	Taking Off the Blindfold - Bret Victor	Project Lambda in Java 8 - Daniel Smith	Scaling scalability: Evolving Twitter Analytics - Dmitriy Ryaboy	Emerging Solutions for CSS Layout - Molly Holzschlag	The Fragment Transition - Corey Latislaw	
5:30-6:20	The State of JavaScript - Brendan Eich					

Peabody Opera House



Thank You Sponsors!

Without the support of our sponsors, this conference would not be possible. Below is a summary of the different sponsor categories and every Strange Loop sponsor. Please find more information about each sponsor in the mobile app, the conference web site, or the rest of this program. THANK YOU!

Platinum Sponsors



Gold Sponsors



Party Sponsors



Silver Sponsors



stripe



Lunch Sponsors



Snack Sponsors



Lanyard Sponsor



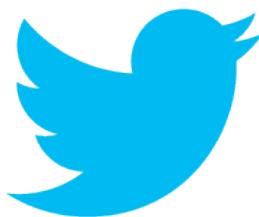
Student Sponsor



Supporting Sponsors



Oasis Digital Solutions Inc.



COUCHBASE

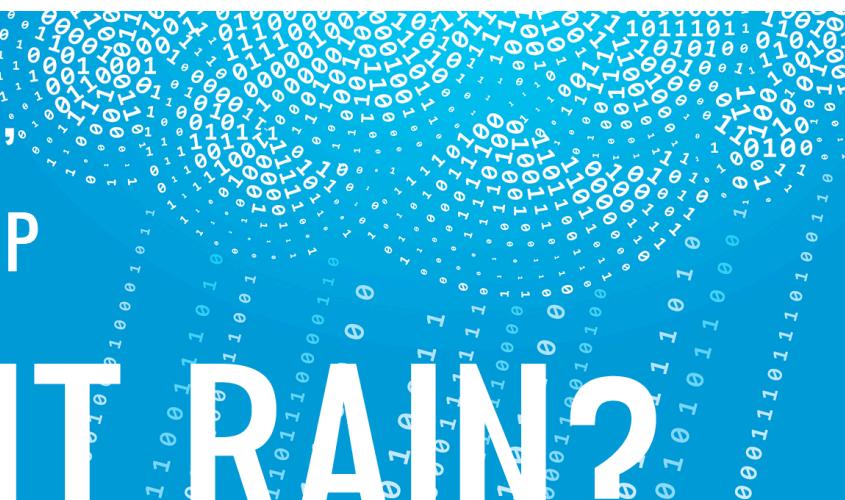
Media Partners



GOT A DISRUPTIVE,
INNOVATIVE,
SCALABLE STARTUP
THAT CAN

MAKE IT RAIN?

APPLY FOR AN ARCH GRANT & YOU COULD WIN



\$50,000

TO APPLY, VISIT ARCHGRANTS.ORG



**Thank you for being part of
Strange Loop 2012.**



You get lifetime access to ebooks you purchase through oreilly.com, and whenever possible we provide them to you in five DRM-free file formats—PDF, eBook, Kindle-compatible .mobi, DAISY, and Android .apk—that you can use on the devices of your choice. Our ebook files are fully searchable, and you can cut-and-paste and print them. We also alert you when we've updated your ebooks with corrections and additions.

Visit <http://oreilly.com/go/ebookrequest> to register for your free ebook.
Limit: one free ebook per attendee

© 2012 O'Reilly Media, Inc. The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. 12/02

O'REILLY®

Sunday, Sept 23rd

Emerging Languages Camp

9:30 - 10:00 AM

Symbiotic Languages: Transpiling into JavaScript

Location: Regency Ballroom A/B

More and more emerging "little" languages are targeting existing virtual machines -- the JVM, LLVM, and JavaScript -- instead of writing their own. Along with the reuse of an existing VM come new challenges. There are advantages in efficiency, conceptual simplicity, and inter-language interoperability to be had in sticking close to the "native" semantics of the target VM.

We'll talk about symbiotic languages in general, and CoffeeScript / JavaScript in particular, looking at the tradeoffs involved in implementing specific features that stray away from vanilla JavaScript. Is it necessary for every worthy new language to invent its own semantics out of whole cloth, or is there success to be found in sharing a partial semantics with your host platform?

Jeremy Ashkenas (@jashkenas) **The New York Times**

Jeremy Ashkenas works on the Interactive News team at the New York Times, and at DocumentCloud.org, helping news organizations analyze and publish the primary source documents behind the news. He created CoffeeScript, Backbone.js, Underscore.js, Docco and Jammit, among other opensource projects.

10:00 - 10:30 AM

Bandicoot: code reuse for the relational model

Location: Regency Ballroom A/B

The relational model proved to be a powerful tool for data manipulation tasks in the last 40 years. The key principles of data independence, flexible access paths, and a small set of predefined operators makes it an ideal candidate for building both simple and complex data centric systems.

Code reuse is an important feature of programming languages which allows developers to compose complex solutions from simple reusable units.

In this talk we present how Bandicoot implements a language which brings together the power of relational algebra and code reuse. It allows developers to write rich data manipulation code which stays readable and extendable.

The presentation will cover the following characteristics of the Bandicoot language:

- attributes sets
- code visibility
- computation reuse (relational variables and operators)

Ostap Cherkashin (@cherkaos)

Ostap is a co-founder and an active contributor to the Bandicoot project. He studied information security at Saint-Petersburg State Polytechnic University and his main interests are in systems programming and creating small fit-for-purpose systems. In his professional life he works as a software engineer in a large investment bank in Zurich.

Julius Chrobak (@julochrobak)

Julius Chrobak is a software engineer who studied math and informatics at the Comenius University in Slovakia. His interests are algorithms, complexity theory, and their applications. Julius has seven years of professional experience in building online transaction processing databases for both small and enterprise deployments. Besides his professional work he is also a co-founder of and an active contributor to the open source project Bandicoot.

10:30 - 11:00 AM

Elm: Making the Web Functional

Location: Regency Ballroom A/B

Elm aims to make web development functional in three ways. First, Elm avoids many of the headaches of web programming by compiling to HTML, CSS, and JavaScript which saves programmers from unpleasant APIs and cross-browser compatibility issues. Second, Elm is a functional language with all of the benefits of type-safety and type-inference. Finally, Elm's approach to GUI design makes it easy to create complicated components and interactions.

Elm is specifically designed for the creation of graphical user interfaces (GUIs). It is built around two paradigms: purely functional layout and Functional Reactive Programming. This talk is an overview of Elm's two major design choices and their consequences for GUI design.

1. In Elm, graphical elements - such as text, images, and video - are primitive values. They can be easily combined and composed to create complicated components. These primitive building blocks abstract away the unnecessary details of building a GUI, allowing the programmer to think on the level of components, not pixels. This is the essence of *purely functional layout*.
2. *Functional Reactive Programming* (FRP) is a relatively new approach to graphical user interfaces and animation. The key abstraction of FRP is the signal. Signals are time-varying values, carrying anything from integers to images. With signals, an animation is simply a signal of images. A GUI is a signal of graphical primitives. FRP cleanly models complicated user interactions by combining and transforming signals.

By grounding our discussion in real examples, we will see that these two features make it simple to create complex components and modular code. If you want to see Elm in action, check out the interactive editor and compiler online at "elm-lang.org":<http://elm-lang.org>.

Evan Czaplicki

Recently graduated from Harvard with a degree in Computer Science, I am now taking a break before I start as a software engineer at Google.

I am interested in making programming more accessible: easier to use and understand. My goals and design sensibilities have been influenced both by a formal study of programming languages and practical experiences as an intern at Google and Microsoft.

tumblr. ENGINEERING OPPORTUNITIES

PLATFORM TEAM LEAD
PRODUCT ENGINEER
CORE APPS ENGINEER
CONTENT ENGINEER
BACKEND SERVICE ENGINEER
iOS ENGINEER

WE MAKE STUFF
TUMBLR.COM/JOB

11:00 - 11:30 AM

Plan: a new dialect of Lisp

Location: Regency Ballroom A/B

Plan is a new dialect of Lisp. Its primary motivation is to fulfil the needs of the practical programmer: the hacker and the engineer. I will explain its philosophy, its origins, and what makes it different (and in my opinion, better) than Scheme, Common Lisp, Clojure, newLISP, etc. (The language is currently something of a testbed for design experiments, so what I talk about will probably change at some point. There is also not really a "good" implementation yet, for most values of "good.")

David Kendal (@dpkendal)

I'm a hacker and essayist from the United Kingdom. I'm currently preparing to move to San Francisco in mid-to-late April. I've programmed in a variety of languages over the years, from popular to unpopular to completely obscure. I know what I want in a programming language, and I've set out to make it.

11:30 am - 12:00 pm

Clever, Classless and Free?

Location: Regency Ballroom A/B

Shen is a portable programming language by Mark Tarver. Ruby is a less portable programming language by Matz. Both are very elegant.

shen.clj uses 750 LOC of Clojure to bootstrap Shen via "K Lambda", the small Lisp Mark designed for porting Shen. Shen itself is a portable version of Qi, which is tied to Common Lisp. This is hence a story of 3-4 Lisps.

Enumerable.java uses ~13k LOC to port a single module of Ruby, while adding Lambdas to Java 5 via a 2-4k LOC Java macro written using ASM, the bytecode toolkit. This is mainly a story of stubbornness and wishful thinking.

I plan to discuss the similarities, the drawbacks, and ultimate futility of both approaches to properly advance programming.

Hakan Raberg (@HakanRaberg)
Ghettojedi Software

Hakan Raberg has programmed professionally for over 15 years and unprofessionally close to 30.

He has plied his trade across the globe in Sweden, London, Bangalore and San Francisco - mainly via

ThoughtWorks. Today he works as an independent consultant in The City of London, specializing in euthanasia of legacy Java systems.

He's the author of shen.clj and Enumerable.java.

12:40 - 1:20 pm

The Reemergence of Datalog

Location: Regency Ballroom A/B

Datalog is a programming language derived from Prolog used for deductive rule definitions and queries. While certainly not a new concept, Datalog is uniquely positioned to serve as a query language for big data systems and as an embedded query language. In this talk I will give an overview of Datalog as a concept including its roots and costs/benefits over languages like Prolog and SQL. I will also provide examples of how Datalog is implemented and used in Datomic, Cascalog, and the Bacwn Clojure library.

Michael Fugus (@fugus)
Relevance Inc.

A programmer in the DC area specializing in artificial intelligence, compilation, code generation, and distributed simulation. A member of the Clojure/core team and author of "The Joy of Clojure".

1:20 - 2:00 pm

Roy

Location: Regency Ballroom A/B

JavaScript the Virtual Machine is everywhere - but JavaScript the Language isn't loved by everyone. Languages that compile to JavaScript (a.k.a. "altJS" languages) have recently surged in popularity to fill this gap.

This talk will introduce Roy, a statically-typed, functional, altJS language. It is heavily inspired by OCaml and Haskell but generates to plain, readable JavaScript.

Brian McKenna (@puffnfresh)
Atlassian

Learned to program by starting with JavaScript. Now calls himself a functional programmer and proponent of proofs as programs. Spends a lot of time trying to combine these distant worlds.

2:00 - 2:40 pm

Julia: A Fast Dynamic Language For Technical Computing

Location: Regency Ballroom A/B

Julia is a dynamic language in the tradition of Lisp, Perl, Python and Ruby. It aims to advance expressiveness and convenience for scientific and technical computing beyond that of environments like Matlab and NumPy, while simultaneously closing the performance gap with compiled languages like C, C++, Fortran and Java.

Most high-performance dynamic language implementations have taken an existing interpreted language and worked to accelerate its execution. In creating Julia, we have reconsidered the basic language design, taking into account the capabilities of modern JIT compilers and the specific needs of technical computing. Our design includes:

- Multiple dispatch as the core language paradigm.
- Exposing a sophisticated type system including parametric dependent types.
- Dynamic type inference to generate fast code from programs with no declarations.
- Aggressive specialization of generated code for types encountered at run-time.

Julia feels light and natural for data exploration and algorithm prototyping, but has performance that lets you deploy your prototypes.

Stefan Karpinski (@StefanKarpinski)
MIT

Stefan Karpinski is one of the co-creators and core developers of the Julia language. He is an applied mathematician and data scientist by trade, having worked at Akamai, Citrix Online, and Etsy, but currently is employed as a researcher at MIT, focused on advancing Julia's design, implementation, documentation, and community. He received an AB in Mathematics from Harvard in 2000, and is long overdue to finish his PhD in Computer Science at UCSB.

2:40 - 3:20 pm

Rust

Location: Regency Ballroom A/B

Rust is a new programming language being built at Mozilla for developing reliable and efficient systems software. Rust is designed to support concurrency and parallelism, for building platforms that take full advantage of modern hardware. Its static type

Processing
terabytes of data,
on 1000s of nodes,
with **in-memory speed**
and **database reliability**
using Java.



gridgain.com

system is safe and expressive, and provides strong guarantees about isolation, concurrency execution, and memory safety.

David Herman (@littlecalculist)
Mozilla

Dave Herman is a programming language propeller-head at Mozilla Research. He works on JavaScript, Rust, and a number of research projects in web technology.

3:20 - 3:50 pm

Break

Location: Regency Foyer

3:50 - 4:30 pm

Grace: an open source educational OO language

Location: Regency Ballroom A/B

We are engaged in the design of a new object-oriented educational programming language called Grace. Our motivation is frustration with available languages, most of which are approaching 20 years old. In this talk, I'll outline the principal features of Grace, discuss open issues, and listen to your reactions while all the choices are still on the table. In particular, I'll give some examples from the design process so far, showing how conceptually orthogonal design decisions all too easily end up as tightly coupled gordian knots. For more information see:<http://gracelang.org/>

James Noble (@jameskjx)
Victoria University of Wellington, New Zealand

James Noble is Professor of Computer Science and Software Engineering at Victoria University of Wellington, New Zealand. His research centres around software design. This includes the design of the users' interface, the parts of software that users have to deal with every day, and the programmers' interface, the internal structures and organisations of software that programmers see only when they are designing, building, or modifying software. His research in both of these areas is coloured by my longstanding interest in object oriented approaches to design, and topics he has studies range from aliasing and object ownership, design patterns, agile methodology, via usability, visualisation and computer music, to postmodernism and the semiotics of programming.

4:30 - 5:10 pm

Elixir: Modern Programming for the Erlang VM

Location: Regency Ballroom A/B

The web is changing, not only how we build and scale applications (NoSQL) but also how people use our applications (HTML5 and Native clients). And those changes will, in the long term, affect how we develop our app's backend. Sometimes developers will require a low level socket API for synchronization, other times developers will need a persistent connection to the client via websockets. Under this perspective, the Erlang VM has shown throughout time it provides a great foundation for handling such scenarios.

Elixir is a programming language built on top of the Erlang VM. Elixir attempts to provide better abstractions and productivity tools like protocols and macros usually required for web development while keeping the Erlang VM semantics to build distributed, fault-tolerant applications.

In this talk, Jose Valim will introduce Elixir and discuss the goals and design decisions behind it. He will also introduce a high-performance web framework called Dynamo built on top of Elixir and show how Elixir is ready to tackle the upcoming challenges in server side development.

Jose Valim (@josevalim)
Plataformatec

Jose Valim (@josevalim) is the lead-developer of PlataformaTec, member of the Ruby on Rails Core Team and author of Crafting Rails Applications. Software developer for 8 years, he graduated in Engineering by the Sao Paulo University and has a Master of Science by Politecnico di Torino, in Italy. He is an active member in the Open Source community.

5:10 - 5:50 pm

Visi: Cultured & Distributed

Location: Regency Ballroom A/B

When designing a new system, whether it be a computer language or anything else that's created to draw a large audience, it's critical to figure out what's important. What are the priorities for the project? How does the project create and reinforce a culture, a society, a social structure that furthers the goals of the project.

Visi is a modeling language targeted at Excel and PHP users. Borrowing the best of various language cultures to create a self-reinforcing mechanism that will grow the language and community in the right direction.

- Excel/1-2-3/VisiCalc -- Open sharing of models. In Visi this will be done via Git.
- Smalltalk/Squeak -- Self-hosted environment encourages "playing" and growing.
- Java -- Great documentation via JavaDocs. Visi is literate... models are Markdown documents.
- Ruby -- Tests.

In this presentation, David Pollak will discuss strategic goals for Visi and how the Visi language and Visi environment will create cultural structures designed to grow that Visi community and allow the community to drive Visi growth, evolution, and adoption.

David Pollak (@dpp)
Visi.Pro

David Pollak founded Visi.Pro, Cloud Computing for the Rest of Us along with the Visi Language open source project. David founded the Lift Web Framework and continuously contributes to Lift.

David wrote Beginning Scala and ran the first Scala-focused conferences. In the past David wrote software for NextStep (which has morphed into iOS) including Mesa, the world's first real-time spreadsheet. You can find David on Twitter @dpp and IRL finding good Pho restaurants or walking Archer his dog along with the rest of his family in San Francisco.

Workshops

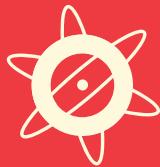
12:30 - 3:00 PM

Professional Javascript and Coffeescript development

Location: Illinois Central

Love it or hate it, with richer client frameworks like backbone, spine and knockout and with server side frameworks like node.js, if you are a professional web developer, you are probably going to find yourself writing some Javascript in the future.

It's also an interesting language. It has been described as a love child between Scheme and Java (with elements of Self and awk), it has prototypical rather than classical inheritance, and it has some



ATOMIC OBJECT

Give a shit.
Make great things.
Love your job.

atomicobject.com

really nice functional programming capabilities (along with some really horrible design decisions).

In two and a half hours we're going to do a whirlwind, hands on tour of the language, teaching you the skills required to become a competent Javascript developer. We'll also briefly touch on Javascript testing using Jasmine and give a sneak peak into Coffeescript - a thin layer over javascript which reduces the syntactic noise, fixes some of the worst design decisions and even adds classes.

Peter Bell (@peterbell) **PowWow**

Peter Bell is CTO and lead developer of PowWow, a lean startup @Dogpatch Labs, NYC.

He presents regularly on javascript, NoSQL, cloud computing, software craftsmanship, ruby, groovy, and requirements and estimating. He has presented at DLD conference, ooPSLA, SpringOne2GX, Code Generation, Practical Product Lines, the British Computer Society Software Practices Advancement conference, cf.Objective(), CF United, Scotch on the Rocks, WebDU, WebManiacs, RubyNation, DevNexus, UberConf, the Rich Web Experience and the No Fluff Just Stuff Enterprise Java tour.

He has been published in IEEE Software, Dr. Dobbs, IBM developerWorks, Information Week, Methods & Tools, Mashed Code, NFJS the Magazine and GroovyMag. In New York he organizes the local node.js meetup and co-organizes CTO school.

Intro to Clojure

Location: Grand I

Clojure is a new dynamic programming language for the Java Virtual Machine. Clojure introduces innovative ideas around state management and concurrency, while inheriting the best ideas from the long history of Lisp-like languages. It is a language designed to solve real problems, some of which are so pervasive in current programming practice that we don't even recognize them as problems. This workshop will provide an introduction to Clojure's syntax and key concepts, with examples of how they can make a real impact in large, complex systems.

Stuart Halloway (@stuarthalloway) **Relevance, Inc**

Stuart Halloway, the author of Programming Clojure, is a co-founder of Relevance, Inc. With over 20 years of experience as a programmer, CTO, and CEO, Stuart has built software systems in finance, health care, contract management, online retail, manufacturing, education, and security automation.

Applying Presentation Patterns: Sharpen Your Talks

Location: Burlington Route

Influence It.

Many technical decisions are guided by an inspiring and convincing talk. Though developers like yourself often invest in sharpening coding skills, equal time should be given to tuning your ever-important communication skills. It's easy, with this bit of design mentoring and content guidance, to have a more noticeable impact on the choice of tools, processes, libraries and languages on projects to which you contribute via your engaging presentations.

Study It.

We'll bet you have such an awesome technical talk already running around in your head. But do you know how to express such a talk in a compelling story line with rich and useful visuals? Do you know when to employ and when to avoid certain educational "devices" within your show? Many presentations on presentations aim to inspire in the realm of design or story telling, but in contrast, this workshop will provide concrete actions and individual mentoring to amp up your talk while keeping the purity of your educational objective.

Do It.

You'll finish this workshop with a sharp and information-rich presentation on a topic of your choosing that will be ready to share next week back at your office. We're confident that with the application of our easy-to-follow patterns, you'll deliver your compelling new talk in your now-impressive style.

Neal Ford (@neal4d) **ThoughtWorks**

Neal Ford is Software Architect and Meme Wrangler at ThoughtWorks, a global IT consultancy with an exclusive focus on end-to-end software development and delivery. He is also the designer and developer of applications, instructional materials, magazine articles, courseware, video/DVD presentations, and author and/or editor of 6 books spanning a variety of technologies, including the most recent The Productive Programmer. He focuses on designing and building of large-scale enterprise applications. He is also an internationally acclaimed speaker, speaking at over 250 developer conferences worldwide, delivering more than 1000 talks. Check out his web site at nealford.com. He welcomes feedback and can be reached at nford@thoughtworks.com.

Vim masterclass

Location: Wabash Cannonball

Learn to wield the awesome text editing power of Vim in this hands on workshop. We'll drill through a series of exercises that are designed to teach the best practices for working with Vim's core functionality. You'll learn to slice and dice text at the speed of thought.

"There are more ways of killing a cat than choking it with cream" Charles Kingsley

In Vim, we can duplicate a line from normal mode, insert mode, or with an Ex command. But which is best? In this masterclass, we'll work through a series of exercises, finding at least two solutions to every problem. In response to the question "which is best?" we'll see that the answer is always: "it depends".

Vim is optimized for repetition. We'll study a few examples of how to use the dot command to repeat the last change. Then we'll develop a strategy for composing repeatable changes, and meet the optimal Dot Formula: a two step solution with a thousand uses. We'll also study best practices for creating robust macros, and learn how to execute them either in series or in parallel.

To level the playing field, we'll be using bare Vim (not vi - we're not savages). You'll be asked to leave your .vimrc at the door.

Drew Neil (@nelstrom) **Studio Nelstrom**

Drew Neil, the "voice of Vimcasts" (<http://vimcasts.org/>), is an independent web developer, teacher, and author of "Practical Vim" (<http://pragprog.com/book/dnvim/practical-vim>).

Google Go: Fast, Lean, Scalable and Fun

Location: Missouri Pacific

This workshop will explore Google Go with an emphasis on writing the kind of high-productivity code which is typical of dynamically typed languages whilst retaining the precise control over runtime performance and resource usage that static type checking enables.

The treatment will be divided into several sections, starting with an introduction to Go's standard toolchain then moving through various practical aspects of syntax, memory model and type system to build familiarity with the language. Here we'll meet the distinction between reference and value types, slices and maps, and the user-defined types which form the basis of Go's object model. This will lead naturally to Go's interface types and the



When it comes to
managing high
velocity data,

ONE SIZE DOES **NOT** FIT ALL

VoltDB allows companies to make real-time decisions that are faster and more informed than ever before

- ✓ Analytics on Real Time Data
- ✓ High Velocity Data Sources
- ✓ Realtime Decisioning
- ✓ Cloud Ready
- ✓ ACID Compliant

VoltDB

The NewSQL database for high velocity applications

Web: voltdb.com | Twitter: @voltdb | LinkedIn: company/voltdb | Facebook: VoltDB

method-based type inference which forms the basis of duck typing.

We'll complete our coverage of Go's type system by studying first-class functions, closures and goroutines along with the CSP concurrency model they support. This will demonstrate that Go is a powerful general purpose programming language suited to writing flexible, scalable, maintainable applications for modern multi-core and distributed environments.

For the final section we'll discuss integrating Go code with existing C libraries and with assembly language, leading into a more general discussion of Go's standard tools for testing and performance benchmarking.

By the end of the workshop you should have a reasonable grasp of how Go stacks up for real-world development and a firm foundation from which to explore further.

Eleanor McHugh @feyeleanor

British hacker Ellie has a fascination with language design and a passion for all things low-level. For many years an active member of the Ruby scene, she's also amongst the first generation of Go adopters and a passionate explorer of its less-discussed corners.

When not coding Ellie enjoys on-line gaming, polyhedral dice, home brewing and gothic music.

Introduction to Apache Hadoop

Location: New York Central

It's a vicious circle: as companies recognize the value of the data they generate, they become more successful, and as they become more successful, they generate more data. Eventually they find their existing systems struggle with the "big data" they've produced. Fortunately, many are learning what the major Web and analytics companies have known for years -- "Apache Hadoop"<http://hadoop.apache.org/> is an effective solution for large-scale data storage and processing.

Originally inspired by Google's "GFS":<http://research.google.com/archive/gfs-sosp2003.pdf> and "MapReduce":<http://research.google.com/archive/mapreduce-osdi04.pdf> papers, Apache Hadoop is an open source framework offering scalable, distributed, fault-tolerant data storage and processing on commodity hardware.

"Introduction to Apache Hadoop" presents a solid introduction with three goals. It will explain what Apache Hadoop is, why it's relevant and how it works. This session requires no previous experience with Apache Hadoop or MapReduce and is

intended to appeal to a broad audience of any technical background.

In addition to learning Hadoop's architecture and fundamental concepts like HDFS and MapReduce, you'll gain practical advice for using this technology effectively. The session will also include an overview of the Hadoop ecosystem (Hive, Pig, HBase, Flume, Oozie and Sqoop), with particular emphasis on how these tools integrate with your existing systems.

Tom Wheeler Cloudera, Inc.

Tom Wheeler's career spans more than fifteen years in the communications, biotech, financial, healthcare, aerospace and defense industries. Before joining Cloudera in 2011, he was a software engineer, technical trainer and occasional system administrator. Ten years ago he worked on the design and implementation of a scalable, fault-tolerant, distributed data processing system. He's overjoyed that Apache Hadoop has since been invented, as this means nobody else need ever suffer this way again.

GPU Programming Crash Course

Location: Frisco

This course is for developers who want to learn how to program and utilize the parallel computing power of the Graphics Processing Unit (GPU) using NVIDIA's CUDA programming framework and, time permitting, OpenCL (although the many of the basic concepts are very similar).

The course will start by introducing the concepts of general purpose GPU programming and go into the process of installing and setting up the development environment on the 3 OS's that support CUDA. We will also talk about the different language bindings for languages like Java, Python and Ruby.

The main gist of the course will involve learning the concepts of CUDA memory management together with the hardware capability of the GPU we are developing on.

Once we are familiar with the core concepts, we will talk about interoperability of the CUDA library with rendering and also the use of atomic primitives to accomplish things which are quite trivial in the traditional CPU case. Then we will talk about the concept of CUDA streams.

We will talk about the different external libraries both 3rd party as well as provided by NVIDIA optimized for the GPU, that implement many useful algorithms for applications ranging from Finance to Medical Imaging and Machine Learning.

Finally we will end the course by talking about GPUs in the cloud as a service and multi-GPU APIs.

Kashif Rasul (@krasul) SpacialDB

Kashif has a PhD. in Mathematics from FU Berlin and is currently working for a startup in the area of geospatial and high performance computing. He has presented at NVIDIA's GTC in 2009 and is also helping with the Java CUDA bindings.

Prior to finishing his PhD. he has worked at the Max Planck Institute in Golm, Germany and at Visage Imaging in Berlin as a software developer in the area of Numerical Computing and Medical Imaging.

Kashif also contributes to a number of Open Source projects and you can follow him on [Github](<http://github.com/kashif>) or twitter [@krasul](<http://twitter.com/krasul>).

Core HTML5 Canvas

Location: Grand 2

In the early 2000's, during the Dark Ages of software development, developers toiled away implementing simplistic and boring applications known as web applications. Because web browsers were not capable of producing the rich graphics and user experiences associated with desktop applications, web applications did little more than present users with unimaginative forms and subsequently process those forms on the server. After that banal activity, the process would begin again, with the application presenting yet another form to the tireless user.

Fortunately, there has been a Renaissance after the Dark Ages. Today, developers are implementing beautiful web applications with rich user experiences, thanks to the collective group of specifications known as HTML5. And one of the most exciting of those technologies is HTML5 Canvas, which provides all of the graphics horsepower you need to implement anything you can imagine, from image filters to word processors and arcade games.

This workshop is a code-fueled, no-nonsense, deep dive into HTML5 Canvas. Taught by the author of Core HTML5 Canvas published by Prentice-Hall in 2012, this workshop will show you how to implement lots of cool stuff in the browser with Canvas, such as image filters, magnifying glasses, paint applications, animations, and video games.

You'll also see how to use Canvas to implement custom controls that you can use in any HTML5 application, and you'll see how to get your Canvas-based applications running on mobile devices, such as iOS5, including how to make your HTML5 application look exactly and behave exactly like an iOS5 native application.

Listen to your data.

splunk®

www.splunk.com/goto/listen

David Geary (@davidgeary)
Clarity Training, Inc.

David Geary is the author of Core HTML5 Canvas to be published by Prentice-Hall in April, 2012.

David is a prolific, passionate, and entertaining public speaker. He has spoken at hundreds of conferences worldwide, including JavaOne, StrangeLoop, The Server-side Symposium, Devoxx, and JAOO. David was a regular speaker for six years on the No Fluff Just Stuff tour, averaging five 90-minute sessions at more than 120 conferences. David is a three-time JavaOne Rock Star.

David has written eight Java books, four of which are best-sellers in their respective categories, including one of the best-selling technical book series of all time: Graphic Java.

3:00 - 3:30 PM

Snack Break

3:30 - 6:00 PM

Building next generation web apps using Backbone

Location: Illinois Central

First there was the client-server request/response and it was good. We then added AJAX requests to improve usability and it was better. But anyone who has worked on an AJAXified website for a while knows that eventually you end up with callback hell trying to keep all of the view elements updated as the model state changes. With client side MVC frameworks, we can bring the kind of separation of concern and DRYness that MVC frameworks have provided on the server side for years.

In this class we'll start by looking at the problems caused as AJAXified websites become more sophisticated and then we'll build an application using backbone.js - one of the most popular and capable client MVC frameworks. Backbone is more of a starting point than a complete solution, so we'll also go through a lot of the best practices that are evolving for developing DRY, maintainable web applications using backbone.

Whether you're an experienced front end developer or are just interested in keeping up with architectural trends in web applications, this class will be a chance to explore a technology that is becoming increasingly common in new web applications. We'll also briefly touch on other client-

MVC frameworks including spine, sammy, angular, knockout and batman to provide an overview of their relative strengths and weaknesses for different classes of projects.

Peter Bell (@peterbell)
PowWow

Peter Bell is CTO and lead developer of PowWow, a lean startup @Dogpatch Labs, NYC.

He presents regularly on javascript, NoSQL, cloud computing, software craftsmanship, ruby, groovy, and requirements and estimating. He has presented at DLD conference, ooPSLA, SpringOne2GX, Code Generation, Practical Product Lines, the British Computer Society Software Practices Advancement conference, cf.Objective(), CF United, Scotch on the Rocks, WebDU, WebManiacs, RubyNation, DevNexus, UberConf, the Rich Web Experience and the No Fluff Just Stuff Enterprise Java tour.

He has been published in IEEE Software, Dr. Dobbs, IBM developerWorks, Information Week, Methods & Tools, Mashed Code, NFJS the Magazine and GroovyMag. In New York he organizes the local node.js meetup and co-organizes CTO school.

Using HBase

Location: Burlington Route

Why should you care about HBase? Especially if you already know and use Riak/MongoDB/Cassandra/etc?

Well, HBase is inspired by Google's battle-hardened "BigTable" architecture and is known to be one of the most scalable distributed databases around. For some perspective, the largest Riak and MongoDB clusters in production are measured in the dozens of nodes - while HBase clusters with hundreds and thousands of nodes aren't unusual.

Even if you aren't running a Top 100 site (yet), HBase could still be very useful to you since it's strongly consistent and easy to reason about.

However, for all of HBase's strengths, it is difficult to get started with. Instead of just running one daemon on each server in your cluster, you'll have the fun of configuring HDFS (which requires you to setup a Namenode, backup Namenode, and datanodes), Zookeeper quorum (which should be 3 dedicated servers), and HBase itself (which will require a master, backup master, and regionservers). And that's assuming you don't want MapReduce or security features (which each require several more servers).

After this workshop you will:

- Have a high level understanding of HBase's internal architecture
- Setup a running HBase server on your laptop (if it runs Mac or Linux)

- Learn how to setup and benchmark a production-ready HBase cluster from scratch
- Learn the fundamentals of HBase's client APIs and use them to build a link shortener on HBase
- Learn how to configure and tune HBase to work with your hardware for your needs (write heavy workload? lots of random reads?)
- Learn about common HBase issues, and debug a real world production problem

After this class, you'll be able to setup, manage, and use a HBase cluster. And, more importantly, be able to decide for yourself when HBase is the the right tool for the job.

Jonathan Hsieh (@jmhsieh)
Cloudera

Jonathan is a Software Engineer with Cloudera, currently focused on the Apache HBase project. He is an Apache HBase committer and PMC member, as well as a committer on the Apache Sqoop project, and a committer and founder of the Apache Flume project. Jonathan has an M.S. in Computer Science from University of Washington and also has an M.S. and a B.S. in Electrical and Computer Engineering from Carnegie Mellon University.

Concurrent Programming Using The Disruptor

Location: Frisco

The Disruptor is an open source concurrent programming framework developed by LMAX Exchange, a financial exchange based in London.

The most interesting thing about it is how the Disruptor has promoted discussions about approaches to writing high performance code, and shown that Java is a serious contender in this space.

Contrary to the current trend of hiding multi-threaded concerns behind languages or frameworks, the Disruptor provides a way to do quite the opposite - to enable developers to think about how to parallelise their architecture in a straightforward and easy to code fashion. In this workshop, Trisha Gee from LMAX Exchange will show examples of how to use the Disruptor to share data between threads, and walk you through how to create your own application using the Disruptor.

Trisha Gee (@trisha_gee)
LMAX Exchange

Trisha is a developer at LMAX Exchange*. She's been working in financial markets for the last 6 years or so, but a fear of boredom and healthy amount of job-hopping before then has given her a wide breadth of experience, in a range of industries,

over the 10+ years she's been a professional developer. Trisha is heavily involved in the London Java Community and the Graduate Developer Community, she believes we shouldn't all have to make the same mistakes again and again.

The award-winning LMAX Exchange is the revolutionary venue for FX trading. Based in London, LMAX Exchange is authorised and regulated by the FSA in the UK. LMAX Exchange operates an MTF* (Multilateral Trading Facility), with all trades cleared through prime brokers.

Machine Learning for Hackers

Location: New York Central

Machine learning exists at the intersection of traditional mathematics and statistics with software engineering and computer science. While machine learning was once a discipline open only to professional researchers with access to massive data sets and high-performance computing environments, the proliferation of high-quality open source libraries and low-cost cloud computing services has provided the opportunity for anyone to leverage these methods to better understand their data. This has brought machine learning to the hacker community, but the black-box nature of popular machine learning libraries and tools means that many would-be users have little understanding of how these tools work under the hood. In this workshop, Drew Conway and John Myles White will introduce machine learning from a hacker's perspective, focusing on algorithms and examples rather than theoretical or mathematical structure. The workshop will be broken into two sections:

1. We will begin with an introduction to using the R programming language for performing machine learning. We will use an example data set of UFO sighting in the United States to review topics such as loading, cleaning, organizing, analyzing and visualizing data.
2. In the second-half we will introduce one of the primary tools in machine learning: linear regression. Here we explore data whose relationship roughly approximates a straight line, which we can use to make predictions about unobserved data. For this we will attempt to predict the number of page views for the top 1,000 websites on the Internet as of 2011.

Drew Conway (@drewconway)
New York University

Drew Conway is a PhD candidate in the Department of Politics at New York University, and is currently serving as the Scientist-in-Residence at IA Ventures. His scholarly research primarily focuses on the modeling of social systems and the emergent behavior of groups of people. Drew

applies this research in political science to gain a better understanding of the dynamics of groups engaged in illicit activity; particularly, terrorist and criminal organizations.

John Myles White (@johnmyleswhite)
Princeton University

John Myles White is a Ph.D. candidate in the Department of Psychology at Princeton University, where he studies human decision-making using theoretical models and experimental techniques. His work combines mathematical models and machine learning techniques to discover patterns in how we choose what's best for us.

Scalding for Data Analysis in Hadoop Systems

Location: Wabash Cannonball

Scalding is a Scala **DSL** for **Cascading**, a widely-used Java API for data analysis in Hadoop clusters. Scalding was recently open-sourced by Twitter. Cascading provides higher-level data flow abstractions that hide many of the low-level complexities in Hadoop's Java API, thereby accelerating application development. Scalding exploits the functional-programming features and elegant DSL support in Scala to allow developers to write concise Cascading programs with a syntax that fits data analysis and transformation in a natural way. Scalding is comparable to **Casalog**, a *Clojure*-based DSL for Cascading.

This hands-on workshop introduces Scalding using examples of typical data analysis problems. Scala syntax is explained as needed. We'll briefly compare Scalding to other high-level language options in common use, such as Casalog, Hive, and Pig. We'll make the case that functional-programming idioms are a natural fit for working with data.

Dean Wampler (@deanwampler)
Think Big Analytics

Dean Wampler is the author of "Functional Programming for Java Developers", the co-author of "Programming Scala", and the co-author of the forthcoming "Programming Hive" (all from O'Reilly). He is a Principal Consultant for Think Big Analytics, specialists in "Big Data" application development, primarily using Hadoop-related technologies. Dean is the founder of the Chicago-Area Scala Enthusiasts (meetup.com/chicagoscala/) and the programming web site polyglotprogramming.com. He is also a contributor to several open-source projects.

Making Your Website Mobile Ready

Location: Grand I

Most people are shocked at how easy it is to adapt a traditional desktop-oriented website design for smart phones and tablets. In this hands-on workshop, Scott Davis shows you how to move away from separate m.com and www.com development efforts to a single, unified, responsive website that looks good on any device.

The mobile web is here, and this workshop aims to help you get your website up to speed. You can make some subtle tweaks to your existing website to make it more mobile friendly -- accounting for the smaller viewport screen size, providing layouts for both portrait and landscape mode, and making your hyperlinks mobile-friendly. Or you can optimize it a lot by providing a dedicated website that leverages HTML5 local storage, application cache, and more.

As the mobile ecosystem continues to grow and fragment with devices from smartphones to tablets -- running iOS, Android, and everything else -- having a coherent mobile web strategy can help smooth out the bumps and move you from a "one size fits all" mentality to a more adaptive, right-sized approach to web development.

Scott Davis (@scottdavis99)
ThirstyHead.com

Scott Davis is the founder of ThirstyHead.com, a training and consulting company that specializes in leading-edge technology solutions like HTML 5, mobile development, SmartTV development, NoSQL, Groovy, and Grails.

Scott has been writing about web development for over 10 years. His books include Getting Started with Grails, Groovy Recipes, GIS for Web Developers, The Google Maps API: Adding Where to Your Web Applications, and JBoss at Work. Scott is also the author of two popular article series at IBM developerWorks -- Mastering Grails and Practically Groovy.

HTML5 and Open Web Hot Topics

Location: Grand 2

As HTML5 is still under development and implementations across browsers seems arbitrary, it becomes up to development teams to stay with the important issues of the day. In this 3 hour workshop, join Molly as she'll cover the following topics of interest, offering up strategies and solutions along the way:

- The Codec Wars - Has h.264 won the day?

What is “*the* **JBoss** **Way**”?

Simple.

Modern.

Productive.

[Learn More >](#)

- Managing security issues in HTML5 and related APIs (Web Sockets, caching, and storage)
- The Open Web Technology Stack - detailed vision of the Open Web
- Hybrid Applications: Current solutions for sustainability and scalability

Molly Holzschlag (@mollydotcom)**Molly.Com, Inc.**

Molly E. Holzschlag has spent the majority of her 25 years in IT working with the Internet, and observed the Web emerge from its first days at CERN. She has spent the years since educating herself and others about the Web, its languages, and its core philosophies. Warm, colorful, and interested in everyone's stories and successes, be sure to come say hello!

Datomic: A New Database System

Location: Missouri Pacific

Datomic is a new distributed database built for the cloud. It deconstructs the traditional database server, separating storage, query and transaction capabilities and scaling them independently. This workshop introduces Datomic's distributed architecture, Datalog-based query language, ACID transactions and transaction notifications, database functions, flexible data model with built-in support for time, its data-centric programming API, and deployment options. If you're interested in building next-generation intelligent applications using a next-generation database, this is the workshop for you!

Stuart Halloway (@stuarthalloway)**Relevance, Inc**

Stuart Halloway, the author of Programming Clojure, is a co-founder of Relevance, Inc. With over 20 years of experience as a programmer, CTO, and CEO, Stuart has built software systems in finance, health care, contract management, online retail, manufacturing, education, and security automation.

Monday, Sept 24th

9:00 - 9:50 AM

In-memory Databases - the Future is Now!

Location: Theater

The rapidly declining costs of DRAM have made in-memory databases a practical option for many applications that need to handle "firehose" data sources and support realtime analytics. In 2008, a team of researchers from MIT and Brown formed a project to examine the feasibility of a relational database prototype based exclusively on an in-memory architecture. Led by Michael Stonebraker and including other noteworthy scientists like Stan Zdonik and Sam Madden, the "H-Store" team trashed many of the prevailing RDBMS design notions. Their research led them on a fascinating journey through transaction management, multi-core and shared nothing scaling, HA, durability, etc. Although the H-Store project tells the in-memory story through the lens of relational databases, many of the concepts and strategies are applicable to high performance data management problems generally. In his keynote talk, Dr. Stonebraker will discuss the architecture of high performance main memory DBMSs, indicating the pros and cons of various choices.

Michael Stonebraker (@voltedb)**VoltDB**

Michael Stonebraker is a renowned researcher and pioneer of database technology. He was the main architect of the INGRES relational DBMS, the object-relational DBMS, POSTGRES, and the federated data system, Mariposa. All three prototypes were developed at the University of California at Berkeley where Stonebraker was a Professor of Computer Science for 25 years.

Dr. Stonebraker has authored scores of research on database technology, operating systems and the architecture of system software. He was awarded the ACM System Software Award for his work on INGRES. He was elected to the National Academy of Engineering in 1998 and is presently an Adjunct Professor of Computer Science at M.I.T. Stonebraker is co-founder and CTO of VoltDB, Inc. (www.voltedb.com).

10:00 - 10:50 AM

Concurrency in Android

Location: Peabody Lounge

The mobile platform adds new challenges to concurrent programming making it much more like what used to be called "real time programming". The Android OS addresses these challenges with three frameworks: a managed application lifecycle, the Looper class, and the AsyncTask template. But is it enough?

Moblie appliances change the nature of the concurrent demands put on applications and the operating systems that support them. Until very recently, concurrency was an optimization trick that allowed an expensive machine to service a queue of tasks effectively. Instead of waiting for something real-world and slow, it could do work on other queued tasks.

With mobile appliances "concurrency" means something more like what used to be called "real time computing". A mobile device user may be, simultaneously, using a web browser, listening to music, taking direction from a navigation service, and receiving text and phone calls. Furthermore multi-core processors are becoming common on mobile platforms: true concurrency will replace synchronous processor sharing.

The Android OS includes tools at three levels of granularity (in addition to those standard in `java.util.concurrent`), to address the need for concurrent execution: the managed application lifecycle, the Looper / Handler types, and the AsyncTask template. While these tools are partially successful each introduces its own new problems. In particular, the interactions between the managed application lifecycle and the AsyncTask can be especially problematic.

G. Blake Meike (@callmeike)

G. Blake Meike is an engineer with more than 30 years of experience, much of it with Java. He has built systems as large as Amazon's massively scalable AutoScaling service and as small as a pre-Android OSS/Linux based Java-like platform for cell-phones. He is currently deep in Android.

He is co-author of two Android books: "Android Application Development" and the very popular "Programming Android", both from O'Reilly.

Blake is currently working on Android internals for a Open Mobile World Wide, an disruptive new startup, and teaching Android classes for Marakana, Inc.

Pontificating Quantification

Location: Theater

In 1969, Tony Hoare published a paper that would change computer science forever. He reflected on the ideas of consequence in program execution and created a set of formal definitions for proving correctness. This is a far cry from how most software is verified today: through automated unit testing in the style of X-Unit. In this talk, Aaron and Daniel will brave the forgotten paths of software verification beyond the realm of unit tests. They will look at some of Hoare's original concepts and how others have built on them, gain a deeper understanding of type theory and the limitations of language-integrated proof systems, and finally examine the restrictions that can be imposed on language expressiveness to make verification more tractable.

Aaron Bedra (@abedra)
Groupon

Aaron Bedra is a Senior Software Engineer at Groupon working on Groupon's Now! real-time deals platform. He is the co-author of Programming Clojure 2nd Edition, Practical Software Security, and of another upcoming Pragmatic Press book.;Daniel Spiewak is a software developer based out of Wisconsin, USA. Over the years, he has worked with Java, Scala, Ruby, C/C++, ML, Clojure and several experimental languages. He currently spends most of his free time researching parser theory and methodologies, particularly areas where the field intersects with functional language design, domain-specific languages and type theory. Daniel has written a number of articles on his weblog, Code Commit, including his popular introductory series, Scala for Java Refugees.

Daniel Spiewak (@djspiewak)
Precog

Daniel Spiewak is a software developer based out of Wisconsin, USA. Over the years, he has worked with Java, Scala, Ruby, C/C++, ML, Clojure and several experimental languages. He currently spends most of his free time researching parser theory and methodologies, particularly areas where the field intersects with functional language design, domain-specific languages and type theory.

Daniel has written a number of articles on his weblog, Code Commit, including his popular introductory series, Scala for Java Refugees.

Understanding Indexing

Location: Midland States

Indexes are used to improve query performance. As a result, good indexes are designed around queries that users find important in their application. This talk presents three simple and effective rules on how to construct indexes around queries that result in good performance.

The rules are explained using a simple model that does NOT rely on understanding B-Trees, Fractal trees, or any other data structure used to store the data on disk. They are derived from these simple properties:

- point queries are slow
- range queries are fast.

Zardosht Kasheff Tokutek

Zardosht has been a senior member of Tokutek's TokuDB engineering team since 2008. Leveraging Fractal Tree Indexing, TokuDB is a high performance storage engine for MySQL that is designed for write-intensive workloads.

Prior to Tokutek, Zardosht worked as a software design engineer at Microsoft on remote terminal services. Zardosht holds B.S. and M.S. degrees in Computer Science from the Massachusetts Institute of Technology. While at MIT, he pursued research on cache-oblivious dynamic search trees under Bradley Kuszmaul at the Computer Science and Artificial Intelligence Laboratory.

Monad examples for normal people, in Python and Clojure

Location: Pepsi Encore

"A monad is a monoid in the category of endofunctors, what's the problem?" That's a quip from James Iry, and he's right - it's unfortunate that most monad tutorials talk about the implementation of monads, instead of showing us how to use library-provided monads to make our business code better. Monads are a really awesome design pattern for abstracting repetitive plumbing from large business applications. This talk will introduce by example common patterns of repetition and plumbing in large codebases, how monads and monad transformers help abstract this plumbing. We will hand-wave over the implementation of most monads because they can be provided by a library anyway, and focus on code examples that use monads, in Python and Clojure side by side.

Dustin Getz (@dustingetz)

enterprise cloud computing. functional programming enthusiast. writes code 7 days a week.

Get a Leg Up with Twitter Bootstrap

Location: Centene Room

If you're like me, you probably can sling some serious server- and client-side code ... but can't choose colors or fonts if your life depended on it. So you code up some basic ugly HTML and hope it can be made pretty later. But what if there was a way to give your application a good (and customizable) look and feel right out of the box ... not just better fonts and layout, but some higher-order user interface elements including modal dialogs, alerts, dropdowns, tooltips, and more? That's Twitter Bootstrap, a base package of CSS and JavaScript that gives your applications that instant "Web 2.0" look and feel.

Howard Lewis Ship (@hlship)
TWD Consulting

Howard Lewis Ship is the creator and lead developer for the Apache Tapestry project, and is a noted expert on Java framework design, web application development, dependency injection, Java meta-programming, and developer productivity. Lately, he's been dipping his toes into alternate languages, including Haskell and Clojure. Howard is an independent consultant, offering Tapestry training, mentoring and project work as well as training in Clojure.

11:00 - 11:50 AM

Famous Unsolved Codes: Kryptos

Location: Midland States

Kryptos is an encrypted sculpture in the middle of CIA Headquarters in Langley, Virginia. Comprised of large copper sheets engraved with thousands of characters, it carries a series of ciphers. Three of the four have been solved, but the fourth section, 97 characters at the very bottom, remains uncracked. It is one of the most famous unsolved codes in the world, even referred to in Dan Brown's "The Da Vinci Code"-sequel "The Lost Symbol". St. Louis cryptographer Elonka Dunin (namesake for the "Nola Kaye" character in Brown's novel), and one of the few people from outside the CIA to have seen the sculpture in person, will review the sculpture's history, going over the parts that have been solved, and discussing efforts to crack the famous unsolved section, K4.

Elonka Dunin (@ElonkaDunin)
Simutronics



Elonka Dunin is Executive Producer and General Manager of Online Community at Simutronics Corporation, Chairperson Emerita and a founding member of the International Game Developers Association's Online Games SIG, and a co-Director of the Global Game Jam. Her company, Simutronics, was founded in 1987 in St. Louis, Missouri, and is a leading developer of online multiplayer games.

Elonka is co-founder and co-leader of a group of cryptographers who are working hard to crack a code on the famous Kryptos sculpture at CIA Headquarters, and led the international team that cracked the related KGB Cyrillic Projector Cipher in 2003. She maintains a list of the World's most famous unsolved codes on her elonka.com site, and in 2006, published The Mammoth Book of Secret Codes and Cryptograms.

Concurrency in iOS

Location: Peabody Lounge

iOS includes Apple's new Grand Central Dispatch framework for managing concurrency. Since Grand Central Dispatch makes heavy use of blocks, Apple's closure-like extension to the C language, this talk will first cover blocks, then go into Grand Central Dispatch. We'll discuss usage and best practices from the ground up for GCD, allowing you to write awesome, performant apps. As an added benefit, since GCD is a system-level framework, everything in this talk applies equally to Mac OS X and iOS!

Jeff Kelley (@SlaunchaMan)
Detroit Labs

Jeff Kelley is a senior-level iOS developer with experience in a wide range of industries. Jeff speaks at conferences and local Cocoa programming groups on topics such as the Objective-C Runtime and Grand Central Dispatch. His projects range from the Chevy Game Time second-screen experience, the Fathead Big Shot augmented reality product viewer, the Podcastic podcast player, to iPad product guides, event managers and virtual tours for car dealerships. Jeff has a reputation of over 6000+ on StackOverflow and is has just completed an introductory iPhone/iPad book for Apress called Learn Cocoa Touch, due out later this year.

Software Architecture using ZeroMQ

Location: Pepsi Encore

ZeroMQ is an open source library that connects your components over different transports and in various patterns. In this talk I'll explain how to use ZeroMQ for large-scale distributed architecture design. The key points are the use of contracts and rapid iterative design cycles. I'm the author of the

upcoming O'Reilly book on ZeroMQ, co-founder of the project and veteran of dozens of large distributed software projects.

Pieter Hintjens (@hintjens)
iMatix Corporation

Pieter Hintjens started a games company over 30 years ago and has been building software products since then. He wrote, "the real physics of software is the physics of people", and today focuses on building communities through "Social Architecture", writing, and helping others use ZeroMQ profitably. He lives with his wife and three young children in Brussels, Belgium, and travels extensively. His Guide to ZeroMQ is coming out as an O'Reilly book (summer 2012). He is writing a new book, "Software and Silicon", about the impact of the Internet on our world.

Functional Design Patterns

Location: Theater

Functional programming languages have patterns too! Some are familiar from the "Gang of Four," others are unique. Using examples in Clojure, this talk will describe design patterns that occur frequently in functional programming languages, and show how they can produce code that is both simpler and more extensible. The word "monad" will not be spoken.

Stuart Sierra (@stuarts Sierra)
Relevance, Inc.

Stuart Sierra is a developer at Relevance, Inc., a member of Clojure/core, and the co-author of Practical Clojure (Apress, 2010) and a forthcoming ClojureScript book from O'Reilly. He has been involved with open-source technologies including Clojure, ClojureScript, Ruby, Rails, Hadoop, and Solr.

Renormalize - The sequel to SQL

Location: Centene Room

Relational databases are pretty useful but:

- Joins are slow, even with lots of indexes and a great optimizer.
- ORMs hide but don't eliminate joins.

Document stores

- Natively handle objects, but,
- Abandon the idea of querying the data using a language.

The Akiban Database Server fundamentally solves the join problem and unifies the two worlds.

- Table groups assemble sets of tables into a new logical construct called table-groups. Table groups are a good match for your intuitive notions about what is related to what, i.e., mirror application objects.
 - New physical organization: Store the joined rows of a table-group interleaved in a b-tree. E.g. a Customer is followed by its Orders. Each Order is followed by its items. Record keys are constructed so that insertion into the b-tree maintains this interleaving.
 - The join is now precomputed, and joins that used to be expensive are essentially free. The rows of the join are consecutive b-tree records.
 - ORMs can now request, for example, a Customer with all of its Orders and Items. A connection setting instructs the server to return the result as a JSON-structured object. Voila, a Customer its Orders and Items in one round trip, with the join computed for the cost of one b-tree probe on the server.
 - Powerful indexing on a table group is now possible.
- What this means
- Table-group interleaving reflects intuitive understanding of objects/documents.
 - Documents can be accessed in a single request. ORMs can leverage this for efficient interaction between ORM and database.
 - Joins within a group are close to free. Complex queries often run 10x-150x faster in memory (compared to MySQL). Denormalization is not necessary.
 - The perfect synthesis of documents and SQL.

Jack Orenstein
Akiban Technologies

Jack Orenstein is a serial founder/architect with deep roots in database system starting with his Ph.D. in Computer Science from McGill University. Jack was a founder of Object Design focused on query processing and SQL interoperability of ObjectStore. A founder of Geophile, focused on spatial query processing. A founder of Binary Mines focused on ORM, and more recently a founder/architect at Archivas, acquired by Hitachi Data Systems. Jack joined Akiban with the founding team in 2009.

Ori Herrnstadt (@oriherrnstadt)
Akiban Technologies

Ori Herrnstadt is co-founder and CTO of Akiban. He has been developing large-scale operational database technology since his days in the Israeli

Army. After experiencing the common problems of operational systems in the military, SaaS, healthcare, enterprise ERP, CRM systems and modern web applications, he started developing the Orthogonal Architecture. In 2009, he co-founded Akiban Technologies to accelerate the development of the technology into the Akiban Server.

11:50 AM - 1:00 PM Lunch (provided)

Location: Kiel Club

12:20 - 12:50 PM

A Whole New World

Location: Theater

Few of us have participated in the creation of our infrastructure—operating systems, compilers, terminals, editors, etc., even though many of us know how to build them in theory. Collectively, we suffer from a learned helplessness around them: to build new high-level tools, we'd also have to rebuild some of the infrastructure, sometimes going all the way down to the kernel. We can't imagine triggering such a large cultural and technological shift, so we don't even try to build truly new tools for ourselves. I've been working on such a stack-busting tool and, though I won't spill the beans on it here, I'll say that it's required me to reconsider and change some very old, universally deployed infrastructure. This talk will briefly demonstrate that project, but only as an example of the underlying trend: our failure to clean up previous decades' messes, the opportunities we miss as a result, and how we might break this cycle.

Gary Bernhardt (@garybernhardt) Destroy All Software

Gary Bernhardt is a creator and destroyer of software, compelled to understand both sides of heated software debates: Vim and Emacs; Python and Ruby; Git and Mercurial. He runs Destroy All Software, which publishes advanced screencasts for serious developers covering Unix, Ruby, OO design, and TDD.

What may not be expected in a country of eternal light?

Location: Pepsi Encore

What may not be expected in a country of eternal light? Building GIS Applications with Scala, Lift, JQuery, & ESRI.

Web applications with geographic information system features can be disjointed, Frankenstein creatures that do neither GIS nor their business purpose well. By treating browser-level features like GIS UI and regular application UI as HTML and Javascript DSLs using Scala and Lift, a much smoother user experience is possible and also enables a stronger integration of GIS into the business logic of your web applications. This achievement may require the sacrifice of a far dearer victim though.

With Mary Shelley's Frankenstein serving as a Greek chorus, my talk calls attention to how programming languages limit how you are allowed to think while demonstrating the possibilities for using DSLs in a commercial application.

Noel Weichbrodt (@noelweichbrodt) VSolvit LLC

Inscrutable hair in an inscrutable world.

Deconstructing P vs NP (or why I hate sudoku)

Location: Midland States

Few modern mathematical problems have received more attention than the question of whether or not the complexity class NP-time is fully defined by the complexity class P-time. This problem, often referenced by its terse proposition "P=NP", holds a fascinating position as a problem which is easy to understand with a fairly obvious and intuitive answer, while simultaneously eluding the most determined attempts at a rigorous proof.

Despite its importance and underlying elegance, the problem of P vs NP remains essentially mysterious in the realm of industrial software engineering. This talk will take a deep dive into the nature of this question. In the process we will gain deeper insight into broad questions of complexity theory, computability theory and combinatorics, as well as acquire a taste for just what it is that makes this problem so difficult to crack.

Daniel Spiewak (@djspiewak) Precog

Daniel Spiewak is a software developer based out of Wisconsin, USA. Over the years, he has worked with Java, Scala, Ruby, C/C++, ML, Clojure and several experimental languages. He currently spends most of his free time researching parser theory and methodologies, particularly areas where the field intersects with functional language design, domain-specific languages and type theory.

Daniel has written a number of articles on his weblog, Code Commit, including his popular introductory series, Scala for Java Refugees.

As We May Do -- Augmented Reality and Computer Vision

Location: Peabody Lounge

Augmented Reality and Computer Vision applications are becoming important and useful. Applications like Google Glasses may change the way we use our computers and mobile devices. Existing visual augmented reality applications such as satellite locators and face detectors as well as non-visual augmented reality applications such as Soundhound are already useful.

Using tools like OpenCV, the vision part of augmented reality applications aren't as hard to develop as you may think. You'll see demos and we'll explore code examples in OpenCV/Python.

Neil Milsted

Neil Milsted is programmer with a wide amount of experience including compilers, databases, data-compression, text editors and voice applications. Neil currently works for a VOIP company in the Chicago suburbs.

1:00 - 1:50 PM

Relational Programming in miniKanren

Location: Pepsi Encore

We will give a whirlwind tour of relational programming in miniKanren, a logic programming language we designed with Oleg Kiselyov, and which is described in 'The Reasoned Schemer' (MIT Press, 2005). miniKanren is simple but powerful, with only three core language forms. miniKanren's implementation is designed to be easily modified and extended, and has been ported from Scheme to more than a dozen host languages, most recently in the form of Clojure's 'core.logic' standard library.

We will present a variety of increasingly complicated miniKanren relations that "run backwards." For example, the addition relation can calculate that $3 + 4 = X$ has the solution $X = 7$, but can also calculate that $3 + X = 7$ has the solution $X = 4$. More interestingly, the addition relation can generate all integer pairs (X, Y) such that $X + Y = 7$. We will demonstrate a relational Scheme interpreter that can *generate* programs that evaluate to a specified value (if any such programs exist!). Finally, we will show how the relational interpreter can be made to generate quines, which are programs that evaluate to themselves.



heroku

Forget Servers. Forget Deployment. Build Apps.

heroku.com

CONTEGIX

technology to empower ideas®

www.contegix.com

- ▶ **MIRACLOUD®**
Public and Private Cloud
- ▶ **COLOCATION**
- ▶ **HYBRID HOSTING**
Create Your Own Hosting
Solution
- ▶ **MANAGED HOSTING**
Dedicated, Cloud and Colocation
- ▶ **MANAGED APPLICATIONS**
- ▶ **GO BEYOND™ SUPPORT**
24/7/365 Tier 3 Engineer Support
100% Network & Power Uptime
Guarantee

STRANGELOOP ATTENDEES:

Receive a FREE, 2-Month Trial of MiraCloud® and see why it is the most flexible cloud solution available. Visit www.MiraCloud.com/strangeloop and enter the code STRANGELOOP2012. Offer expires Oct 25, 2012. (More details on attached MiraCloud® flyer)

Daniel Friedman (@dfried00)
Indiana University

Daniel P. Friedman is Professor of Computer Science at Indiana University. He is co-author of The Little Schemer, The Seasoned Schemer, The Reasoned Schemer, Scheme and the Art of Programming, and Essentials of Programming Languages, 3rd Edition, all published by MIT Press.

William Byrd (@webyrd)
Indiana University

William E. Byrd is a Postdoctoral Fellow at the Center for Research in Extreme Scale Technologies (CREST) at Indiana University. He is co-author of The Reasoned Schemer, and co-designer of several declarative languages: miniKanren (logic programming), Harlan (GPU programming), and Kanor (cluster programming). His StarCraft 2 handle is 'Rojex' (character code 715).

Postgres Demystified

Location: Midland States

Answering questions of your data and keeping your database interactions performant does not have to be rocket science. This talk will highlight many of the tools that exist within Postgres that can be accessible to both experienced SQL users and novice SQL users. Each of these tools will be highlighted by a real life use case of when its valuable and a clear example of how to use it in terms that non-DBAs understand. Demystified topics will include

- hstore (NoSQL inside your SQL)
- Window Functions (Grouped data calculations)
- Execution plan (Understanding query performance)
- Foreign Data Wrappers (Oracle/Redis/Mysql from Postgres)
- Concurrent indexes (Adding indexes without production impact)

Craig Kerstiens (@craigkerstiens)
Heroku

I'm a product guy at Heroku. I write code in Python, work with startups, and solve problems.

1:00 - 1:20 PM

Designing voice navigated apps

Location: Peabody Lounge

Using your voice to navigate your device, is something people have been working on for decades. Since Apple introduced Siri with the iPhone 4S in 2011, we now know that voice

recognition and having a conversation with your phone works. It feels better and more natural for a human being to work that way.

But how and when should you use voice recognition when developing apps? What do you need to take into account? What should you avoid to annoy your users? How can it be that the use of voice is only now turning into something common? And how do you take your first steps into coding up your own voice navigated app?

In this talk I'll give you a demo of our travel-planner mobile app (iOS and Android) and talk about the UX/Voice design considerations we made, lessons we learned from users feedback and the frameworks (Google, Nuance) we used.

Robert van Loghem (@soundbites)
Xebia

Robert is always interested in the latest and greatest when it comes to; User experience, mobility, communication, infrastructure and coming up with some crazy creative solution by combining tech.

Since 2010, He gets a kick out of building apps for mobile devices, especially ones that his girlfriend loves, all this doing TDD of-course! So evangelizing what he practices he's heading up Xebia's mobile division in 2011, and loving it! It's great to work with all these technologies, Android, iOS and HTML5 and pouring a nice Continuous Integration sauce all over it, to deliver secure, robust, integrated and performant apps, that look great!"

A Type Driven Approach to Functional Design

Location: Theater

Design is the thing that we all assume that we can do. Once we know the syntax of a language we write code to solve our problem, often in an exploratory way. We are left wondering whether there is a more deterministic route toward our solutions.

In this talk, Michael Feathers will describe an approach toward planning the design of the functional portions of an application by using a variation of Haskell type expression syntax. The advantage of using this syntax is that it lays bare opportunities for refactoring and alternative program structure. It can also guide implementation in a variety of static and dynamic functional languages without being overly prescriptive.

Michael Feathers (@mfeathers)
Groupon

Michael Feathers is a Member of the Technical Staff at Groupon. Prior to joining Groupon, Michael was the Chief Scientist of Obtiva, and a Senior

Consultant with Object Mentor International. Over the years, Michael has spent a great deal of time helping teams after design over time in code bases. Michael is also the author of the book Working Effectively with Legacy Code (Prentice Hall, 2004).

This is Your Workflow on Catnip

Location: Centene Room

Once you get used to working with a REPL, it hurts to go back to the old write-compile-restart cycle of less fortunate languages. But the REPL can only take you so far; even if you add tools like Firebug to inspect and manipulate your application's state, for all practical purposes the REPL is really just a glorified debugger.

But suppose you could go further? Suppose that instead of just that command prompt, you could have your whole IDE embedded in your runtime environment? Catnip is a project aiming to do just that, giving you a fully featured code editing environment as well as a REPL, all running inside your actual Clojure project, integrating with it naturally and transparently.

This presentation explores the productivity gains of having your IDE tightly coupled with a live, running web application, where everything you do is reflected in the application in real time, and where the application itself provides the feedback you need to supercharge your workflow in a way that your enterprise ancestors could never have imagined.

Bodil Stokke (@bodiltv)

Bodil has been a web developer since before the web was cool, and has shepherded many big projects built on web technologies, among them two generations of web map applications for Eniro, Scandinavia's leading business search provider. She currently works as a freelance consultant, helping clients drag their projects kicking and screaming out of the enterprise world into the shiny new future of technologies like Git, Clojure and Node.js. She lives in Oslo, Norway with her pet Roomba and two chili plants.

1:30 - 1:50 PM

Doppio: Building a JVM in the Browser

Location: Centene Room

Modern browsers provide sandboxed versions of many native system interfaces, such as graphics rendering and a filesystem. Thus in theory we should be able to replicate most of the desktop experience on the web -- except for the fact that a

great deal of applications are not written in Javascript. Enabling the browser to run other languages would add whole classes of applications to the web platform.

Doppio is an effort to bring the JVM languages to the web by implementing a JVM in Coffeescript. While the JVM specification is technically language-agnostic, the original JVM is written in C and C++, and its architecture reflects that. We'll discuss some of the challenges of implementing the spec and porting the libraries to a high-level, non-systems language, particularly when it does not expose threads directly. We'll also talk about how NodeJS was invaluable for development.

Jez Ng (@jezng)

Jez is a Javascript geek and Vim enthusiast at Amherst College. He's also proud to be one of the developers of the Vimium browser extension, as well as a former Mozilla intern.

CJ Carey (@perimosocordiae)

CJ is an alum of Wash U, currently pursuing a PhD in CS at UMass Amherst. In addition to his research in Machine Learning, he dabbles in programming language design and theory.

Jonny Leahey (@HeyJonLeah)

Jonny graduated from UMass Amherst with a BS, but he stuck around and is currently pursuing an MS in CS. He researches reinforcement learning, but he also has an interest in systems.

Zipkin: a distributed tracing framework

Location: Peabody Lounge

Zipkin allows Twitter to trace a subset of all requests to the site. For each traced request we collect detailed data about the path taken through our systems as well as timings. This data allows us to visualize a request and ultimately aids us in pinpointing where and possibly why a response took longer than expected.

Tracing is done by instrumenting libraries such as Finagle used for internal communication between our services. The generated trace data is then logged to collectors that store and index the information in Cassandra. On top of Cassandra we also run a query daemon and a web based UI that help us make sense of the data.

Our goal is to have Zipkin open sourced in time for this talk. Using the released code we'll demonstrate how easy it is to enable tracing for your system and the benefits it brings.

Johan Oskarsson (@skr) Twitter

Johan Oskarsson is a software developer at Twitter in San Francisco. He made his way there via university in Stockholm and as a developer at Last.fm in London.

At Twitter Johan spends most of his time coding in Scala for the runtime team. He's previously been involved in various open source projects including a short stint as a committer for Apache Hadoop and Apache Cassandra.

Johan founded and ran the Hadoop User Group in London, which he is proud of, as well as the first ever NOSQL meetup, which he does not quite know how to feel about.

The Higher Order Rubyist

Location: Theater

Higher Order Perl changed the life of many the Perl hacker and indeed may have changed the Perl community on a whole. For better or worse, Ruby has much stronger OO facilities than Perl which have led to a slower adoption of Functional techniques within the community. As we have learned from texts such as "Concepts, Models and Techniques of Computer Programming", languages that support multiparadigm programming are more easily able to express various domains. This talk will be a survey of tools and techniques for using Ruby in an Object-Functional style, along with the pros and cons of such an approach.

Robert Pitts (@rbxbx) Freelance, at the moment.

Robert is very interested in the intersections of art, computer science, and community. He believes there is no "one true way" and thinks that we all have a lot to learn from each other*, so perhaps we should be nicer :) *insert requisite "except for COBOL" joke here

2:00 - 2:50 PM

A Taxonomy of Scala

Location: Pepsi Encore

A 50 minute tour of all of the more esoteric concepts people new to Scala will hear in the community, covering topics such as actors, functional programming, Type Theory and Category Theory. This is a real-world explanation of what these concepts mean, with examples of how they're used.

Jamie Allen (@jamie_allen) Typesafe

Jamie Allen has over 18 years of experience delivering enterprise solutions across myriad industries, platforms, environments and languages. He has been developing enterprise applications with Scala since 2009, primarily using Actors for fault tolerance and managing concurrency at scale.

ClojureScript: Better Semantics at Low, Low Prices!

Location: Theater

Transpiling a language with richer semantics than the target language always presents some fascinating challenges with regards to performance. In this talk we'll explore the variety of strategies employed to get ClojureScript to perform well on modern JavaScript engines. We'll look at how ClojureScript optimizes code size, expression-oriented semantics, numerics, operations on JS primitive types, persistent data structures, and polymorphic function dispatch.

David Nolen (@swannodette) The New York Times

David Nolen is a JavaScript developer for The New York Times. He also runs a Brooklyn based workshop called Kitchen Table Coders. In his free time he does fun stuff with Clojure and ClojureScript.

Erlang Web Development with Yaws

Location: Centene Room

Yaws -- "Yet Another Web Server" -- is an open source Erlang web server started by Claes "Klacke" Wikström in 2001. It is a full-featured scalable web server that can run standalone or be embedded into other applications. Yaws supports a very long list of features ranging from old-school CGI to modern-day WebSockets, all driven by the needs of real-world applications. Even though it's an 11 year old project, Yaws is still under active development, continuing to gain features and enhancements as requested by its many users in the Erlang community.

In this talk, Steve, a committer on the Yaws project since 2008, will provide an overview of Yaws and its many features, and will show code examples to highlight some of those features. Steve will also discuss some of the internals of Yaws as well as explaining his general thoughts on the incredible

Hiring skilled web, database, cloud, and cluster developers

Leveraging the latest programming languages, the web, and cloud technologies to build software for the genomic revolution

Partek.getInstance().add(you);

WHERE SCIENCE AND CREATIVITY MEET



www.partek.com

Chesterfield Valley location with a Silicon Valley vibe

Casual, come as you are, environment with flexible hours and the brightest minds

Comprehensive benefit package including 100% paid health and dental insurance

Equal opportunity employer

stripe

Accept payments online.

power that Erlang brings to web development and distributed systems.

**Steve Vinoski (@stevevinoski)
Basho Technologies**

Steve Vinoski is an architect at Basho Technologies in Cambridge, MA, USA. He's worked on distributed systems and middleware systems for over 20 years, including distributed object systems, service-oriented systems, and RESTful web services. His interest in software quality and development speed led Steve to start exploring and using Erlang in 2006, and he's used it as his primary development language ever since. A magazine columnist since 1995, he currently writes "The Functional Web" column for IEEE Internet Computing where he explores the use of functional programming languages for web development.

Building an Impenetrable ZooKeeper

Location: Peabody Lounge

ZooKeeper provides reliable and timely coordination of processes. Given the many cluster resources leveraged by distributed ZooKeeper, it's frequently the first to notice issues affecting cluster health, which explains its moniker, "the canary in the Hadoop coal mine." Attend this session and learn (1) how to configure ZooKeeper reliably, (2) how to monitor ZooKeeper closely, and (3) how to resolve ZooKeeper errors efficiently. Culling from the diverse environments we've supported, we'll share what it takes to set up an impenetrable ZooKeeper environment, what specifically to monitor, and which ZooKeeper errors indicate something amiss with your hardware, network, or HBase configuration.

**Kathleen Ting (@kate_ting)
Cloudera**

As a Customer Operations Engineer at Cloudera, Kathleen Ting stays up to keep your Hadoop production cluster up. Within the Hadoop platform, her focus is on Sqoop, Flume, ZooKeeper, and MapReduce. She is a committer on the Apache Sqoop project.

Executing Queries on a Sharded Database

Location: Midland States

Determining a data storage solution as your web application scales can be the most difficult part of web development, and takes time away from developing application features. MongoDB, Redis, Postgres, Riak, Cassandra, Voldemort, NoSQL, MySQL, NewSQL -- the options are overwhelming,

and all claim to be elastic, fault-tolerant, durable, and give great performance for both reads and writes. In the first portion of this talk I'll discuss these different storage solutions and explain what is really important when choosing a datastore -- your application data schema and feature requirements.

No matter what datastore you choose, you will eventually have to consider sharding your data store to support growing traffic. Two key challenges arise: (1) web workloads often do not have one clear partitioning and (2) it is challenging to determine how to efficiently execute queries over partitioned tables.

To address these challenges I present Dixie, a SQL query planner, optimizer, and executor for databases horizontally partitioned over multiple servers. Dixie shows that we shouldn't give up on SQL databases just yet. Dixie automates the exploitation of tables with multiple copies partitioned in different ways, in order to increase throughput by expanding the set of queries that need not be sent to all servers. Central to Dixie's design are a cost model and plan generator that are mindful of queries small enough that query overhead may dominate the cost. For a large class of joins, which traditional wisdom suggests require tables partitioned on the join keys, Dixie can find higher-performance plans using other partitionings.

**Neha Narula (@neha)
MIT**

Neha Narula is a PhD candidate at MIT building fast, scalable distributed systems. In a previous life she was a Senior Software Engineer at Google, where she designed the first version of Blobstore, a system for storing and serving petabytes of immutable data, and worked on Native Client, a system for running native code securely through the browser. She cares a lot about coffee, bourbon, how we get our news, and securing user data.

2:50 - 3:30 PM

Break

Location: Foyer

3:30 - 4:20 PM

Information Rich Programming with F# 3.0

Location: Peabody Lounge

Modern programming thrives on rich spaces of data, information and services. F# 3.0 brings integrated support for Information Rich

Programming to the .NET platform. F# Type Providers and F# Queries greatly simplify data-rich analytical programming, allowing programmers to easily access and manipulate a variety of data sources. In this talk, I'll introduce these exciting new features and how they can be used to leverage technologies such as OData, SQL Server, WSDL services, SharePoint, and Windows Azure Marketplace.

**Donna Malayeri (@lindydonna)
Microsoft**

Donna Malayeri is the program manager of "Visual Studio F#":<http://fsharp.net>. Before joining Microsoft, she worked on the "Scala":<http://www.scala-lang.org/> team in Lausanne, Switzerland, focusing on the Scala Eclipse plugin and improvements to scaladoc. She holds a PhD in computer science from Carnegie Mellon University, on the topic of "type systems for improving code reusability":<http://www.cs.cmu.edu/~donna/pubs.html>. She also sometimes writes provocative and controversial--yet award-winning--papers for CMU's prestigious "SIGBOVIK":<http://sigbovik.org/> conference series, the premier forum for all work in its various areas of focus.

Types vs Tests : An Epic Battle?

Location: Theater

What happens when you put a testing fiend and a static typing bigot on the stage and ask them to discuss the best way to build software? Amanda and Paul both have a passion for providing business value with every semicolon (or lack thereof) but they have a very different way of going about it. Can anyone really believe that being held hostage to a compiler is a good idea? How can we ever know we have enough tests and who wants to maintain them? Can these two methods live in harmony? Should they? Can we even get through the session without being blood?

**Amanda Laucher (@pandamonial)
Measured Progress**

Amanda Laucher has been developing software almost her entire life. Solving complex business problems with code is her passion. She has spent many years consulting and decided that making some old rich guy more money wasn't satisfying. Amanda has taken on the Enterprise Architect role at Measured Progress, a not-for-profit, trying to overcome challenges in k-12 personalized learning.

**Paul Snively (@psnively)
VMWare**

Paul Snively was a Lisp programmer for two decades before finally growing up and using static types. He was the technical editor for "Beginning Scala" and thankfully uses Scala on the job every

day. Paul works on the vCloud Services team at VMware.

Data Structures: The Code That Isn't There

Location: Midland States

Most programmers rely on a few core data structures, but they're missing out on useful properties that more specialized data structures provide.

The wrong data structures can bog implementation down in irrelevant detail or create behaviors which waste time and effort, but the right ones can give powerful guarantees for free. My talk will present lesser-known data structures and their unique advantages:

- Binary decision diagrams act on whole sets of items at a time, rather than as collections of individual elements. They also take up less space as more items are added, automatically detecting and eliminating redundancy. Example use case: Datalog queries.
- Skip lists are simple data structures whose design leads to balanced binary tree-like performance, without any need for non-localized operations such as rebalancing. Example use case: Demonstrating how simple invariants can lead to powerful emergent properties.
- Rolling hashes can find deterministic breaking points in buffers of binary data, enabling consistent chunking and re-use as data changes. Example use case: rsync.
- Jumprobes (a data structure of my own invention) automatically de-duplicate content stored in them, including data shared between multiple files. Modified content can be stored with very little additional overhead, allowing for cheap versioning. Finally, the next several fragments can always be retrieved in parallel, enabling simple buffering for streaming media. Example use case: scatterbrain, a distributed filesystem (to be released soon).

Scott Vokes (@silentbicycle)

Atomic Object

I've been programming since I was a kid. I'm interested in everything from embedded systems to distributed search platforms. And bicycling. I live in Grand Rapids, MI.

Lessons from Erlang: Principles of Reliable Systems

Location: Centene Room

Erlang is famed for building systems of incredible uptime and reliability! What makes this language so adept at solving hard systems engineering problems? Is the answer cloaked in arcane mystery known only to elite Erlang magi? As it turns out, Erlang's design reflects obvious tenets, born of Ericsson's decades of experience engineering telecom infrastructure. In this presentation, we will look at the principles that inspired Erlang's design -- and importantly, how you can apply them in any language!

We'll start with a short history of Erlang -- in particular, the type of problems that Ericsson faced in the 90s that drove the creation of a new language. As it turns out, these problems are familiar to anyone who's built web applications: massive concurrency, extreme uptime, fault tolerance, performance, and of course, time to market.

Next, we'll dive into the principles themselves: - Isolation

- Fault detection and recovery
- Separation of concerns
- Back box design
- State management
- Dependency management

Finally, we'll discuss practical ways to apply these principles in other languages including Ruby, Python, and Java. Techniques include:

- Use of "fork-exec" for process isolation - Process supervision using tools like runit and launchd - Narrowing the scope of an "application"
- Avoiding shared memory using message passing - Simplifying state
- Minimizing dependencies

If you're responsible for building systems, or just curious to learn more about the topic, this talk will distill the most important features of Erlang and give you a jump start in leveraging them in your own language!

Garrett Smith (@gar1t)
CloudBees

Garrett is an architect at CloudBees, a leading Java platform-as-a-service vendor. Garrett works on the services that drive CloudBees' Java runtime environments, host to tens of thousands of dynamic, scalable web applications and databases.

In his spare time, Garrett teaches Erlang, hacks code, sips whiskey, and creates videos like "MongoDb Is Web Scale" and "Node.js Is Bad Ass Rock Star Tech".

Graph: composable production systems in Clojure

Location: Pepsi Encore

Prismatic's newsfeed generation service has dozens of components including storage systems, in-memory caches, pub/sub messaging, indices, HTTP handlers, and dynamic relevance ranking algorithms, all of which work together in an intricate dance to serve personalized feeds.

Designing an architecture of this complexity is itself a nontrivial task; and to run it in production, we also need to be able to monitor each component for performance and failures, reuse sub-components in other services, and test our services by mocking out components.

To help tame this complexity, we've developed Graph: a simple, declarative approach for specifying system compositions. A Graph specification is just a Clojure map from node names to functions with named arguments, each of which can refer to the output of a previous step or an input to the graph. This approach is very general; in addition to compiling Graphs down to production services (where nodes build components), we also use Graphs for real-time newsfeed generation and streaming web document analysis. And because Graphs are just Clojure maps, it is trivial to reason about them, mock out components (assoc), monitor individual nodes (map observer functions over leaves), and compose them with a variety of execution strategies to intelligently schedule real-time computation.

Jason Wolfe (@w01fe)
Prismatic

Jason Wolfe builds large systems, machine learning, and ranking algorithms for news personalization at Prismatic. Jason recently completed his PhD in Computer Science at UC Berkeley, specializing in Artificial Intelligence.

4:30 - 5:20 PM

The Database as a Value

Location: Theater

Proponents of functional programming tout its many benefits, most of which are available only within a particular process, or afforded by a particular programming language feature. Anything outside of that is considered I/O, dangerous and difficult to reason about. But real systems almost always cross process and language boundaries, and most require, crucially, a very gnarly bit of shared



At our core, we're a technology company applying our talents to the domain of investment management.

We've created a system that combines artificial intelligence and keen human insight — a system that's constantly learning, improving, and advancing.

We like to solve hard problems.

What kinds of problems do you want to solve?

We're a destination for the intellectually curious who want to learn, invent, create, and watch what they build make a difference.

Start a conversation. Please contact
Recruiting@twosigma.com



state - a database. In this talk we will examine how Datomic renders the database into that most prized and easy-to-reason-about construct, a value, and makes it available to multiple processes in multiple languages, functional and not. Along the way, we'll discuss the importance of immutability and time in representing information, the reification of process, and the mechanisms of durable persistent data structures. No knowledge of functional programming is required.

Rich Hickey (@richhickey)

Datomic

Rich Hickey, the author of Clojure and designer of Datomic, is a software developer with over 20 years of experience in various domains. Rich has worked on scheduling systems, broadcast automation, audio analysis and fingerprinting, database design, yield management, exit poll systems, and machine listening, in a variety of languages.

Real-World Redis

Location: Pepsi Encore

Redis is a data structure server, but yet all too often, it is used to do simple data caching. This is not because its internal data structures are not powerful, but I believe, because they require libraries which wrap the functionality into something meaningful for modeling a particular problem or domain. In this talk, we will cover 3 important use cases for Redis data structures that are drawn from real-world experience and production applications handling millions of users and GB of data:

- Leaderboards - also known as scoreboards or high score tables - used in video games or in gaming competition sites
- Relationships (e.g. friendships) - used in "social" sites
- Activity feeds - also known as timelines in "social" sites

The talk will cover these use cases in detail and the development of libraries around each separate use case. Particular attention for each service will be devoted to service failover, scaling and performance issues.

David Czarnecki (@CzarneckiD)

Agora Games

David Czarnecki is a lead engineer at Agora Games (a part of Major League Gaming), <http://www.agoragames.com>, a company that specializes in building middleware to power video games, both in-game and on the web. He is the author of 2 books for O'Reilly and had previously been developing exclusively in Java for nearly 15 years. He has been developing in Ruby and Rails since joining Agora Games in 2008 and has been a speaker at conferences like JavaOne, RailsConf and Ruby

Midwest. David loves video games, especially the Guitar Hero series and Left 4 Dead.

River Trail - Parallel Programming in JavaScript

Location: Midland States

Parallel hardware is ubiquitous yet web developers are missing out as JavaScript so far has been stuck in the sequential past. With River Trail we are bringing parallel programming to JavaScript and enable web developers to exploit the additional performance promised by parallel hardware. In this session, we will discuss the need for parallel programming on the web, explore the design space and, most importantly, give an introduction to parallel programming with River Trail.

Stephan Herhut (@herhut)

Intel

Stephan Herhut is a Research Scientist at Intel Labs where he works on improving programmability of parallel architectures. He is enthusiastic about increasing programmers' productivity without sacrificing performance goals. Lately, he is working to bring the performance of parallel hardware to JavaScript without destroying its soul.

Engineering Elegance: The Secrets of Square's Stack

Location: Centene Room

In 2010, NY Times columnist David Pogue described Square as "beautiful, simple and a joy to use." Square's dedication to user experience continues to strike a chord. Today, more than 2M merchants and individuals use Square to process \$6B annually.

How does Square deliver its trademark seamlessness? Simple. By routinely tackling daunting engineering challenges, shielding merchants and payers alike from the inherent complexities of payment processing.

Bob takes the audience on a tour of Square's stack, tracing the payment lifecycle from end to end. From magnetic stripe decoding to multi-homed data centers, Bob talks about lessons learned—what worked and what didn't—and the variety of technologies and open source frameworks concealed behind Square's simple façade.

Bob Lee (@crazybob)

Square

Square's CTO Bob Lee is a hands on technical leader, engineering everything from smartphone apps, to DSP algorithms, to embedded firmware, to

payments processing infrastructure. Prior to Square, Bob led the development of Android's core libraries at Google. He created the Jolt award-winning Guice framework. He co-authored the bestselling book "Bitter EJB." He leads the specification of JSR-330 "Dependency Injection for Java," and he's a member of the JSR-335 "Lambda expressions" and JSR-166 "Concurrency Utilities" expert groups.

F# for Trading

Location: Peabody Lounge

Effective Trading is about having up-to-date detailed information. Data is acquired from a variety of sources and combined to imply new information. F# is becoming a popular choice for this kind of application. Based on hands on experience of building a Commodity Trading platform I will describe and demonstrate areas where F# gives you leverage like domain modelling, computation and concurrency.

Phillip Trelford (@ptrelford)

Trayport Limited

Phil Trelford is a Software Architect at an ISV supplying real-time electronic trading software. His career so far spans over 20 years, with experience in video games, leisure, retail and financial sectors. Phil's recent commercial development work has been with C++, C#, SQL, JavaScript, and includes over 5 years developing F# applications.

5:30 - 6:20 PM

Pushing the Limits of Web Browsers

Location: Theater

Innovation is crucial for keeping web browsers a vibrant development platform. In the past four years, amazing performance improvements of JavaScript have enabled new kinds of web applications. However, this is far from sufficient. We clearly need to address the inefficiencies of developing large complex web applications. This talk will discuss advances in virtual machine technology and programming languages that likely will shape the future of web programming.

Lars Bak

Google Inc.

Lars Bak is a veteran virtual machinist. His passion for designing and implementing object-oriented virtual machines has left marks on several software systems: Beta, Self, Strongtalk, Sun's HotSpot and CLDC HI, OOVM Smalltalk, V8, and Dart. Since joining Google in the fall of 2006, Lars has been responsible for the design and implementation of V8 & Dart. He graduated from Aarhus University in 1988 with a MS degree in computer science.

Tuesday, Sept. 25th

9:00 - 9:50 AM

Computing Like the Brain

Location: Theater

The neocortex works on principles that are fundamentally different than traditional computers. In this talk I will describe recent advances in understanding these principles and their impact on future computer architectures. The talk will start with a description of sparse distributed representations, which are the fundamental units of information in brains. I will then discuss how these representations are learned and how the brain processes them to build predictive models from high velocity sensory data streams. Numenta has built a streaming analytics engine based on these principles. I will give a brief introduction to this product and speculate on the future of machine intelligence.

Jeff Hawkins (@Numenta)
Numenta

Jeff Hawkins is a co-founder of Numenta. Numenta is creating a new computing platform designed on the principles of the neocortex. Previous to Numenta he founded the Redwood Neuroscience Institute and two mobile computing companies, Palm and Handspring, where he was the architect of computing products such as the PalmPilot and Treo smartphone. In 2004 he wrote the book "On Intelligence" which describes progress in understanding the neocortex.

Jeff Hawkins earned his B.S. in electrical engineering from Cornell University in 1979. He was elected to the National Academy of Engineering in 2003.

10:00 - 10:50 AM

H2O: open source BigDatabase for interactive analytics

Location: Midland States

H2O is the open source BigDatabase for fast interactive analytics.

We auto-parallelize analytic queries and ye plain olde' Java; we run MapReduce and MR languages like Pig & Hive and a subset of R & NumPy. We're using Distributed Fork/Join as our basic execution model; it supports a wide variety of parallel programming idioms. D/F/J gives us dynamic fine grained load-balancing for M/R jobs (cures: Curse of the Last Reducer), and also for irregular jobs like graphs and streams. We read and write from HDFS and S3 data, and index it into our Key / Value store. The K/V store is based on a cloud of Distributed Non-Blocking HashMaps, with a strong consistency model (the JMM for the cloud) using vector clocks, even with multi-datacenter replication. The K/V store gives us fine grained knowledge and control of data placement (e.g. supporting hot-block redistribution and atomic counter updates). We provide schema-on-read supporting BigTable, BSON as well as POJO formats via a little compiler magic (schemas as a DSL). We are bringing Dremel to the Enterprise. We are bringing Moore's law back.

Cliff Click

Cliff Click is the CTO and Co-Founder of 0xdata, a firm dedicated to creating a new way to think about web-scale data storage and real-time analytics. I wrote my first compiler when I was 15 (Pascal to TRS Z-80!), although my most famous compiler is the HotSpot Server Compiler (the Sea of Nodes IR). I helped Azul Systems build an 864 core pure-Java mainframe that keeps GC pauses on 500Gb heaps to under 10ms, and worked on all aspects of that JVM. Before that I worked on HotSpot at Sun Microsystems, and am at least partially responsible for bringing Java into the mainstream.

I am invited to speak regularly at industry and academic conferences and has published many papers about HotSpot technology. I hold a PhD in Computer Science from Rice University and about 15 patents.

Making the Web Faster at Google and Beyond

Location: Peabody Lounge

Google's "Make the Web Faster" team spends all of its time helping optimize internal Google products, as well as the infrastructure of the web as a whole: optimizing TCP, working on developing new protocols such as SPDY, developing open source tools such as mod_pagespeed and HTTP Archive, measuring and modeling mobile performance, and of course, developing a myriad of tools for Javascript and browser debugging and optimization.

In this session, we'll take a quick run through of the lessons we learned at Google and how you can

apply what we've learned and developed in your own applications.

We'll start from the bottom up layers: what is latency, why do we care, how do we optimize it? Then, we'll look at TCP and protocol optimizations developed and deployed at Google, and then move up the stack to talk about mobile and browser performance - how do we model mobile network performance, how do we measure user response time, and ultimately, how do we deliver the best possible user experience to the visitor?

Ilya Grigorik (@igrigorik)
Google Inc.

Ilya Grigorik is a web performance engineer, an open-source and Ruby evangelist, a data geek, and a proverbial early adopter of all things digital. He is currently working with the "Make the Web Faster" team at Google - working on exactly that goal.

Y Not? - Adventures in Functional Programming

Location: Theater

One of the deepest mysteries in the functional programming world is the Y-Combinator. Many have heard of it, but few have mastered its mysteries. Although fairly useless in real world software, understanding how the Y-Combinator works and why it is important gives the student an important insight into the nature of functional programming.

Join with us on this journey of understanding. Be prepared to curry your functions and bind your lambdas as we delve into the whys and wherefores of this paragon of functional programming. Although you will probably never have a need for the combinator, the effort put forth to understand it will improve your functional programming chops. This talk is not for the faint of heart, but the successful student will be richly rewarded.

Also, you will understand why "Y-Combinator" is the perfect name for Paul Graham's start-up funding company.

Jim Weirich (@jimweirich)
EdgeCase

Jim Weirich first learned about computers when his college adviser suggested he take a computer science course: "It will be useful, and you might enjoy it." With those prophetic words, Jim has been developing now for over 25 years, working with everything from crunching rocket launch data on supercomputers to wiring up servos and LEDs on micro-controllers. Currently he loves working in Ruby and Rails as the Chief Scientist at EdgeCase,

Smart **People.** Smart **Technology.** Smart **Careers.**®



National Information Solutions Cooperative (NISC) is an information technology company that develops and supports software and hardware solutions for Member/Customers who are primarily utility cooperatives and telecommunications companies across the nation.

NISC provides a competitive wage along with opportunities for professional growth. Our comprehensive benefits package includes medical, dental, vision, disability, 401k and paid time off.

Visit www.NISC.coop/Careers to check out
our **full-time** and **internship** opportunities.



866.999.6472

www.NISC.coop/Careers



perceptivesoftware
a Lexmark company

but you can also find him strumming on his ukulele as time permits.

Making Hadoop Real Time with Scala and GridGain

Location: Centene Room

The topic of this presentation is about demonstrating how GridGain software allows to build a real time application that loads data from Hadoop HDFS and process it in real time - and how GridGain's approach is different to the one employed by Hadoop framework that is based on offline batch-oriented processing.

Most of the presentation will be concentrate on the live "from scratch" coding of the application that takes an ubiquitous Hadoop example of calculating word frequencies in the large body of text and implementing it using GridGain to work in 100% real time context instead.

The live coding part of the presentation will be done using Scala programming language that is natively supported by GridGain.

The initial 10 minutes of the presentation will be devoted to explaining the big data concepts and high level explanation about GridGain and Hadoop approaches to solving it.

Overall - the presentation is extremely hands-on and require significant level of understanding of Java or Scala and basic level of understanding for distributed programming concepts such as In-Memory Data Grid and Compute Grids.

**Nikita Ivanov (@gridgain)
GridGain Systems**

As a founder of GridGain Systems Mr. Ivanov provides technological direction and vision for GridGain. Mr. Ivanov brings almost 20 years of experience in software development, a vision and pragmatic view of where development technology is going, and high quality standards in software engineering.

In 1996, Mr. Ivanov was one of the pioneers in using Java technology for server side middleware development while working at T-Systems GmbH, one of the largest European System Integrator.

Nikita has held various positions architecting and leading software product development for start-up companies and working with well-established companies such as Adaptec, Visa and BEA Systems. Mr. Ivanov is an active member of Java middleware community and is a contributor to Java specifications.

Behind the mirror

Location: Pepsi Encore

At Microsoft I had the unique opportunity to sit behind a one way mirror and watch people try to solve problems. I saw firsthand how they use their tools, how they go about understanding new code and finding bugs, but what struck me was how little we've done to help people get their work done. It made me realize that we don't fully understand our own craft and that there's so much more we can discover. In this talk we'll discuss the things I learned behind that mirror and from the front of a classroom, so that we can imagine what the future of tools could be.

**Chris Granger (@ibdknox)
Kodowa**

I grew up as part of the nintendo generation, having learned the parts of a computer at the age of two and later learning numbers and colors from a Sesame Street game on the NES. I started programming at the age of ten and took my first paid development gig at 17. Since then I've built websites large and small, written frameworks and libraries used by thousands, taught developers around the world, and helped envision the future of development at Microsoft. These days, I'm the co-founder of Kodowa, where we're building Light Table, a new vision for how we create and learn.

11:00 - 11:50 AM

Runaway complexity in Big Data... and a plan to stop it

Location: Theater

Big Data has dramatically increased the complexity of building data systems. Big Data forces you to leave the comfortable world of ACID, transactions, and relations, and thrusts you into a challenging world of distributed systems, CAP, and restrictive data models.

You cannot battle complexity with ever more complex systems. This leads to restrictive systems that are difficult to operate and have poor performance. The only way to reasonably address the complexity of Big Data systems is to fundamentally rethink your approach to avoid that complexity in the first place. A key insight is that the ability to store and process very large amounts of data opens up entirely new ways of building systems that were not possible pre-"Big Data".

NoSQL is not a panacea. Nor is Hadoop, Storm, or any of the other tools out there for Big Data. Yet there is a way to use these tools in conjunction with

one another to build complete and robust realtime data systems with a minimum of complexity. These techniques are possible today and can be implemented and operated by small teams.

In this talk you'll learn:

- How a huge amount of complexity stems from the CRUD paradigm, and why you only need (and want) CR
- Why embracing immutability is the key to simplifying data systems
- Where NoSQL fits into the big picture
- The "Lambda Architecture": a generic approach to building data systems using a combination of batch processing and realtime processing

**Nathan Marz (@nathanmarz)
Twitter**

Nathan Marz is an engineer at Twitter. Previously Nathan was the lead engineer of BackType which was acquired by Twitter in July of 2011. Nathan has been involved in the Big Data space for more than four years. He is the author of Cascalog, a high level abstraction for MapReduce, and Storm, a distributed and fault-tolerant realtime computation system. These projects are relied upon by dozens and dozens of companies. He is the author of an upcoming book for Manning Publications called "Big Data: Principles and best practices of scalable realtime data systems" and writes a blog at <http://nathanmarz.com>.

The Racket Way

Location: Pepsi Encore

The Racket way is that (1) everything is a program, (2) concepts are programming-language constructs, (3) the programming language is extensible, and (4) everything composes. This talk is primarily a demonstration of what the Racket way means, and it includes a tour of a language for writing prose documents, a language for creating slide presentations, constructs for language extensibility and language creation, OS-like programming-language constructs, and examples that put all of those pieces together.

**Matthew Flatt
PLT and University of Utah**

Matthew Flatt is an associate professor in the School of Computing at the University of Utah, where he works on extensible programming languages, runtime systems, and applications of functional programming. He is one of the developers of the Racket programming language and a co-author of the introductory programming textbook *_How to Design Programs_*. He received his PhD from Rice University in 1999.

11:00 - 11:20 AM

Apache Cassandra Anti Patterns

Location: Peabody Lounge

Apache Cassandra is finding wide spread adoption across a variety of industries and as new people come on board with it, the same mistakes are often made. There is lots of information about how to use Cassandra but not too much information on what *not* to do. This presentation covers the most common, and often surprising, mistakes made when first deploying Apache Cassandra.

Listen to me explain (and complain) about the problems and mistakes (both in deployment and code), so you don't have to find out about them the hardway!

Matthew Dennis (@mdennis)**DataStax**

Matt was the first employee at DataStax (previously Riptano) and currently focuses on high level architecture, design, data models, deployment and algorithms for some of the largest, highest volume and most fault tolerant distributed systems in the world. Prior to DataStax, Matt held various technical leadership and principal development roles at Crossroads Systems and Troux Technologies and even spent time as a System Administrator in a former life. Matt studied Computer Science as an undergraduate and Software Engineering as a graduate student at the University of Texas at Austin in the College of Natural Sciences and the College of Engineering respectively.

Cross-Browser Testing with BrowserStack

Location: Centene Room

If you're serious about cross-browser testing, you probably have several browsers and VMs installed or you have multiple testing machines. Maintaining an extensive testing environment can take a lot of time and money. BrowserStack provides an elegant solution to the cross-browser testing problem with its on-demand VMs that you can access directly from your browser or via a simple API. In this session, we'll see what BrowserStack offers, how debugging in BrowserStack is likely easier than debugging on your local VMs, and how to leverage their API.

Scott Gonzalez (@scott_gonzalez)

Scott Gonzalez is a web application developer living in Raleigh, North Carolina. He has been

contributing to jQuery since 2007 and is currently a development lead for jQuery UI, and active in the node.js and WHATWG/W3C communities. Scott also writes tutorials on nemikor.com and is a co-author of the "jQuery Cookbook" from O'Reilly.

Compiling Scala to LLVM

Location: Midland States

This talk will present ongoing work to implement a new backend for the Scala compiler that targets the Low Level Virtual Machine (LLVM). LLVM aims to provide a universal intermediate representation for compilers to target and a framework for program transformations and analyses. LLVM also provides facilities for ahead-of-time and just-in-time native code generation. Targeting LLVM allows us to take advantage of this framework to compile Scala source code to optimized native executables. I will discuss the design and implementation of the backend and runtime system.

Geoff Reedy (@gereedy)

Geoff Reedy is a self-confessed programming languages geek from Albuquerque, NM. He cut his teeth with Perl and C, moved on to Python, then saw the light of strong static typing and functional programming in Scala, Haskell, OCaml (but he still likes Python sometimes). He holds equal interest in programming language theory, design and implementation. His current big project is an LLVM backend for the Scala compiler.

11:30 - 11:50 PM

Programming by Voice: Becoming a Computer Whisperer

Location: Centene Room

Two years ago I developed a case of Emacs Pinkie (RSI) so severe my hands went numb and I could no longer type or work. Desperate, I tried voice recognition. At first programming with it was painfully slow but, as I couldn't type, I persevered. After several months of vocab tweaking and duct-tape coding in Emacs Lisp and Python, I had a system that enabled me to code faster and more efficiently by voice than I ever had by hand. In a fast-paced live demo, I will create a small system using Clojure/Clojurescript, plus a few other languages for good measure, and deploy it without touching the keyboard. I hope to convince you that voice recognition is no longer a crutch for the disabled or limited to plain prose. It's now a highly effective tool that could benefit all programmers.

Tavis Rudd (@tavirudd)

Tavis Rudd is a programmer who literally talks to his computer.

Transactions: Overused or misunderstood?

Location: Peabody Lounge

Transaction processing has been at the heart of computing systems for over 4 decades: they provide an easily understood fault tolerance mechanism that ensures data consistency in the presence of failures and concurrent access. Many of the things we do each day are driven by transaction systems, yet others are not when perhaps they should. Why is this? Transactions, and specifically the two-phase commit they typically use, have come in for a lot of bad press over the years, with people citing performance overhead as a sufficient justification for not using them. In this presentation we'll look at transactions and try to address some of the concerns, whilst at the same time indicating where they should not be used. We'll also look at how transactions are evolving into new forms to address changes in middleware, such as Web Services, NoSQL, REST and mobile infrastructures.

Mark Little (@nmcl)**Red Hat**

I work for Red Hat, where I lead JBoss technical direction and research/development. Prior to this I was SOA Technical Development Manager and Director of Standards. I was Chief Architect and co-founder at Arjuna Technologies, an HP spin-off (where I was a Distinguished Engineer). I've been working in the area of reliable distributed systems since the mid-80's. My PhD was on fault-tolerant distributed systems, replication, and transactions. I'm also a professor at Newcastle University.

Numeric Programming in Scala with Spire

Location: Midland States

Spire aims to make generic numeric programming in Scala fast and easy. It provides a system of type classes that let you write generic numeric algorithms, abstracting away from the actual numeric type (eg. Double), without sacrificing the speed of a direct implementation. Spire also includes several new number types. There are types for complex and interval arithmetic, as well as new, high precision number types for rational arithmetic and exact geometric computation with the ability to guarantee the precision of your result. The goal of this talk is to introduce the audience to Spire's



AGILE EXPERTS. DELIVERY KINGS. IT'S WHAT WE DO.

THIS IS WHO WE ARE.

Asynchrony Solutions is an award-winning IT consulting firm that was established in 1999. We focus on systems integration, custom-application development and secure collaboration and serve the United States Department of Defense, other public-sector organizations and Fortune 1000 companies.

Asynchrony has grown consistently over the last six years and in 2005, its first eligible year, was named to Inc. magazine's list of the 500 fastest-growing private companies in the United States. Asynchrony has maintained the same key management team (led by its four founding principals) since its inception and has a stable workforce -- approximately 150 employees -- with very little turnover. Several of our employees have been with Asynchrony since its founding, and about 20% of our employees have been with us five years or longer.

Our history of high employee retention supports our track record of exceptional delivery, customer satisfaction and client retention.

We are always looking for top notch Developers experienced in Java, C#.NET, Android and iPhone/iPad Development. For additional information on our current needs check us out at jobs.asynchrony.com or send your resume to recruiting@asynchrony.com.

Asynchrony is an Equal Employment Opportunity Employer. Copyright ©2012. All Rights Reserved.



features and show how they can use it in their next project.

Tom Switzer (@tixxit) SickKids

Tom works at the Hospital for Sick Children in Toronto making web apps for bioinformaticians and biologists.

Erik Osheim (@d6)

Erik (@d6) builds geospatial applications at [Azavea](#). He is one of the authors of [GeoTrellis](#), a high-performance geoprocessing engine and programming toolkit written in Scala.

11:50 - 1:00 PM

Lunch

Location: Kiel Club

12:10 - 12:50 PM

Humanity 2.0

Location: Theater

Human technology has been evolving for almost two million years. The human race has been evolving for countless years more. How is humanity related to the technology we create? Recent technological advances have created critical problems for us, and the human race will soon be forced to confront these issues using the only tool we really have: technology itself.

How do we and our creations fit within the larger scope of the universe in general? Once "true scale" is applied to our cosmic situation, can we imagine better ways to utilize technology to expand our place within the cosmos?

Inspired by great scientists and thinkers from the past and today, this talk aims to pin down the human situation on an epic scale, identifying how and why we're in the position we are in, and what future paths we may take as we continue to progress our own existence.

Matthew Taylor (@rhyolight) Numenta

Matt Taylor is a Principal Software Engineer at Numenta, currently working to bring the intelligence within the neocortex of the human brain to a web service near you. He is originally from the St. Louis, Missouri area, but currently lives in Cupertino, CA with his wife of 10 years and two children.

Getting Physical: Networked Hardware with Node.js

Location: Peabody Lounge

Building networked web applications in the browser is fun, but wouldn't it be great to break out of the computer and get some physical, tangible interaction? Using Node.js, Arduino, and a variety of wireless and wired networking options, we can build custom game controllers, reactive interfaces and get feedback from cyberspace to meatspace.

This talk will walk you through a demonstration application (like controlling a networked pong game with a physical joystick) with basic electronics and Node.js and survey a number of networking interfaces, such as Ethernet, WiFi and XBee.

Ted Hayes Limina.Studio

Ted Hayes is a poet-inventor: conceiving objects and experiences that explore the sublime and the enigmatic through recombination and deconstruction. He is a proponent of what he has dubbed "æResearch Art," or art as science experiment, and actively investigates the themes, technologies and ramifications of autonomy, emergence, semiotics, pattern recognition, and neural networks.

Ted runs a research & development consultancy called Limina.Studio where he makes professional stuff like an internet-enabled bubble gun. See more of his work at <http://log.liminastudio.com>.

Computer architecture of the 1960's

Location: Pepsi Encore

Prof Peter Denning and Prof Jack Dennis wrote in a 2010 column of Communications of the ACM that the "æresearch of the 1960s and 1970s solved many problems that are being encountered today" and "æurge their resurrection."

One technology of 1960s that needs to be rediscovered is the Burroughs non-von Neumann stack/descriptor architecture in combination with the Algol language. It will be shown that this architecture eliminates many kinds of internet virus attacks and common programming errors (ask a Unisys Clear-Path user).

The structure of the Illiac IV supercomputer (circa 1969), a SIMD computer with 64 processing elements will also be described. The Illiac IV begat the Burroughs Scientific Processor (BSP) which had 16 processing elements and 17 memory units (to reduce memory access conflicts).

This presentation will also examine the evolution of computer languages from Algol 60, PL/360, BLISS, Algol W, PL/1, to C and C++. However, as Professor emeritus C. A. Hoare remarked: "[Algol] is a language so far ahead of its time that it was not only an improvement on its predecessors but also on nearly all its successors."

Slides of vintage ads will be shown to convey a sense of the times: "æYou are now reading the only language you need to program the Burroughs B 5500" (COBOL), or "æFlip-Flops for \$0.75, Hurry before the price goes down!" (Fairchild Semiconductor), and "æ100,000 multiply adds for \$100,000" (SDS attached processor).

The talk will conclude with a description of a speculative architecture combining attributes of the functional programming language Haskell, stacks, descriptors, and hierarchical memories. This architecture should ease the burden of programming the massively parallel computers being built today.

Carlton Mills

Carlton Mills' first job after graduation was to write the host OS for the Illiac IV supercomputer project at the University of Illinois (1967). He free-lanced for a while, finally working for Gcom, a small data communications start-up, from 1984 until retirement in 2007.

How Plato and Aristotle invented modern programming

Location: Midland States

The philosopher A. N. Whitehead once quipped that all philosophy is a footnote to Plato. So is all of computer science. This lively talk explores the philosophical systems devised by Plato and Aristotle, showing how Plato laid the foundations for what is now Object Oriented Programming, while Aristotle's dynamic model is at the core of functional programming. It is only now, 2500 years later, that we can clearly see the practical implications of their respective metaphysical systems. What lessons can we learn by revisiting the roots of Western philosophy?

Matt Butcher (@technosophos) Hewlett-Packard

Matt is a Developer Experience Engineer at HP Cloud. He is the author of seven technology books and numerous articles on topics including Java, JavaScript, and PHP development as well as LDAP. An advocate of Open Source, he maintains or contributes to numerous projects. And, yes, he's a real, live philosopher, too. He is a PhD candidate at Loyola University Chicago, where he specializes in Theory of Knowledge (Epistemology).

1:00 - 1:50 PM

Guess lazily! Making a program guess and guess well

Location: Pepsi Encore

Guessing is a part of life and science. We form a hypothesis, work out the consequences and compare with observations. Lots of problems are formulated by first assuming that the solution exists and then describing the properties it should have. Planning, scheduling, diagnostic, learning problems and Sudoku all follow this pattern. Guessing is good not only for describing these problems but also for solving them. We make a guess -- often a series of guesses -- to build, for example, a schedule, and check if it satisfies resource, timeliness and other constraints. Often, we guess again.

How do we write "guess the value of this variable" in code? How do we code "guess again"? How to put in prior knowledge favoring some guesses? The talk first will answer these questions.

Naive guessing however is hopeless even for toy problems. We often have to make lots of guesses before we build a candidate solution to check against the constraints. Only a tiny or even infinitesimal proportion of these guesses yield a successful candidate. How to make good guesses? That is very hard to know: Most physical, biological, sociological, etc. models are set up to compute consequences of the causes rather than the other way around. It helps to reformulate the question: how to avoid too many bad guesses? The talk will describe and illustrate a general principle, found in any serious logic, non-deterministic or probabilistic programming system.

The techniques explained in the talk are not tied to any language and can be used even in C. However, functional languages, especially typed ones, have a serious advantage. No prior knowledge of logic or non-deterministic programming is required. The ability to read introductory OCaml or Haskell code will be helpful. The participants will learn how to guess in their favorite language, and what it takes to succeed doing so. They will see laziness, unification and constraint propagation in the same light.

Oleg Kiselyov

<http://okmij.org/ftp/>

Oleg Kiselyov is a Computer Scientist in Monterey, CA. His web site is <http://okmij.org/ftp/>.

Stop Using Native HTML5

Location: Centene Room

No, don't stop using HTML5, just stop using the native HTML5 JavaScript API's directly. They're still in flux, there's still bugs that need to be shim'd, etc. Your code needs to use these features, but it needs to be more robust.

What you need is to use in your apps is a thin facade (wrapper API) around those features, so that as things change, bugs come and go, etc, all that needs to change is the internals of your facade, and not your actual app code.

"h5ive" is a project to build these thin facade APIs for the various HTML5 native APIs. We'll examine why we need something like this, and how h5ive will help us build more robust HTML5 apps.

**Kyle Simpson (@getify)
Getify Solutions**

Kyle Simpson is a JavaScript Systems Architect from Austin, TX. He focuses on JavaScript, web performance optimization, and "middle-end" application architecture. If something can't be done in JavaScript or web stack technology, he's probably bored by it. He runs several open-source projects, including LABjs, HandlebarJS, and BikechainJS.

Eventually Consistent Data Structures

Location: Theater

There are many reasons to use an eventually-consistent database -- like Riak, Voldemort, or Cassandra -- including increased availability, lower latency, and fault-tolerance. However, doing so requires a mental shift in how to structure client applications, and certain types of traditional data-structures, like sets, registers, and counters can't be resolved simply in the face of race-conditions. It is difficult to achieve "logical monotonicity" except for the most trivial data-types.

That is, until the advent of Convergent Replicated Data Types (CRDTs). CRDTs are data-structures that tolerate eventual consistency. They replace traditional data-structure implementations and all have the property that, given any number of conflicting versions of the same datum, there is a single state on which they converge (monotonicity). This talk will discuss some of the most useful CRDTs and how to apply them to solve real-world data problems.

**Sean Cribbs (@seancribbs)
Basho Technologies**

Sean is a hacker, writer, and speaker currently working for Basho Technologies, where he works on Riak, the distributed database. Before Basho, Sean was a freelance web developer and technical lead of the Radiant open-source web-publishing software which is built with Ruby on Rails. Prior to that, he studied Music Theory and Composition at the

graduate level and directed a chorus after receiving bachelors degrees in Computer Science and Music from the University of Tulsa.

Clojure +Datomic +Storm=Your Personal Digital Secretary

Location: Midland States

Everyone knows the value of networking. In fact, LinkedIn's motto is "Relationships Matter". Having a healthy, active network is a great benefit in all aspects of life - both personal and professional.

Unfortunately, networking doesn't come easily to most people. Most find it awkward, and let inertia and inactivity effect even existing relationships. Zolodeck fixes this problem by offering a digital secretary whose job it is to ensure you never lose touch with folks you know, and helps improve your network.

By combining Clojure, Datomic, Storm, and a host of artificial intelligence algorithms, Zolodeck makes your life easy by automating your networking. This is a story about the internals of this new service, and provides a whirlwind look at several state of the art components that it is built upon.

Clojure

- The DSL for professional networking
- Richardson Maturity Level 2 RESTful services

Datomic

- Separating databases and storage
- Different representations for different purposes -

Storm

- Stream processing to storm processing
- Updating relationship scores in real-time

Cloud

- dev-ops to no-ops

This is a breadth-first look at these pieces, and how they fit together to power the engine of a tiny startup. The talk describes the architecture of this service, and shows how a small team can leverage high-powered tools to amplify what is doable in the real-time, big-data space.

**Amit Rathore (@amitrathore)
Runa, Inc.**

VP of Engineering at Runa, Inc. - Author of Clojure in Action

Developers,

We're excited to be back with you at Strange Loop, and we're honored again to be a sponsor.

The work you do and the products you build truly inspire us. We're passionate developers as well, and we love building tools and services that allow you to painlessly accept payments so you can focus on what really matters: creating incredible stuff.

We hope you'll explore Braintree. We can do amazing things together.

Enjoy the conference and stay awesome,

- Hammer
Braintree Developer



[Try Our Sandbox](#)



basho

Android App Anatomy

Location: Peabody Lounge

Android has come a long way since version 1.0. Thanks to the compatibility library, we can use most of the hot new APIs on any Android device. In this talk, Eric Burke shows how to craft modern Android applications in a clean, maintainable way using the newest APIs.

Topics include:

- Lifecycle deep dive: loaders, fragments, and dialogs
- One activity, many fragments
- Avoiding the God object with an event bus
- Ensuring big images don't break your app
- Following Android design guidelines on any device

Eric Burke (@burke_eric)
Square

Eric Burke is the Android Engineering Manager at Square where he works on the Pay with Square and Square Card Reader Android applications. Eric has presented at technical conferences, taught many Java classes, and even wrote three books for O'Reilly. He is passionate about creating delightful, pixel-perfect user interfaces on Android.

2:00 - 2:50 PM

Up up and Out: Scaling software with Akka 2

Location: Pepsi Encore

We believe that one should never have to choose between productivity and scalability, which has been the case with traditional approaches to concurrency and distribution. The cause of that has been the wrong tools and the wrong layer of abstraction and Akka is here to change that. Akka is a unified runtime and programming model for scaling both UP (utilizing multi-core processors) and OUT (utilizing the grid/cloud). With Akka 2 this will be taken to a whole new level with its "œDistributed by Design". Akka 2 provides location transparency by abstracting away both these tangents of scalability by turning them into an operations and configuration task. This gives the Akka runtime freedom to do adaptive automatic load-balancing, cluster rebalancing, replication and partitioning. In this talk you will learn what Akka is

and how it can be used to solve hard scalability problems. Akka is available at <http://akka.io> (under Apache 2 license).

Jonas Boner (@jboner)
Typesafe

Jonas Boner is a husband, dad, programmer, teacher, speaker, musician, writer and Java Champion. He is the CTO of Typesafe and is an active contributor to the Open Source community; most notably created the Akka Project, AspectWerkz Aspect-Oriented Programming (AOP) framework, committer to the Terracotta JVM clustering technology and been part of the Eclipse AspectJ team. <http://jonasboner.com>

ql.io: Putting Node.js to Work for Mobile

Location: Peabody Lounge

ql.io makes it easy to build HTTP gateways to reduce network hops, cut bandwidth usages, and improve overall latency and availability for apps that are far away from data centers - such as mobile apps. At eBay we have been implementing and serving several end-user facing use cases through ql.io. These apps use ql.io's SQL and JSON flavored scripting language to describe data access from HTTP APIs.

While building a functionally rich and scalable HTTP gateway on node.js to meet our performance and scalability goals itself was a challenging task, building ql.io also involved making node.js operationally ready to meet eBay's availability standards.

In this talk, I will introduce the key concepts behind built ql.io, how we're putting ql.io and node.js to action, what it took us to make node.js operationally ready, and the lessons we learned in this process.

Subbu Allamaraju (@sallamar)
eBay

Subbu Allamaraju is the creator of ql.io and leads the development and operationalization of ql.io and related projects at eBay. He is well-known for his work on RESTful Web Services Cookbook.

The Audubon Society for Partial Failures

Location: Theater

Failure in a modern distributed system is a complicated affair. Many distributed systems are deployed into production with multiple bugs and can limp along on one leg for months due to the self-healing properties of their highly available architecture. Be that as it may, apply enough load and eventually things will cease to work when you need them the most. This talk presents a taxonomy

of distributed systems failures and bugs in the wild, as seen through the lens of the network. By classifying the failures we find, we can come closer to being able to proactively detect them before they develop into full blown outages.

Cliff Moon (@moonpolysoft)
Boundary

Cliff Moon is Founder and Chief Technical Officer at Boundary. Prior to Boundary, Cliff was a lead engineer for Powerset (natural language search engine acquired by Microsoft) where he was instrumental in the design, implementation, launch, and operation of many of the company's production services. Cliff is an active contributor to open source projects, developing the first open-source implementation of Amazon Dynamo and originating the Dynamo Framework. Cliff is an active and well-regarded member of the NoSQL, Scala, and Erlang communities.

Wolfram's data analysis platform

Location: Midland States

Wolfram | Alpha is the attempt to make accessible to the public a large corpus of sophisticated computations on curated data. The end goal of Wolfram | Alpha is to encode all of algorithmic human knowledge.

Recently, Wolfram | Alpha began to tackle user-uploaded tables, images, text, and other types of data. I'll do a demonstration, talk about some of the underlying ideas and algorithms, and discuss how we are complementing Mathematica with Scala to allow for more elaborate data processing and analysis pipelines.

Taliesin Beynon (@taliesinb)
Wolfram Alpha LLC

I studied Math and Physics, although I was always a hacker at heart. I've worked previously at computer vision and cloud computing startups. I run an R&D lab inside Wolfram which does new things. We were responsible for the data upload feature of Pro (<http://blog.wolframalpha.com/2012/02/09/launching-a-democratization-of-data-science/>).

AngularJS - HTML redesigned for web apps

Location: Centene Room

HTML is great for declaring static documents, but it falters when we try to use it for declaring dynamic views in web-applications. AngularJS lets you extend HTML vocabulary for your application. The resulting environment is extraordinarily expressive,

readable, and quick to develop. Other frameworks deal with HTML's shortcomings by either abstracting away HTML, CSS, and / or JavaScript or by providing an imperative way for manipulating the DOM. Neither of these address the root problem that HTML was not designed for dynamic views.

Misko Hevery (@mhevery) Google

Misko Hevery works as an Agile Coach at Google where he is responsible for coaching Googlers to maintain the high level of automated testing culture. This allows Google to do frequent releases of its web applications with consistent high quality. Previously he worked at Adobe, Sun Microsystems, Intel, and Xerox (to name a few), where he became an expert in building web applications in web related technologies such as Java, JavaScript, Flex and ActionScript. He is very involved in Open Source community and an author of several open source projects such as Angular (<http://angularjs.org>) and JsTestDriver (<http://code.google.com/p/js-test-driver>).

3:30 - 4:20 PM

Expressing abstraction - Abstracting expression

Location: Theater

What is expressiveness? How do you quantify the power of a programming language? How do you design programming languages to maximize productivity? Why are new languages still being designed? Shouldn't we have created the perfect one by now? After over 60 years of language design, these are still hard questions to talk about. In this presentation I will talk about how I think about expressiveness and language design. I will present some of things I've discovered about abstraction, language power and other aspects related to programming languages.

Ola Bini (@olabini) Thoughtworks

Ola Bini works as a language geek for ThoughtWorks in Chicago. He is from Sweden but don't hold that against him. He is one of the JRuby core developers and have been involved in JRuby development since 2006. At one point in time, Ola got tired of all existing programming languages and decided to create his own, called Ioke. Then he did it again, and started work on Seph. He has written a book called Practical JRuby on Rails Projects for APress, and coauthored Using JRuby for the Pragmatic Programmers, talked at numerous conferences, and contributed to a large amount of

open source projects. He is also a member of the JSR292 Expert Group.

His main passion lies in implementing languages, working on regular expression engines and trying to figure out how to create good YAML parsers.

Ozma, an extension of Scala with Oz concurrency

Location: Centene Room

The programming language Ozma is an extension of Scala with concurrency concepts imported from Oz [1]. It is based on deterministic dataflow concurrency. This is extremely simple: 'val' values in Ozma are all single assignment variables, and all operations synchronize transparently on this assignment. This extension has the incredible property that race conditions are impossible. Combined with lightweight threads, deterministic dataflow makes concurrent programming effortless; it is much simpler than any other model we know of. In addition, Ozma integrates both lazy execution and message passing with the dataflow concurrency.

Ozma has been successfully implemented [2] on the Mozart programming system, which was built specifically for Oz and has strong support for declarative concurrency. There does not exist an implementation on the JVM yet.

This talk first gives a hands-on presentation of concurrent programming in Ozma. Afterwards, we explore the issues arising when implementing Ozma on the JVM, and give solutions or workarounds for these issues. [1]<http://www.mozart-oz.org/> and <http://ctm.info.ucl.ac.be/> [2] <https://github.com/sjrd/ozma>

Sebastien Doeraene (@sjrdoeraene) University Catholique de Louvain

Sebastien Doeraene is a PhD student at the University Catholique de Louvain, Belgium. His research is focused on the use of conflict-free replicated data types for large-scale distributed programming, from a language design point of view. He is the lead designer and developer of Ozma, an extension of Scala with Oz concurrency, as well as co-lead designer and developer of Mozart2, a new implementation of Oz. Outside of the programming language community, he was also the winner of the programming contest Prologin in 2010.

Go: code that grows with grace

Location: Pepsi Encore

One of the Go Programming Language's key design goals is code adaptability; that it should be easy to take a simple design and build upon it in a clean and natural way. In this talk I describe a simple "chat roulette" server that matches pairs of incoming TCP connections, and then use Go's concurrency mechanisms, interfaces, and standard library to extend it with a web interface and other features. Although the function of the program changes dramatically, the inherent flexibility of Go allows the original design to remain intact as it grows.

Andrew Gerrand (@enneff) Google

Andrew Gerrand is a Developer Advocate for the Go team at Google Sydney. His mission is to teach the world to love Go as much as he does, one audience at a time.

Building visual, data-driven UIs with ClojureScript

Location: Peabody Lounge

Modern web browsers provide a versatile foundation for graphical, interactive applications. ClojureScript brings rich semantics to the browser and, combined with Clojure on the server, gives developers an insane amount of leverage to build novel tools. Kevin will discuss using state machines with data-driven DOM-manipulation to build visual applications on the web and iPad.

He will evince how ClojureScript's rich data structures and abstractions foster declarative programming, and outline some testing and development benefits from shared client/server language. Concrete examples will be drawn from data visualization applications in the energy and biotech fields.

Kevin Lynagh (@lynaghk) Keming Labs

Kevin visualizes data and makes statistical interfaces on the web. He has written enough JavaScript to be terribly excited about ClojureScript. Before Clojure he wrote machine learning and analytics tools in R and Scala. In 2010 he wrote a thesis on protein structure, for which Reed College inexplicably awarded him a physics degree. Kevin lives in Portland, Oregon, and spends as much time rock climbing as he does in the REPL.

Designing Fault Tolerant Distributed Applications

Location: Midland States



PLATFORM ENGINEERING

▶ JOIN OUR TEAM



WHAT WE DO

Platform Engineering is responsible for delivering a rock-solid gaming foundation that keeps up with our amazing worldwide growth. While others are experimenting with web-scale solutions, our platform team is leveraging the latest open source technologies ([Java](#), [Rails](#), [Chef](#), [Grails](#), [NoSQL/BigData](#) and [Scala](#)) to build one of the largest gaming platforms in the world. If you're interested in solving unique scalability problems, Riot is looking for you!

WHO WE ARE

Riot Games is a fast-paced company on the cutting edge of the games-as-a-service industry, developing serious games but without taking ourselves too seriously. At Riot, you will have the opportunity to work with a veteran development team anchored by accomplished leaders in engineering, design, art and publishing. With offices in St. Louis and Los Angeles, Riot Games provides competitive salaries and benefits and the opportunity to create some of the best and most frequently played games in the world. (New expanded St. Louis office opening late 2011.)



HOW TO JOIN US

Riot is actively looking for high-octane engineers that love the challenge of leveraging the latest technologies to build a global-scale gaming platform. Please visit: www.riotgames.com/careers for open platform (PvP.net) positions or email your resume to sdelap@riotgames.com.



PVP.NET

APPLY AT WWW.RIOTGAMES.COM/CAREERS

Today's applications and services are expected to be "distributed by default." Operational requirements demand that new applications be deployable on one server or a hundred - coordinating work, balancing load, and handling failures cleanly by redistributing work when nodes fail. While this is straightforward for stateless, shared-nothing applications like web or worker tiers, those which require coordination prove more complex.

Faced with this challenge in building a distributed streaming analytics engine and its supporting self-coordinating, fault-tolerant services, Boundary's developed and open-sourced a JVM-based clustering library called Ordasity [1]. Since its release last fall, community members have contributed documentation, patches, along with an additional implementation written in Go.

Built atop ZooKeeper, Ordasity is an implementation of a pattern for creating masterless, self-organizing services. With this pattern abstracted away, engineers can write clustered services in as little as 25 lines of code backed by solid operational tooling that provides graceful draining, hot handoff of work between nodes, continuous automatic load balancing, and a clean admin interface.

Using this framework as an example without restricting focus to it alone, this session offers a deep dive into techniques for coordination, load balancing, methods of containing cascading failures, and designing for resilience in distributed applications, along with a demo of building a service that accomplishes the same in five minutes.
--- [1] <https://github.com/boundary/ordasity>

Scott Andreas (@cscotta)
Boundary, Inc.

Engineer at Boundary, Inc. hacking on distributed stream analytics systems in Scala and Java. Former engineer at Urban Airship designing and implementing low-latency messaging systems for mobile devices.

4:30 - 5:20 PM

Taking Off the Blindfold

Location: Theater

Imagine a painter who steps up to the easel, ties a blindfold around his eyes, and begins to paint. Every few minutes he peeks under the blindfold, catches a brief glimpse of how the painting compares to the picture in his head, and then tugs the blindfold back down to apply a few more brushstrokes.

As programmers, we think of ourselves as producing code, but code is merely the medium, a means to an end. The purpose of code is to manipulate data, and patterns of data are our true

product. And as data artists, we work blindfolded, grasping at invisible data structures, imagining them in our heads, occasionally executing our programs for a brief peek at reality before continuing to blindly code.

This talk will demonstrate techniques and environments that make the dynamic state of a program visible and tangible, and encourage programmers to examine and respond to data as part of the coding process. Throw away the blindfold, and dare to see the data while creating the code.

Bret Victor (@worrydream)
worrydream.com

Bret Victor invents tools that enable people to understand and create. He has designed experimental UI concepts at Apple, interactive data graphics for Al Gore, and musical instruments at Alesis.

Scaling scalability: Evolving Twitter Analytics

Location: Midland States

Over the past few years, Twitter's Hadoop-based infrastructure has experienced explosive growth in a number of dimensions -- terabytes stored, jobs processed, nodes active, and number of engineers producing and consuming data. Along the way we encountered a number of challenges despite the data-scalability of Hadoop-based technologies. In this talk, I will describe these problems and the solutions we developed as we expanded from 30 nodes and one team of 3 people to many hundreds of nodes, multiple teams, dozens of people, and thousands of daily jobs.

Dmitriy Ryaboy (@squarecog)
Twitter, Inc

Dmitriy Ryaboy (@squarecog) leads Twitter's Analytics Infrastructure team. He's previously worked at Cloudera, Ask.com, and Lawrence Berkeley National Laboratory. He holds a Master's degree in VLIS from CMU and a Bachelor's in EECS from UC Berkeley.

Emerging Solutions for CSS Layout

Location: Peabody Lounge

In this session, attendees will learn about the current work of the W3C's CSS Working Group in solving a long-standing design problem in CSS: Layout. Topics covered will include Flexible Box Model, Multicol, Grids, Template Layout, Paged Media and

Generated Content for Paged Media, and the truly awesome CSS Regions.

Molly Holzschlag (@mollydotcom)
Molly.Com, Inc.

Molly E. Holzschlag has spent the majority of her 25 years in IT working with the Internet, and observed the Web emerge and evolve from its first days at CERN. She has spent the years since educating herself and others about the Web, its languages, and its core philosophies. Warm, colorful, and interested in everyone's stories and successes, be sure to come say hello!

Project Lambda in Java 8

Location: Pepsi Encore

Java SE 8 will include major enhancements to the Java Programming Language and its core libraries. This suite of new features, known as Project Lambda in the OpenJDK community, includes lambda expressions, default methods, and parallel collections (and much more!). The result will be a next-generation Java programming experience with more flexibility and better abstractions.

This talk will introduce the new Java features and offer a behind-the-scenes view of how they evolved and why they work the way that they do.

Daniel Smith
Oracle

Daniel Smith is a Java language designer and compiler developer at Oracle. He currently leads the specification efforts for JSR 335: Lambda Expressions for the Java Programming Language. He received a PhD in Computer Science from Rice University, doing research on type inference in object-oriented languages. He previously worked at LogicBlox, developing a database query language grounded in logic programming.

The Fragment Transition

Location: Centene Room

Android is evolving, but developers aren't keeping pace.

Fragments, which let you break your UI into smaller functional pieces, were released over a year ago. Fragments enable scaling of your app to tablets and the next generation of devices. There's a huge opportunity to lead the market, yet few businesses and developers are leading the charge.

Corey will show you techniques for starting a fresh project, renovating an existing Android code base, optimizing for tablets, and balancing use of new APIs, while remaining backward compatible. She

has been using fragments as long as they have been around in the XfinityTV app as well as her personal projects. She'd never start a new project without them!

science in computer science from the University of Illinois. Eich and his wife have five children.

Corey Latislaw (@corey_latislaw)

Chariot Solutions

Corey Leigh Latislaw is a Senior Mobile Architect at Chariot Solutions. She helps businesses determine their mobile needs and then builds them the appropriate solutions. She created the Philadelphia Android Alliance, a rebel army whose mission is to conquer Apple's market share, and co-founded a start-up that builds apps for the greater good. Corey is passionate about encouraging broader participation in computing sciences, local food, the outdoors, photography, and living in downtown Philly. In a past life, she helped build the XfinityTV Android & iOS applications at Comcast Interactive Media and worked on various security projects at Cisco Systems. You can find her at coreylatislaw.com.

5:30 - 6:20 PM

The State of JavaScript

Location: Theater

JavaScript notoriously began life as a quirky rush-job built under duress and the shadow of Java. It has evolved through 17 years of valuable lead-user library and web application construction, plus growing attention from browser vendors and the governing "Ecma" standards body, into a popular and fast dynamic programming language.

JS has also become the only candidate in sight for the needed role of safe, cross-browser, low-to-mid-level object file format for the Web, with dozens of programming languages compiling to it and its browser runtime APIs. I will demonstrate the latest language extensions being standardized as ES6, and show how surprisingly efficient a modern JavaScript virtual machine can do running non-trivial programs written in languages such as C++ compiled to JS.

Brendan Eich (@BrendanEich)

Mozilla

Brendan Eich is CTO of Mozilla and widely recognized for his enduring contributions to the Internet revolution. In 1995, Eich invented JavaScript (ECMAScript), the Internet's most widely used programming language. He also co-founded the mozilla.org project in 1998, serving as chief architect. Eich helped launch the award winning Firefox Web browser in November 2004 and Thunderbird e-mail client in December 2004. Today, Eich's central focus is guiding the future technical work to keep Mozilla vital and competitive. He holds a bachelor of science in math and computer science from Santa Clara University and a master of