Strange Loop 2010 Program

Keynote Sponsors

Party Sponsors

Refreshment Sponsors

Session Sponsors

Lanyard Sponsor

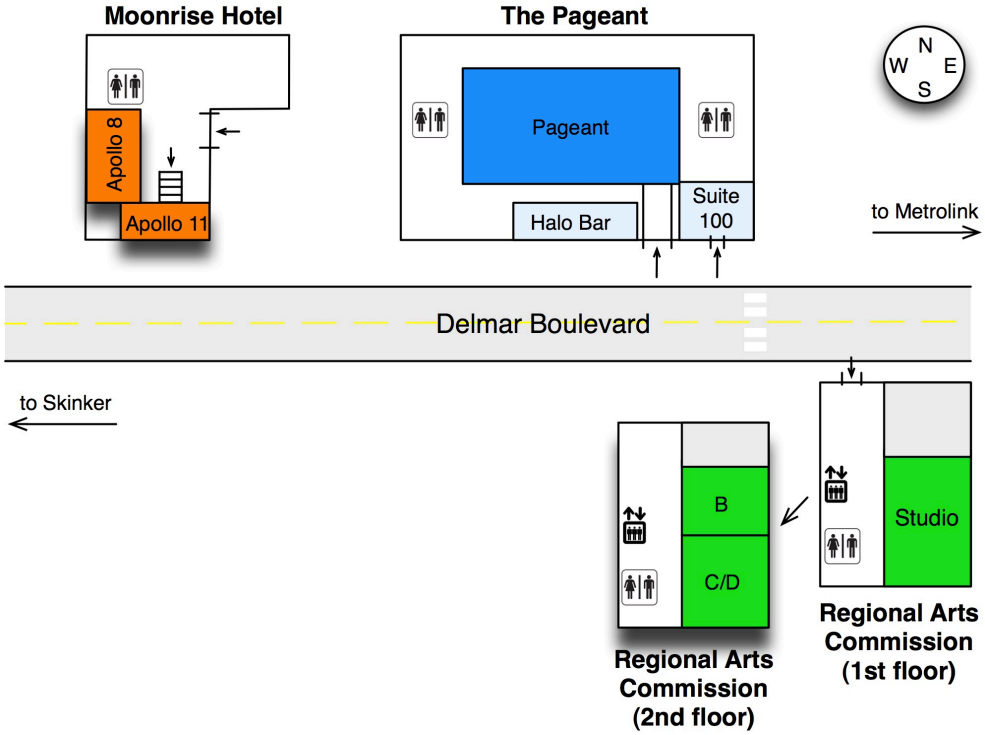
Video Partner

Whiteboard Partner

Supporting Sponsors

Parking

Venue Map



Wednesday, Oct. 13th

6-8 pm - Moonrise Hotel Lobby Mezzanine - Welcome Reception and Registration

If you'd like to pick up your registration materials early and meet other attendees, stop by the Moonrise Hotel the night before the conference. The reception will take place on the mezzanine above the lobby and will include an open bar.

Thursday, Oct. 14th

	Apollo 8	Apollo 11	Pageant	RAC Studio	RAC C/D	RAC B
8:30-8:45			**Welcome** - *Alex Miller*			
8:45-9:45	**Machine Learning: a Love Story** - *Hilary Mason*					
10:00-10:50	**Open Source EDA** - *Edward Yavno*	**Riak: From Small to Large** - *Rusty Klophaus*	**The Semantic Web: Hot or Not?** - *Brian Sletten*	**The Evolution of the Flickr Architecture, 2010 Edition** - *Mikhail Panchenko*	**Gradle - The Polyglot Build System** - *Ken Sipe*	**A Little Bit About Terracotta BigMemory** - *Steve Harris*
11:00-11:50	**Automate or Die** - *Trotter Cashion*	**GoLightly: Building VM-based Language Runtimes in Go** - *Eleanor McHugh*	**Adopting Apache Cassandra** - *Eben Hewitt*	**Making Your Open Source Project More Like Rails** - *Yehuda Katz*	**That's My App, Running in Your Background** - *Michael Galpin*	**Gaelyk: Lightweight Groovy on the Google App Engine** - *Tim Berglund*
	Lunch - on your own					
1:30-1:50	**Triplestore Testing in the Cloud with Clojure** - *Ryan Senior*	**Lua, Tiny Embeddable Scripting that Doesn't Suck** - *Kyle Cordes*	**Immutable Object vs Unsynchronized State** - *Joshua Bennett*	**HyperGraphDB - Data Management for Complex Systems** - *Borislav Iordanov*	**Civic Hacking** - *Luigi Montanez*	**You Already Use Closure (in Ruby)** - *Justin Love*
2:00-2:50	**Perl 6 - The Apocalypse** - *Matt Follett*	**Real World Modeling with MongoDB** - *Steve Smith*	**NoSQL at Twitter** - *Kevin Weil*	**Mobile HTML 5.0** - *Michael Galpin*	**Dude, That's Some Strange UI Architecture** - *Kyle Simpson*	**Java Provisioning in the Cloud** - *Adrian Cole*
2:50-3:15	***Break - sponsored by Lisa Rokusek / AgentHR***					
3:15-4:05	**Unifying the Search Engine and NoSQL DBMS with a Universal Index** - *Chris Biow*	**Best Bugs, Best Practices, and Best Tools for Hibernate** - *Patrycja Wegrzynowicz*	**Parallel Programming with Node.js** - *Ryan Dahl*	**Clojure's Solutions to the Expression Problem** - *Chris Houser*	**Complexity Theory and Software Development** - *Tim Berglund*	**Securing and Extending Puppet for World Domination** - *Richard Crowley*
4:15-5:15	**How To Think About Parallel Programming: Not!** - *Guy Steele*					
	Dinner - on your own					
6:30-9:30	***Party featuring Strange Passions track***					

8:30-8:35 am - Pageant - “Welcome” - Alex Miller

Welcome to the conference and any last minute updates and announcements.

8:45-9:45 am - Pageant - “Machine Learning: A Love Story” - Hilary Mason
Machine learning has come a long way in recent years -- from a long-marginalized field so old it still has the word "machine" in the name, to the last, best hope for making sense of our massive flows of data.

The art of 'data science' is asking the right questions; the answers are generally trivial or impossible. This talk will focus more on questions than on answers. I'll give a brief history of the field with a focus on the fundamental math and algorithmic tools that we use to address these kinds of problems, then walk through several descriptive and predictive scenarios.

Finally, I'll show one example system using bit.ly data in-depth, from the backend infrastructure through the algorithms and data processing layer to show a functioning product.

Attendees should expect to hear some good stories of data gone right and data gone awry, and walk away with a few new clever tricks.

10-10:50 am - Apollo 8 - “Open Source EDA” - Edward Yavno
From Realtime Web applications to low latency Trading Systems, Event Driven Architecture (EDA) is used to build scalable, distributed, time sensitive applications. This presentation will demonstrate examples and will concentrate on practical hands-on implementation of EDA systems, its layers and components. It will introduce Open Source Java EDA Stack consisting of established open source products, will review their APIs and will show how to put it all together to build a distributed event driven system.

10-10:50 am - Apollo 11 - “Riak: From Small to Large” - Rusty Klophaus
Riak (http://wiki.basho.com), a Dynamo-inspired, open-source key/value datastore, was built to scale from a single machine to a 100+ server cluster without driving you or your operations team crazy. This presentation points out the characteristics of Riak that become important in small, medium, and large clusters, and then demonstrates the Riak API via the Python client library.

10-10:50 am - Pageant - “The Semantic Web: Hot or Not?” - Brian Sletten
What the hell is the Semantic Web and why have we been talking about it for 16 years?

Is it A.I.? Is it Web 3.0? Is it exciting and imminent or can you continue to ignore it? Tim Berners-Lee initiated the development of the most amazing information exchange platform we could have imagined and it has changed nearly every aspect of our lives. Interestingly, the Web we know is only a fraction of what he intended. He has since been championing extensions to this platform (part of the original vision!) to improve the machine processability of the Web. We are seeing the emergence of not just a Web of Documents, but a Web of Data too. As we watch it unfold, there is every reason to believe it will have an even bigger impact on our worlds, how we communicate and how we share information.

That's at least the story. What is the truth?

We are seeing rapid development and adoption of these Semantic Web technologies in both public and private ways. The truth is it is mostly very hot, it will change everything and you should absolutely care. I will tell you why.

10-10:50 am - RAC Studio - "The Evolution of the Flickr Architecture, 2010 Edition" - Mikhail Panchenko
Flickr was created over 6 years ago - an eternity in tech years. While the site's general architecture has has held up fairly well, some pieces have started creaking at the seams. A new crop of engineers has started evolving the site's architecture to take advantage of new technologies.

We'll discuss what the Flickr team is doing to keep up with the times while operating a quickly-aging, labyrinthian codebase. Topics will include, but will not be limited to:

* Data Architecture:
  * MySQL
    * why it's still our hammer - some numbers
    * the tools we use to make managing MySQL easier
    * knowing when it's not the right tool for the job
  * Redis: picking up MySQL's slack
* PHP
  * testing in a legacy, procedural codebase
  * introducing objects and exceptions without breaking things
* Java: the dark horse; how we use various specialized daemons to help scale the site
* Hadoop: crunching the numbers
* Frontend performance: Going from 5 seconds to below ~1
* Deploying: we still deploy ~10 times a day and it's still awesome

The topics will be accompanied by appropriate war stories. Come hear what a new generation of engineers is doing to keep Flickr running smoothly as it continues to grow. If you're not interested in that, come laugh at the tales of all the unbelievably dumb things we've done over the years.

10-10:50 am - RAC C/D - "Gradle - The Polyglot Build System" - Ken Sipe
In the Java build space, first there was ANT, which provided a reliable way to build without an IDE. Then there was Maven, which provided standardization in build life cycles and dependency management. Yet there still seems to be frustrations with maintaining a good build system... whether it is just too much XML or too many POMs. Frankly XML is just limiting as a DSL for describing a build for anything that falls outside of what the original builders of the framework envisioned. Gradle provides a solution that provides convention over configuration approach to the build process and an approach at building that isn't based XML.

This session assumes no familiarly with Gradle as it introduces this new approach at building projects. It is very helpful to be able to read and understand groovy to get the most from the session. This session will look at multi-language or polyglot projects, as well as integration to ANT and Maven. It will conclude with building custom plugins for the Gradle build process.

10-10:50 am - RAC B - "A Little Bit About Terracotta BigMemory" - Steve Harris
In talking to our users it is clear that applications are getting more and more data hungry. According to IDC, data requirements are growing at an annual rate of 60 percent. There is good news though. Server class machines purchased this year have a minimum of 8 Gig of RAM and likely have 32 Gig. Cisco is now selling mainstream UCS boxes with over 380 Gig of RAM. Memory has gotten big and extremely cheap compared to things like developer time and user satisfaction.

Unfortunately a problem exists as well. For Java/JVM applications it is becoming an ever increasing challenge to use all that data and memory due to GC Pauses.

In this talk I'm going to cover the problems we identified and the technology we built to solve those problems.

I'll cover:
- A bit about it's history and the history of the problem
- The where, when why of BigMemory
- Throughput, latency, Garbage Collection, SLA and scaling characteristics
- Configuration of Ehcache with BigMemory in an existing application with just a few lines of config code
- Ehcache's tiered storage architecture: MemoryStore, the OffHeapMemoryStore and the DiskStore
- Ehcache BigMemory with scale-out
- Best practices
- Implications for your caching architecture

11-11:50 am - Apollo 8 - "Automate or Die" - Trotter Cashion
System provisioning is all too often a cardboard house held together by duct tape. Different versions of the same software live on each machine, leading to difficult to diagnose, "not on my machine" bugs. Without automation, system provisioning is an impossibly hard problem that can destroy a business. Thankfully, Chef helps solve this problem through declarative build scripts that bring all machines into alignment. In this talk, we'll discuss how to get started with chef-solo, what to do when it's not enough, and how to fit it into your organization. Come hear how Chef can make your life better, and how easy it is to use.

11-11:50 am - Apollo 11 - "GoLightly: Building VM-based Language Runtimes in Go" - Eleanor McHugh
The GoLightly library (http://github.com/feyeleanor/GoLightly/) is a toolkit for building flexible virtual machines. Instead of limiting you to a particular execution model it

defines basic primitives which are useful in implementing stack-, register- and vector-based execution models along with simple communications models for multicore designs.

GoLightly started as a research tool to help design a high-performance virtual machine for Ruby written in Go but its scope has since expanded to be a general-purpose library with the aim of allowing any high-level language runtime to easily exploit multicore concurrency and the vector-processing features of modern consumer processors.

In this session we'll draw on the GoLightly and related codebases to explore how Go supports concurrency as well as examining its approach to object-orientation and type safety - including the dirty tricks occasionally required to override it.

Here we'll also encounter the Go testing and benchmarking framework and use code examples to illustrate the many trade-offs involved in VM design including some or all of: dispatch models; bytecode and threaded interpretation; control flow and activation records; interrupts; memory allocation; stack- and register- architectures; vector-processing (SIMD) extensions; JIT and AOT compilation; system calls, blocking and processor-inspired weirdness.

By the end of the session attendees should be comfortable reading Go source code, have a basic feel for developing with the language and the necessary background to write their own VMs in whatever language they happen to prefer.

11-11:50 am - Pageant - “Adopting Apache Cassandra” - Eben Hewitt
The Cassandra database is distributed, highly-available, fault-tolerant, and offers an elastic scaling model—all of which make Cassandra a powerful proposition for mission-critical applications. It’s used by many of the world’s biggest web properties, including Facebook, Twitter, Digg, StumbleUpon, Reddit, Cisco, and others.

This is all fantastic, but there’s no free lunch—Cassandra is not a relational database, but rather follows in the footsteps of columnar data stores such as Google BigTable and Amazon’s Dynamo. As such, getting your head around how Cassandra works can be daunting to say the least: there’s a lot of new terminology (what’s a Hinted Handoff? What’s a SuperColumn?? What do I need to know about Vector Clocks??? Argh!).

There are some complex algorithms in Cassandra, and new ways of handling basic operations in order to achieve the benefits mentioned above. Cassandra only recently emerged from Incubator status, and there aren’t a lot of tools available yet to smooth your path toward adoption. This talk can help you understand everything you need to know to get started using Cassandra. We’ll sort out all the terminology and foundational concepts, and then dive into a practical set of ways to get started putting Cassandra to work in your applications today.

11-11:50 am - RAC Studio - “Making Your Open Source Project More Like Rails” - Yehuda Katz
Ruby on Rails has shown the world that it's possible to build an open-source project that is optimized for developer happiness. In the five years since it was originally announced, Rails has gone through the typical hype cycle: enthusiastic early adopters followed by negative press attention followed by a quieter period of productivity.

What challenges did Rails face as it made its way to being a successful, well-regarded framework, and how did it overcome them? In this talk, Yehuda will talk about how the Rails team faced its challenges, and what lessons you can take away when working on your own open source project.

11-11:50 am - RAC C/D - “That’s My App, Running in Your Background” - Michael Galpin
You have seen the ads where Android based devices like to brag about how awesome their multitasking is and now even the iPhone claims to have multitasking. Unfortunately it’s pseudo-multitasking borrowed from Android, but fear not. Android has “real” multitasking as well. It’s easy to do, but even easier to screw up. In this talk you’ll learn how to do it right, and how to do it without killing a phone’s battery. We’ll discuss the dreaded “P” word (polling), as well as alternatives such as Android’s cloud to device messaging and persistent connections.

11-11:50 am - RAC B - “Gaelyk: Lightweight Groovy on the Google App Engine” - Tim Berglund
You love Groovy and you're a believer in cloud computing. For a larger project you might choose Grails and hosting on Amazon EC2, but what if you want to take advantage of the nearly massless deployments of a cloud provider like the Google App Engine? You could make Grails work, but it's not always the best fit. Enter Gaelyk.

Gaelyk is a lightweight Groovy web application framework built specifically for the Google App Engine. In this session, we'll talk through the simple abstractions it offers, then show how easy it is to code and deploy a useful application to the cloud.

1:30-1:50 pm - Apollo 8 - “Triplestore Testing in the Cloud with Clojure” - Ryan Senior
Determining what RDF repository to use for a project can be a daunting task. With so many repository choices, benchmarks and usage scenarios, where do you start? This talk discusses how Revelytix answered that question. The talk will cover the test framework written by Revelytix in Clojure, including a language for defining tests, a harness for executing the tests and using CouchDB to store the results. Example Clojure code will be included along with a discussion around the available RDF benchmarks. The talk will also discuss the test harness using EC2 instances for cheap performance testing and how we interpreted those results using Incanter.

1:30-1:50 pm - Apollo 11 - "Lua, Tiny Embeddable Scripting That Doesn't Suck" - Kyle Cordes
This talk will show how and WHY to use Lua (as opposed to the zillion other scripting languages) for embedded scripting inside of larger, non-Lua projects. Lua is safe, fast, simple, learning, and more popular that you might expect.

1:30-1:50 pm - Pageant - "Immutable Object vs Unsynchronized State" - Joshua Bennett
Maintaining state is a common place among today's complex systems, and choosing how systems interact with this state is one of the earliest design decisions that is made. With the rise of the multi-core processor, concurrency is becoming more and more common place and dealing with state transforms into a potential debugging nightmare. In this session we will discuss the difference between mutable and immutable state; how your systems behave when dealing with mutable versus immutable state; as well as learn when and where the best fits are for mutable and immutable state. Finally we will finish up with some common mutable (and immutable) anti-patterns and learn how to avoid them.

1:30-1:50 pm - RAC Studio - "HyperGraphDB: Data Management for Complex Systems" - Borislav Iordanov
While the problem of handling massive amounts of data has been at the forefront of database research both in industry and in academia, addressing the complexity of domain models has remained solely a concern of application architects forced to align often highly incompatible problem and solution domains.

HyperGraphDB is a database with a unique memory/data model based on generalized hypergraphs. Those are graphs where edges can point to an arbitrary number of nodes and even to other edges. Thus higher order relationships are expressed naturally which automatically solves most headaches related to domain data modeling. Entities (nodes and edges) have arbitrary values managed by a comprehensive type system embedded as a hypergraph itself.

In a sense, HyperGraphDB is a dynamic-schema database general enough to easily accommodate any meta-model and integrate entities of different formal representations while maintaining high performance through aggressive indexing. In that respect, it is as much a knowledge management system suitable for AI applications as it is database for conventional enterprise systems. Key to such capability are its open-architecture and extremely general formal basis.

In this talk, I will present some of the more interesting aspects of the HyperGraphDB architecture and discuss some of the subtleties in balancing generality, practicality and efficiency in such an open-ended, yet highly organized memory model. I will compare it to other graph databases and put in the larger context of the recent NOSQL movement.

1:30-1:50 pm - RAC C/D - “Civic Hacking” - Luigi Montanez
How can software developers change cities, states, and countries for the better? Last year, we saw an explosion of interest around government transparency. The Open Government movement, spearheaded by open source developers, seeks to make government more accountable and responsible by turning open government data into citizen-focused, civic-minded applications. This talk will guide you through the Gov 2.0 landscape. You'll learn about the data sets and APIs available freely available for your use, the tools and skills you'll need to be a successful civic hacker, and you'll get a thorough overview of the current civic apps out there. Civic hacking will enhance your open source portfolio while making a difference in your community and country.

1:30-1:50 pm - RAC B - “You Already Use Closure (in Ruby)” - Justin Love
Closure, especially used through lambda, can be a powerful tool, which lies behind a lot of magic in Ruby (as well as other languages). Unfortunately, these tools are often misunderstood by those who didn’t cut their teeth on Lisp or Haskell. This talk will show you where closure is used in innocent looking every-day ruby constructs you already use, where it appears in popular Ruby libraries and frameworks, and how to take better advantage of closure in your own code. It will also be evident that Ruby makes some compromises in order to maintain it's many other features.

2:00-2:50 pm - Apollo 8 - “Perl 6 - The Apocalypse” - Matt Follett
Perl6 has been a long time coming with the original design dating back to 2000. This year marks the release of Rakudo Star. Rakudo Star is a useable, though not complete, implementation of Perl6 on top of the Parrot VM. This talk will cover some of the history Perl6 development and explain many of the features and advancements that Perl6 provides today. The talk will include such concepts as hyperoperators, Perl6 closures, optional typing, roles, and the new type system.

This talk will not assume any previous knowledge of Perl5 or Perl6.

2:00-2:50 pm - Apollo 11 - “Real World Modeling with MongoDB” - Steve Smith
Learn to break out of the habits of relational databases and model your data in a better, more meaningful way using MongoDB. Find out where the flexibility of MongoDB can let you rethink how you interact with your data, and how this flexibility makes your application cleaner, faster, and better.

2:00-2:50 pm - Pageant - “NoSQL at Twitter” - Kevin Weil
Non-relational data stores are growing in popularity, due both to the massive growth in the size of business datasets and to non-traditional access patterns caused by e.g. social graphs. Twitter faces both challenges, and so it is no surprise that we are making increasing use of NoSQL systems such as Hadoop, Cassandra, Redis, and our own open source social graph store, Flock. In this presentation, I will focus on how we use these systems at Twitter, with specific examples of where we ran into problems with a traditional MySQL-based architecture.

2:00-2:50 pm - RAC Studio - “Mobile HTML 5.0” - Michael Galpin
Are the smartphone wars wearing your out? When asked to choose between Objective-C and Java do you answer “None of the Above”? Do you think app stores are so 1995? Then there is good news for you and it’s called the mobile web. This isn’t about trying to port iFart to the browser, and it’s definitely not about tweaking an existing website so it doesn’t look awful on your mom’s iPhone. It is about writing full featured, engaging applications on the web. This talk is all about how to create killer web apps using HTML5, CSS3, as well as some other not-so-standard technologies available on a wide variety of popular smartphones. We’re talking about multi-threaded, high performance apps that can track your movement or even take pictures of whatever you think is interesting.

2:00-2:50 pm - RAC C/D - “Dude, That’s Some Strange UI Architecture” - Kyle Simpson
Whether you know it or not, every web application platform has UI Architecture, the stuff between the front-end and the back-end (aka, the "middle-end"). You know -- things like Templating, URL Routing, Data Validation/Formatting, Ajax, Compression/Optimization, etc. The problem is, you probably didn't realize it was there, and worse, you probably have no exposure to or control over those pieces.

The traditional MVC pattern is weak when it comes to this area, and leads to poorly maintainable and poorly performing web application architecture. We need a new, radically Strange pattern -- what I call CVC (Client-View-Controller) -- which is completely decoupled from both the front-end and back-end, uses JavaScript (both in the browser and on the server) to keep code DRY (reusable in both contexts), represents data *only* as JSON so it's completely portable, and allows the entire architecture to be tuned for maximum web performance optimization.

Web application frameworks/platforms are usually written by back-end developers, for back-end developers. But this ignores most of the important needs of the front-end, and also handcuffs front-end developers from having the control they need to do things properly. CVC+JavaScript puts the power of middle-end UI Architecture back in the hands of those best suited to control it: the front-end developers.

Come prepared to dive deep into UI Architecture and JavaScript (especially on the server), and get ready to be a middle-end architect!

2:00-2:50 pm - RAC B - “Java Provisioning in the Cloud” - Adrian Cole
This session will overview cloud provisioning tools written in Java and Clojure. First, we'll overview general provisioning concerns and how devops relates to the java landscape. We will show examples that work on several clouds via use of the jclouds framework. Examples will include using Whirr to manage Hadoop and Zookeeper clusters, Chef to manage all of your cloud node configuration, and Clojure for ad-hoc cloud administration tasks.

2:50-3:15 pm - Pageant - Snack Break
Join your fellow attendees for a quick snack at the Pageant. Snack and refreshments sponsored by Lisa Rokusek of Agent HR.

3:15-4:05 pm - Apollo 8 - "Unifying the Search Engine and NoSQL DBMS with a Universal Index" - Chris Biow
In contrast to single-function architectures, MarkLogic Server takes an unusual approach to collapsing the usual hierarchies of types of servers that make up a complete application, combining Search, a NoSQL DBMS, and an application server in a single kernel. The computational foundation for this hybrid is the Universal Index.

In this talk, we'll begin with the familiar text indexing data structures and algorithms that underlie search engine technologies. We'll extend that index to cover document structure and semantics, add scalar range indexing in one and two dimensions (including geospatial application), and then incorporate "reverse" indexing of queries. We will demonstrate a novel type of "matchmaking" query whose evaluation is based on a composition of forward and reverse index evaluation, in a true "strange loop" path through abstraction levels. Finally, we'll explore the means by which all of this indexing may efficiently run concurrently with querying, using MultiVersion Concurrency Control and Log-Structured Merge Trees, providing ACID transactions together with lock-free query evaluation, built-in sharding, terabyte-per-server scale-out, replication, and query distribution.

We will conclude with examples of production applications built on this architecture for geospatial service discovery at warriorgateway.org, social networking at bx.businessweek.com, and knowledge management on a US Air Force application.

3:15-4:05 pm - Apollo 11 - "Best Bugs, Best Practices, and Best Tools for Hibernate" - Patrycja Wegrzynowicz
Hibernate is the most popular object-relational mapping library for Java (and for most other JVM-based languages), providing a framework for mapping an object-oriented domain model to a traditional relational database. An application of hibernate to simple models is straightforward and almost effortless. However, in case of more complex models we usually run into some issues relating to performance or correctness.

We will show several flaws in the demo from 'Java Persistence with Hibernate' - CaveatEmptor (yes, it has several bugs including a serious locking-related issue!) and other open-source projects.

The hibernate-related flaws will be accompanied by alternative solutions and best practices, which help to improve performance and quality of both, database and object-oriented, models. We will explore patterns and practices mainly in the context of object-oriented model, specifically how to meet object-oriented principles, yet to ensure correctness and efficiency of hibernate mappings.

Additionally, we will present a free online tool that helps in automated discovery of concurrency-related issues with hibernate and database transactions. The tool uses static analysis to analyze the bytecode of any JVM-based application and to find bugs related to hibernate.

Upon completion of this presentation, attendees should better understand the potential hibernate issues along with patterns to use hibernate in a correct and elegant way. Moreover, attendees will learn how to automatically discover a certain class of hibernate-related bugs.

3:15-4:05 pm - Pageant - "Parallel Programming with Node.js" - Ryan Dahl
Node.js is inherently a single threaded, single process programming environment -- a Node script can run on at most one CPU/Core at a time. This has lead some to comment that it's an inappropriate technology for the our high density CPU reality. Node does, however, scale well to multiple CPUs with a simple actor layer. This talk will explain why simple event loop processes are good foundations for scalable, parallel network apps and techniques to use Node programs across multiple CPUs and operating systems.

3:15-4:05 pm - RAC Studio - "Clojure's Solutions to the Expression Problem" - Chris Houser
If you've done much work in any language with polymorphism, you've probably encountered the expression problem, whether you knew its name or not. Chances are even come up with a solution or two yourself. I'll define the expression problem, demonstrate some common solutions, then dig into how Clojure's multimethods and protocols each solve the problem while avoiding weaknesses of other solutions. Along the way you'll get a sense of how Clojure's approach to datatypes differs from classic object-oriented languages.

3:15-4:05 pm - RAC C/D - "Complexity Theory and Software Development" - Tim Berglund
Some systems are too large to be understood entirely by any one human mind, and can't readily be modeled using traditional mathematical tools. They are composed of a diverse array of individual components capable of interacting with each other and adapting to a changing environment. As systems, they produce behavior that differs in kind from the behavior of their components. Complexity Theory is an emerging discipline that seeks to describe such phenomena previously encountered in biology, sociology, economics, and other disciplines.

Beyond new ways of looking at ant colonies, fashion trends, and national economies, complexity theory promises powerful insights to software development. The Internet—perhaps the most valuable piece of computing infrastructure of the present day—may fit the description of a complex system. Large corporate organizations in which developers are employed have complex characteristics. Even the code base you're working on right now may share characteristics with a complex system. In this session, we'll explore

what makes a complex system, what advantages complexity has to offer us, and how to harness these benefits in the systems we build.

3:15-4:05 pm - RAC B - “Securing and Extending Puppet for World Domination” - Richard Crowley
Configuration management tools like Puppet and Chef are becoming essential to online business. They bring order and precision where there was once ~/bin/doit5. Surge's attendees may not have given their allegiance to a particular tool but I'm sure they're on-board with the idea of configuration management. In this session I'll share my experience integrating Puppet into the DevStructure service as part of our user-facing infrastructure. DevStructure offers development environments as a service and uses Puppet as the bridge between our web application and each of our users' servers.

Most DevStructure traffic can't be behind a firewall so security can't be subpar. I'll present the security concerns endemic to configuration management and operating over the Internet in general.

I'll then walk through our solutions. Some use common tools like iptables and stunnel; some come from Puppet; some are the result of architectural decisions.

We need our system configurations to react not only to code changes but data changes. I'll walk through Puppet's plugin API and some of its internals. We'll build an example plugin that alters the configuration as directed by a web service. Regardless of your choice of configuration management tool, reacting to data changes is a powerful way to scale your infrastructure.

4:15-5:15 pm - Pageant - “How To Think About Parallel Programming: Not!” - Guy Steele
Anyone remember the old days, when for good performance you had to worry carefully about which register should hold which variable, and when? Sometimes we still do this to get extremely high performance from critical inner loops, especially when using specialized processing hardware such as GPUs.

On the other hand, we have been able to write ever more complex and ever more capable software systems only by sacrificing such micromanagement and using general-purpose tools and abstractions for coding the bulk of our software. Along the way, we have discovered that code generated by automated tools often does a better job than hand-crafted code.

And we learn to code in such a way that the behavior of our code does not depend critically on the detailed optimization decisions that we have delegated to the tools. If we want to let a compiler's register allocator have the freedom to put variables in registers, we stop writing code that takes the address of a variable, as in the C expression &myvar . If we want to allow an automatic storage allocator to do its job, we must write code that works properly independently of where an object or array happens to have been allocated, and perhaps independently of whether the object or array happens to

be automatically relocated in the middle of a computation. Once we do this, we don't have to think about memory placement. Good programming language design can get us from the place where we must remember "don't use this difficult feature" to the place where it's not even on the radar screen because the language provides other, better ways to think and get things done. (Example: Java doesn't even have a way to take the address of a variable.)

Likewise, the best way to write code for multiple processors is not to have to think about multiple processors. We need to get to the point where we worry about the assignment of tasks to processors just about as much as we worry about the assignment of data to memory---which is to say, only for truly critical portions of the code---and for the most part leave such decisions to automated tools.

This will require further adjustments in our programming habits---adjustments that, we argue, in the end will make programs easier to understand and maintain as well as easier to run on parallel processors. The key is not to focus on a particular technology but on useful invariants. Here, as in the past, good programming language design can help to encourage good programming habits.

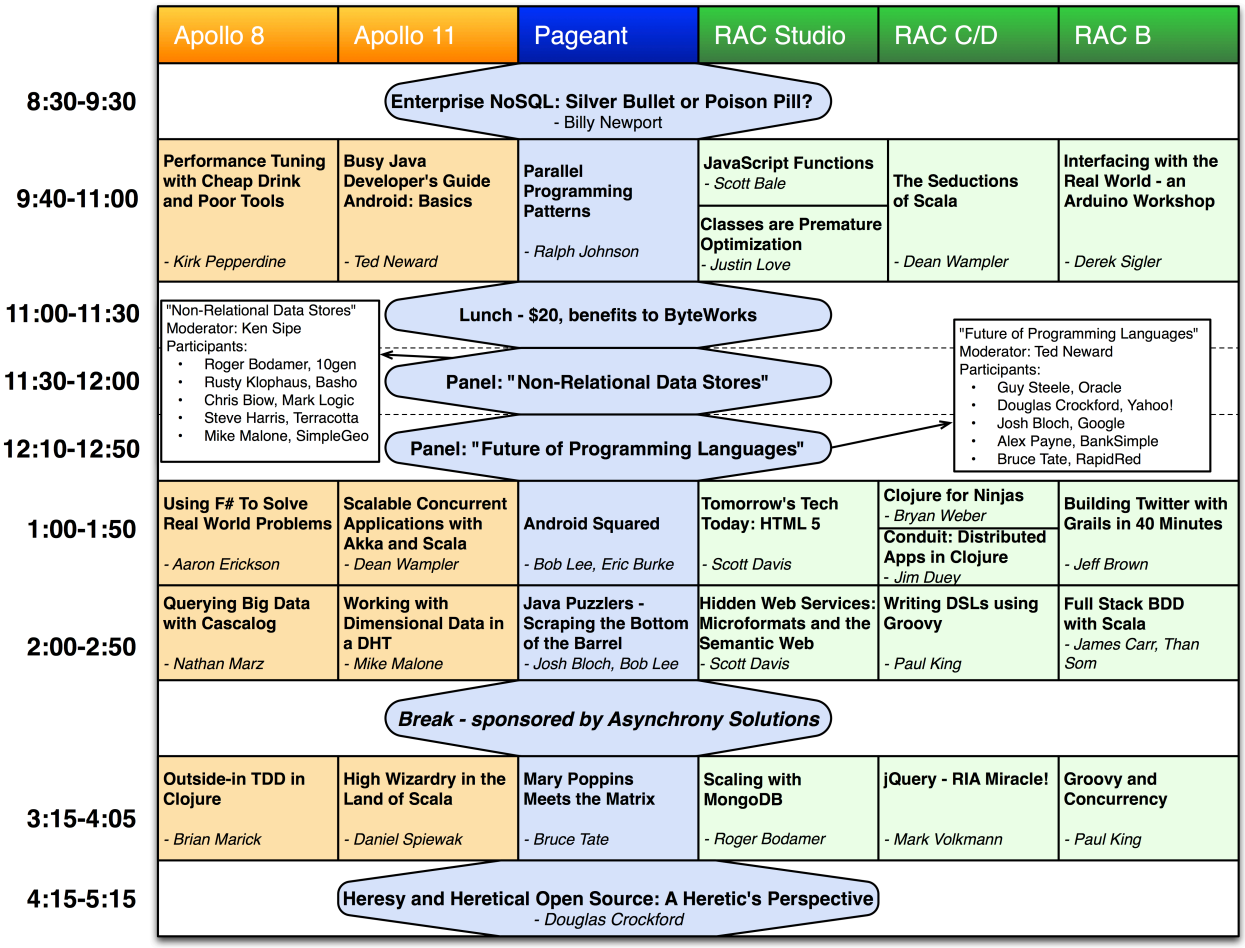
6:30-9:30 pm - Pageant - Party and Strange Passions
Strange Loop 2010 will feature the return of the enigmatic Strange Passions track. Strange Passions is an opportunity for conference attendees to present short (10 minute) talks on non-technical topics. Passionate developers tend to be passionate about other things as well and we had awesome talks in 2009 on subjects like neurons, astronomy, options trading, building houses in Mexico, and more. More details can be found here.

The Strange Loop party and drinks are sponsored by Mark Logic and Google.

Set 1 - 7:00 - 8:00
Sipping the Nectar of Life - Jacob Kenner
s/2 Years/10 Minutes/ of Photography - Michael Schade
Why technologists should care about politics - Jesse Phelps
Entering Without Breaking - Galen Collins

Set 2 - 8:15 - 9:15
A Brief Tour of Replicant Architecture - Aaron Westre
Why is there a lower-case 'a'? - Joel Neely
How to make friends and intoxicate people - Eleanor McHugh
The Science of Musical Counterpoint and Illusion - Daniel Spiewak

Friday, Oct. 15th



8:30-9:30 am - Pageant - “Enterprise NoSQL: Silver Bullet or Poison Pill?” - Billy Newport

NoSQL has become the latest darling technology. We will examine its roots, why it became popular in that context, and whether it can extend its reach into mainstream enterprise applications.

9:40-11 am - Apollo 8 - “Performance Tuning with Cheap Drink and Poor Tools” - Kirk Pepperdine

After a brief introduction to a methodology to performance tune Java applications, the audience will guide me through the steps needed to tune an application. The application models a number of performance problems that are common in real world applications. During the session, I will introduce a number of tools designed to expose causal code paths to each specific problem.

9:40-11 am - Apollo 11 - "Busy Java Developer's Guide to Android: Basics" - Ted Neward
Android is a new mobile development platform, based on the Java language and tool set, designed to allow developers to get up to speed writing mobile code on any of a number of handsets quickly. In this presentation, we'll go over the basic setup of the Android toolchain, how to deploy to a device, and basic constructs in the Android world. Attendees should be intermediate to advanced Java developers, as no time will be spent on Java basics, just the Android parts. Attendees are encouraged to bring laptops to the session (and your Android-based device, if you have one) to fill out code as we go, but the limited time frame means a focus on fast delivery of content and example code; have your fingers warmed up (and the SDK downloaded!) before you get here.

9:40-11 am - Pageant - "Parallel Programming Patterns: Data Parallelism" - Ralph Johnson
Parallel programming has a well deserved reputation for being complicated, but it doesn't have to be that way. There are several styles of parallel programming that can make programming easier. This talk will describe data parallel programming. I'll talk about data parallelism as a set of patterns, and describe the recent libraries from Intel and Microsoft for data parallel programming in C++, as well as libraries for doing it in Java and C#. I'll compare and contrast data parallelism with other styles of parallel programming (such as actor programming) so you can get an idea when you should use each style.

9:40-10:20 am - RAC Studio - "JavaScript Functions: The Good Parts - Idioms for Encapsulation and Inheritance" - Scott Bale
Pretty much anything interesting in JavaScript happens with functions. Because of their peculiar properties, a number of unusual idioms have emerged around using them for encapsulation or inheritance. You may be surprised to learn of the possibilities for large-scale, modular, extendable JavaScript. Along the way we'll cover such esoterica as function closures, object prototypes and the 'this' keyword. Many thanks to Douglas Crockford for his excellent book JavaScript: The Good Parts, to which this talk owes a heavy debt.

10:20-11 am - RAC Studio - "Classes are Premature Optimization" - Justin Love
Classes are premature optimization, forcing the programmer to freeze method implementations and often memory layouts during the design stage in order to make things easier for the compiler writer. Classes are also accidental complexity forcing the programmer to deal with rules and limits unrelated to the problem domain, and sometimes expend extra effort working around the class system. This talk will gaze into the soul of object oriented programming to see why classes might not always be beneficial (though they often are). We'll look at alternate visions from the hard core classlessness of Self to the modern renaissance of Javascript. Finally, there will be a review of patterns for the proper application classless patterns.

9:40-11 am - RAC C/D - "The Seductions of Scala" - Dean Wampler
Scala is a statically-typed, hybrid object-functional language for the JVM and .NET CLR. On the JVM, it has the potential to replace Java as the standard, general-purpose language. Even James Gosling has praised Scala.

This session demonstrates how Scala improves your productivity in several ways. Scala provides full support for functional programming, which is essential for writing robust concurrent applications. It fixes deficiencies in Java's and C#'s object models, enabling better modularity and *mixin composition*. It is a great platform for DSL creation. It gives you all this, yet Scala code interoperates seamlessly with your existing Java or .NET code, preserving your investment.

9:40-11 am - RAC B - "Interfacing with the Real World - an Arduino Workshop" - Derek Sigler
Through the embracing of open source hardware and software, Arduino units have become a favorite platform for software engineers to interface with the real world. Come see how gratifying it can be to see your code escape from the inside of a machine and function autonomously in the real world. We'll discuss current microcontrollers, provide some demonstrations and let you play with writing your own programs.

11-11:30 am - Pageant - Lunch to benefit Byteworks
Lunch will be available (but not required) for $20 with all profits going to benefit St. Louis non-profit organization ByteWorks. Lunch can be purchased ahead of time through the registration system OR some lunches may be purchased on-site (but availability will be limited).

ByteWorks is dedicated to advancing computer literacy to primarily at-risk and/or low-income children between the ages of 9 and 15 years old.

The children participate in a series of six classes, and are introduced to a range of computing topics including internet safety, word processing, and even programming.

If you purchase the lunch, you will also be entered in a raffle to win fabulous and potentially magical prizes. Currently donated items:

- $100 Best Buy gift card - Ardon Consulting / Angela Schnieders
- $100 Newegg gift card - Strategic Staffing Solutions
- Geek Grab Bags / noun / 3s / geak gerab baags / 1. Grab bags of techy toy goodness to excite one's inner geek.

11:30-12:00 pm - Pageant - Panel: "Non-Relational Data Stores"
This panel will be moderated by Ken Sipe and focus on the future of nosql and other non-relational data stores.

- Moderator: Ken Sipe - Perficient
- Rusty Klophaus - Basho (Riak)

- Roger Bodamer - 10gen (MongoDB)
- Chris Biow - Mark Logic (Mark Logic)
- Steve Harris - Terracotta (Terracotta)
- Mike Malone - SimpleGeo (Cassandra, others)

12:10-12:50 pm - Pageant - Panel: “Future of Programming Languages”
This panel will be moderated by Ted Neward and focus on the future of programming languages, language features, concurrency, and more.

- Moderator: Ted Neward
- Guy Steele - Oracle
- Alex Payne - BankSimple
- Josh Bloch - Google
- Bruce Tate - RapidRed
- Douglas Crockford - Yahoo!

1-1:50 pm - Apollo 8 - “Using F# to Solve Real World Problems” - Aaron Erickson
So you have seen a few sessions on F# or functional programming that introduce the concept. Maybe you are now excited about functional programming - or maybe you need a little more convincing to really see how it is used in the "real world". Come to this session, and see examples of useful F# applications that can potentially have real impact on humanity. In the session, we will cover the following topics:
- Using F# based Monte-Carlo simulation to avoid another global financial crisis
- Using F# based Natural Language Processing to solve the problem of "incivility on the Internet"

As we go through these examples, you will see how F# is particularly useful for solving these types of problems that involve simulation and categorization.

1-1:50 pm - Apollo 11 - “Scalable Concurrent Applications with Akka and Scala” - Dean Wampler
Akka (akkasource.org) is a Scala and Java framework for building highly scalable, fault-tolerant applications. Akka is inspired by Erlang's OTP framework. It uses *Actors* as the primary concurrency mechanism, with support for fault tolerance through transactional actors ("transactors"), actor supervision and error recovery, distributed processing, and transactional state management using Clojure-style persistent data structures, with or without "durable" persistence from NoSQL data stores like MongoDB, Redis, and Cassandra.

Akka also provides convenient integration APIs to 3rd-party libraries for web applications, REST, messaging queues, security, Spring and Guice dependency injection, etc.

Using a live example, this session demonstrates the capabilities of Akka and the productivity you can enjoy while using it.

1-1:50 pm - Pageant - “Android Squared” - Bob Lee, Eric Burke
Square enables users to accept card payments on Android devices. Square reads magnetic stripe data through the microphone port using a free reader and sends receipts via email or SMS. Square has been featured in the Android Market and at Google I/O.

Bob and Eric, the programmers behind Square, will demonstrate how magnetic stripe decoding works. They’ll describe Square's unique approach to Intent-based APIs used in the point-of-sale API. They’ll share tips for taming the activity stack and building device-independent user interfaces.

Finally, they’ll give a sneak peak into their upcoming open source Android library Retrofit. Retrofit provides utilities for dependency injection (using Google Guice), simple and fast persistence, REST communication and dialog management.

1-1:50 pm - RAC Studio - “Tomorrow’s Tech Today: HTML 5” - Scott Davis
As software engineers, we take comfort in the idea of concrete specifications. As web developers, our hearts are either broken (frequently!), or we recognize the W3C's role is a delicate balance of leading the browser developers in new and exciting directions while, in their own words, "paving over the cow paths" of existing, de facto standards.

HTML 5 offers dramatic new improvements for page organization, offering out-of-the-box support for elements like header, footer, nav, section, and article. HTML 5 adds native support for form features such as placeholder text, autocomplete, autofocus, and validation. Additionally, there are a host of new form elements available (email, url, number, range, date, and search) that gracefully degrade in "classic" web browsers -- IE, I'm looking at you!

In this talk, you won't be subjected to discussions about the features that will appear in some distant future release of a web browser. Instead, you'll see the HTML 5 features that are already being used by Google, Apple, and others. You'll see the features that are supported by today's browsers, ready for you to use right now.

1-1:20 pm - RAC C/D - “Clojure for Ninjas” - Bryan Weber
This talk is not only for Ninjas! One of the most frequently cited benefits of Clojure is being able to take advantage of the Java libraries and ecosystem. This talk will cover calling Java from Clojure and just as importantly calling Clojure from Java. Any ninjas that attend will come out of the session knowing how to use Clojure on a Java project without being detected... well, almost anyway.

1:30-1:50 pm - RAC C/D - “Conduit: Distributed Apps in Clojure” - Jim Duey
Writing applications that are distributed across multiple machines implies sending messages between the different logical portions of the code. The book "Enterprise Integration Patterns" went a long way towards documenting the various standard ways this message passing could be envisioned. Libraries like Apache Camel provide

concrete implementations of these ideas, but have limitations that come from the languages they are implemented in or target.

I introduce a library, called Conduit, that provides a clean conceptual framework for thinking about and composing distributed applications. EIP patterns can easily be constructed, reasoned about and connected using a small number of basic operators that hide the complexity of sending and receiving messages across various transports. The library can be extended easily to implement any transport that a user might require. An AMQP transport will be demonstrated and methods to extend to other transports explained. Establishing a foundation for thinking about distributed applications is the primary thrust of the talk so that developers will have a different perspective to approach such problems with.

You should attend if you want to stop doing distributed and multi-threaded apps the "hard way". This talk will show you a better way of thinking about and then implementing your designs.

1-1:50 pm - RAC B - “Building Twitter with Grails in 40 Minutes” - Jeff Brown
In this session Jeff Brown, core member of the Grails development team and a senior engineer at SpringSource, will demonstrate how the basics of Twitter can be built using Grails and JMS in only 40 minutes. A fast paced and code-driven presentation, Jeff will build a Twitter-like application from scratch using Grails and its rapid application development capability. By bringing together Spring, JMS and Java persistence techniques, Jeff will also provide advanced tips and techniques for constructing Grails applications that can be deployed on to the Java EE platform.

Attendees will learn:
- How to construct a basic Grails project with Spring-based domain objects
- How to incorporate messaging and persistence into your Grails application
- How to adapt basic configuration to suit the needs of your application

2-2:50 pm - Apollo 8 - “Querying Big Data with Cascalog” - Nathan Marz
Cascalog is a tool for querying data on Hadoop with Clojure in a concise, expressive, and highly readable manner. Cascalog combines two cutting edge technologies in Clojure and Hadoop and resurrects an old one in Datalog. Cascalog is high performance, flexible, and robust.

Most query languages, like SQL, Pig, and Hive, are custom languages -- and this leads to huge amounts of accidental complexity. Constructing queries dynamically by doing string manipulation is haphazard and leads to further complexity such as SQL injection attacks. The nature of Cascalog being a domain specific language in Clojure avoids these accidental complexities and allows a programmer to manipulate queries as first-class entities within the language. The Datalog syntax of Cascalog is simpler and more expressive than SQL-based languages.

Besides being a valuable tool in itself, Cascalog is a demonstration of the power of the Clojure programming language. Building an integrated query language like Cascalog is just not possible in any other language.

This talk will include a live demo of Cascalog.

Cascalog is hosted on Github at http://github.com/nathanmarz/cascalog and an introductory tutorial can be found at http://nathanmarz.com/blog/introducing-cascalog .

2-2:50 pm - Apollo 11 - "Working with Dimensional Data in a DHT" - Mike Malone
Recently a new class of database technologies has developed offering massively scalable distributed hash table functionality. Relative to more traditional relational database systems, these systems are simple to operate and capable of managing massive data sets. These characteristics come at a cost though: an impoverished query language that, in practice, can handle little more than exact-match lookups at scale.

This talk will explore the real world technical challenges we faced at SimpleGeo while building a web-scale spatial database on top of Apache Cassandra. Cassandra is a distributed database that falls into the broad category of second-generation systems described above. We chose Cassandra after carefully considering desirable database characteristics based on our prior experiences building large scale web applications. Cassandra offers operational simplicity, decentralized operations, no single points of failure, online load balancing and re-balancing, and linear horizontal scalability.

Unfortunately, Cassandra fell far short of providing the sort of sophisticated spatial queries we needed. We developed a short term solution that was good enough for most use cases, but far from optimal. Long term, our challenge was to bridge the gap without compromising any of the desirable qualities that led us to choose Cassandra in the first place.

The result is a robust general purpose mechanism for overlaying sophisticated data structures on top of distributed hash tables. By overlaying a spatial tree, for example, we're able to durably persist massive amounts of spatial data and service complex nearest-neighbor and multidimensional range queries across billions of rows fast enough for an online consumer facing application. We continue to improve and evolve the system, but we're eager to share what we've learned so far.

2-2:50 pm - Pageant - "Java Puzzlers - Scraping the Bottom of the Barrel" - Josh Bloch, Bob Lee
How can they do it? How can Josh Bloch and Bob Lee keep coming up with such great programming puzzlers year after year? They can't! In this, the eighth installment of the perennial crowd pleaser, Click and Hack the Type-It brothers are truly scraping the bottom of the barrel. But some of the dregs they come up with may still astonish, delight, and educate. Either that or you can have a good laugh at their expense. There's only one way to find out... Come to "Java Puzzlers —Scraping the Bottom of the Barrel." And come early—as always, overripe fruit will be given to the first fifty attendees.

2-2:50 pm - RAC Studio - “Hidden Web Services: Microformats and the Semantic Web” - Scott Davis
The hard line between web pages (pure presentation) and web services (pure data) is finally beginning to blur. Companies as varied as Best Buy, Twitter, Facebook, LinkedIn, Flickr, TripIt, O'Reilly, and even People magazine have decorated their web pages with hidden, semantic metadata. The results are impressive: a 30% increase in traffic for Best Buy, a 15% increase in click-through rate reported by Yahoo!, and dramatic Google PageRank improvements.

In this talk, we'll explore popular microformats such as hCard (the HTML equivalent of vCard) for contact information, hCalendar (the equivalent of iCalendar) for events, hAtom for syndication, and much more. We'll use Java and Groovy to tease out the hidden data in plain old HTML pages for use in everyday applications. You'll also see how Firefox and Safari plug-ins integrate the browser with your address book and your calendar in unprecedented ways.

This is not yet another staid, academic discussion of the future of the semantic web -- this is a pragmatic discussion of how the technology is being used right now to deliver real web services AND web pages at the same time.

2-2:50 pm - RAC C/D - “Writing DSLs Using Groovy” - Paul King
There have been many attempts to create languages which allow us to express our problems in higher-level languages: from COBOL to Object-Oriented languages, from Logic processing languages and SQL to rules engines. All have taken us forward in leaps and bounds but have failed to get very close to the language of the subject matter expert.

This talk examines how dynamic languages in general and Groovy in particular take us even further towards this goal. Groovy, is a popular and successful dynamic language for the JVM. It offers many features that allow you to create embedded Domain-Specific Languages: Closures, compile-time and run-time metaprogramming, operator overloading, named arguments, a more concise and expressive syntax and more.

2-2:50 pm - RAC B - “Full Stack BDD with Scala” - James Carr, Than Som
Come learn all about Behavior Driven Development and see it action to help define system behavior from the top level down to the unit level, describing the need for code to exist and then writing the code to meet that need. This will include a live demo of creating a complete feature from the outside and working our way in one piece at a time. Concepts demonstrated can be applied in other languages.

2:50-3:15 pm - Pageant - Snack Break
Join your fellow attendees for a quick snack at the Pageant. Snack and refreshments sponsored by Asynchrony Solutions.

3:15-4:05 pm - Apollo 8 - "Outside-in TDD in Clojure" - Brian Marick
It's sometimes said that design in functional languages is done in the interpreter. You try out different ideas, decide on appropriate representations and abstractions for the problem domain, then build a working program on top of them. Back in my Lisp days, that's the way I did things, more or less.

Since then, though, I've fallen in among disreputable companions---programmers in object-oriented languages, specifically those in the Agile school. They argue that you should start with the minimum abstractions required to get that first feature out. As you add new features, you should use your good taste (and refactoring guidelines) to refine and extend the abstractions in the code---abstractions that always match the real problem domain, not your prediction of what it will be.

In my own programming, that approach has worked well, so as I jumped on the Clojure bandwagon, I wanted to use it. In this talk, I will show how it works by building a program.

3:15-4:05 pm - Apollo 11 - "High Wizardry in the Land of Scala" - Daniel Spiewak
Scala is an intensely powerful language. One of the most obvious ways in which this manifests is the syntax, which is wonderfully amenable to internal DSLs and flexible APIs (not to mention endless reams of obfuscated sources and fanciful operators). However, despite the superficial flash of Scala's syntactic skin, its true power lies in the type system and in the language's deep semantic constructs.

This talk will dive into some of the more remote regions of the kingdom of Scala. Specifically, we will cover the following topics:

- Higher-Kinds (what they are and how they can be applied)
- Type-Level Encodings (*really* exploiting Scala's type system)
- Typeclasses (just like Haskell...except not)
- Delimited Continuations (and you thought kinds were confusing!)

Please note that this is an advanced talk targeted at the Scala practitioner who is already fairly comfortable with the language. With that said, we hope the talk will remain reasonably accessible to the Scala beginner - so long as they don't object to the presentation of odd and esoteric language features with disturbing enthusiasm.

3:15-4:05 pm - Pageant - "Mary Poppins Meets the Matrix" - Bruce Tate
Every foreign language you learn makes you a little smarter, and even shapes the way you think. In the Pragmatic Programmer, Dave Thomas and Andy Hunt say that a developer should learn a new programming language every year. In the upcoming book Seven Languages in Seven Weeks, Bruce takes this challenging advice to the extreme. The book, in progress, helps a developer solve a nontrivial problem in each of seven different programming languages, spanning four different programming paradigms. In this talk, Bruce will take a light hearted look through the evolution of programming

languages, paying special attention to the seven languages in his book, Ruby, Io, Prolog, Scala, Erlang, Clojure, and Haskell.

3:15-4:05 pm - RAC Studio - "Scaling with MongoDB" - Roger Bodamer
MongoDB is an open source, non-relational, document oriented database ideally suited for web development. MongoDB bridges the gap between key-value stores (which are fast and highly scalable) and traditional RDBMS systems (which provide rich queries and deep functionality).

MongoDB has been designed to scale horizontally via an auto-sharding architecture. Auto-sharding permits the development of large-scale data clusters that incorporate additional machines dynamically, automatically accomodate changes in load and data distribution, and ensure automated failover.

This talk will cover replication and replica sets, auto-sharding, and scaling use cases.

3:15-4:05 pm - RAC C/D - "jQuery- RIA Miracle!" - Mark Volkmann
Developing rich internet applications using HTML, CSS and JavaScript
is now a common exercise. While it can be done using only those
client-side technologies, JavaScript libraries greatly simplify
the task. They also shield developers from most browser differences.
The jQuery JavaScript library is a very popular option.

This talk demonstrates building a web application using jQuery.
It includes use of jQuery selectors to access DOM elements and attributes,
DOM traversal, CSS manipulation, mouse and keyboard event handling,
Ajax calls to server-side code using JSON to represent data,
dynamic HTML modification, and jQuery UI widgets.

3:15-4:05 pm - RAC B - "Groovy and Concurrency" - Paul King
This talk looks at using Groovy for multi-threaded, concurrent and grid computing. It covers everything from using processes, multiple threads, the concurrency libraries earmarked for Java 7, functional programming, actors including GPars, as well as map/reduce, grid and cloud computing frameworks. We'll look at leveraging Java techniques as well as Groovy specific approaches.

* Multiple Processes with Ant, Java and Groovy
* Multiple threads - Java and Groovy support
* The java.util.concurrent APIs, Fork/Join, Atomicity and more
* Useful Java libraries: Google collections and others
* Actor/Dataflow libraries: Jetlang, GPars
* Polyglot solutions with Scala and Clojure
* Grid computing and cloud solutions
* Testing multi-threaded programs

4:15-5:15 pm - Pageant - “Heresy and Heretical Open Source: A Heretic’s Perspective” - Douglas Crockford

This whirlwind tour looks at software from the unlikely premise that there is goodness in JavaScript, and that Intellectual Property Law has drifted broadly away from its original purpose, becoming a significant obstacle to the Progreſs of Science and useful Arts.

Crew

Dr. Strange Loop - Alex Miller
Event Coordinator - Nick Cowan

Volunteers
- Aaron Stevens
- Ben Lee
- Bill Zimmerly
- Brad Hogan
- Brad Hogenmiller
- Chad Burrus
- CJ Carey
- Don Ellis
- Douglas Rogers Jr.
- Erin Steinbruegge
- Gary Sheldon
- Gordon Sommers
- Jeremy Schlatter
- Jerry Stutte
- Lee Lammert
- Lisa Keller
- Mark Xie
- Michael Jette
- Paras Tiwari
- Robin Rath
- Stan Rosenthal
- Zac Duncan