HPTS – October 7 – 10, 2007
ABSTRACTS

Monday 8:45AM - 9:30AM Andrew Fikes Google

A Google Engineer's Toolkit: GFS, Bigtable and MapReduce

Google's engineers face a broad array of software architecture challenges. They have to build high-throughput, data-intensive systems for indexing the web, as well as low-latency, user-facing applications such as Orkut. To complicate things, applications are deployed in shared computing environments in geographically distributed datacenters. In this talk, I'll provide an overview of the Google computing environment, the architectures of GFS, Bigtable and MapReduce, and discuss how Google engineers leverage them to build different classes of applications.

Monday 9:30AM – 10:00AM Rene Aeberhard PayPal

??? -- <need title> -- ###

<### need abstract ###>

Speaker confirmed…

Monday 10:30AM – 11:00AM Marvin Theimer Amazon.Com

Amazon's EC2 and S3

A significant fraction of the workloads in the high-performance computing space is comprised of so-called "high throughput" jobs rather than "high capacity" jobs. These jobs run well on server farms composed of commodity hardware and do not require high-end "supercomputers". Although the cost of hardware for HPC has dropped dramatically, to the point where many ordinary users can afford to store substantial data sets and run substantial workloads over those data sets, the operational costs of HPC systems are still quite high -- especially when dynamic growth of capacity and the complexity of storing and efficiently accessing many terabytes or even petabytes of data have to be taken into account. Collaborative access to both computation and data adds a further dimension of complexity.

The design of Amazon's S3/EC2-based utility computing service has chosen to provide a very simple model of storage and computation: Users employ a web services-based interface to store and retrieve arbitrary-sized blocks of data in S3. Data is guaranteed to be stored reliably and with high availability. All operational aspects of storage, such as explicit backup, recovery from failures, and hierarchical/locality-oriented storage management facilities, are hidden from the user. To support computation, users can create, share, and run virtual machine images on dynamically allocated host resources in EC2. EC2-based computations have cheap, high-speed access to S3-based data.

Careful exploitation of economies of scale, combined with a simple model, have enabled the design and implementation to (a) provide an extremely cheap price point ($0.15 per GB/month for storage and $0.10 per CPU/hour) and (b) focus primarily on the key problems of scale, availability, locality, and operational management rather than on the provision of complex features that would make these problems more difficult to solve and that users may or may not need. The proposed talk will discuss the design philosophy behind S3 and EC2, the benefits we think we have gained, and some of the problems and challenges that we have encountered.

Monday 11:00AM – 11:30AM Vijay Raghavendra eBay

eBay's next generation application and presentation infrastructure

Large scale web transaction systems like eBay present unique and interesting challenges for scaling the application and presentation tier while increasing developer productivity and improving system performance. In this talk we will discuss the challenges, architecture and the tools that drive eBay's next generation application and presentation infrastructure.

Monday 11:30AM – 12:00PM Eric Baldeschwieler Yahoo

"Grid" Computing and large scale transaction systems (at Yahoo!)

At Yahoo! we deploy very high volume transaction systems. Our next generation systems use a hybrid of fairly traditional transactional techniques in combination with large scale batch processing techniques to scale cost effectively. In this talk we will discuss the architecture of such systems and some example deployments.

Monday 1:30PM - 2:00PM Stan Bailes Amazon.Com

SABLE -- Amazon.Com's Scalable Business Logic Environment

Huge scale, highly available systems can be built using simple components which implement a scalable data model: Business entities are defined to be the unit of data distribution and relations are implemented as messages between distributed entities.

A version of this model is presented which allows an implementation to guarantee eventual consistency, even in the face of replica failures or network partitions.

Monday 2:00PM – 2:30PM Matt Youill Betfair

Flywheel Transaction Processing System

Betfair (www.betfair.com), a UK based betting exchange, transacts the equivalent of over half the combined equity trading volume of every major stock exchange in the world. In common with other industries, particularly the financial services sector, Betfair is experiencing both an increase in transaction volumes and decrease in value per transaction. Due to the increasingly demanding nature of exchange betting, Betfair's strategic aim is to significantly raise its transaction rate capability, greatly reduce the marginal cost per transaction, provide operational simplicity, and enable a geographically distributed transaction engine.

The Flywheel project is a two year effort to produce a prototype exchange transaction engine capable of inexpensively processing in excess of 50,000 serial transactions per second with sub 250 ms latencies. Using a combination of techniques, some old and some new, Flywheel not only meets these targets but shows far greater theoretical limits.

Monday 3:00PM –3:30PM James Hamilton Microsoft

Writing Efficient to Operate Internet-Scale Services

A typical internet-scale software service might span 10,000 systems and support several million concurrent users. Failures will occur many times a day. The traditional model of relying on operator intervention for these situations does not scale. Live services must be designed to continue to serve customers through failures and to be able to recover from them without operator intervention. This requires a focus on application simplicity and automation. What may seem easy with 10 servers can be a nightmare on 1,000 to 10,000 and beyond.

High-scale live software services, compared with shrink-wrapped products, require different approaches to design, development, test, deployment, administration, and support. This presentation describes best practices across disciplines for the design and development of this new breed of system that is rapidly becoming a big part of Microsoft's future.

Monday 3:30PM – 4:00PM Jon Ruggiero Workday

Keep It Sparse, Keep It in Memory

The advent of 64bit JVMs and cheap memory are changing the dynamics of what is possible for large scale business applications. Jon will talk about the innovative and intensely object oriented toolset used to build, run and maintain a next generation ERP application at Workday.

Monday 4:00PM – 4:30PM Todd McKinnon Salesforce.Com

Plug Your Code in Here

Salesforce.com has successfully pioneered on-demand CRM. With the introduction of APEX Code, salesforce.com enters a new phase of what is possible on the salesforce.com platform. Todd will talk about the challenges and possibilities of allowing customer and partner specific code in a truly shared multi-tenant environment.

Monday 4:30PM – 5:00PM Mike Myer RightNow

RightNow Architecture

Mike Myer is CTO of RightNow. RightNow runs some of the most demanding high availability customer support applications in a multi-tenant environment. Mike will talk about the RightNow architecture and how it has grown over the years, including the use of open source databases, and the recent move away from a web based client to a Windows/web service based smart client.

Monday 7:30PM – 9:30PM Im-moderator – Shel Finkelstein

Poster Sessions

The Poster Sessions comprise 10-12 talks of 10 minutes each. Shel will accept volunteers for these presentations on Monday and will arbitrate which presentations are accepted and the order of their presentations. There are no rules about the content of the talks (e.g. knitting, skiing, hiking, transactional systems, service oriented architectures, or even database management). Presentations are strictly limited to 10 minutes in duration.

Existing presenters are allowed to do poster sessions on a ***different*** topic (provided it is entertaining).

You may sign up on site with Shel Finkelstein or cheat and negotiate a spot in advance by sending him an email at Shel.Finkelstein@SAP.com

Tuesday 8:30AM – 9:30AM Mike Stonebraker MIT

KEYNOTE: It's Time for a Complete Rewrite

In a collection of recent papers, we predicted the end of "one size fits all" as a commercial relational DBMS paradigm. These papers presented reasons and experimental evidence that showed that the major RDBMS vendors can be outperformed by 1-2 orders of magnitude by specialized engines in essentially any vertical market of significant size. These include:

- OLTP
- Data Warehouses
- Stream Processing
- Text
- Scientific Databases

Assuming that specialized engines dominate these markets over time, the current relational DBMSs are merely 25 year old legacy code lines that should be retired in favor of a collection of "from scratch" specialized engines. The DBMS vendors (and the research community) should start with a clean sheet of paper and design systems for tomorrow's requirements, not continue to push code lines and architectures designed for yesterday's needs.

This purpose of this talk is two-fold:

1) to show where the dramatic performance advantage comes from, and
2) to discuss some of the implications of the end of "one size fits all"

Tuesday 9:30AM – 10:00AM Vishal Sikka SAP

The Layers Above: Next Programming Models for Enterprise Software

The construction and deployment of enterprise applications is governed largely by the technology layers that surround these applications. And different layers in turn often have different rates of change associated with them. For example, while Moore's law largely governs CPU evolution, multi-core/many-core designs will take a while for apps to adopt (~7+ years per Dave Patterson). There is a major new programming language roughly every 10 years (per Alan Kay) and several minor ones in the interim. New UI technologies emerge roughly twice a year. Just as we were getting done with Ajax, MSFT announced Silverlight and Adobe AIR. Not to mention the vast changes in price/performance economics of memory vs disk, relative network performance, the new economics of large scale data centers and power consumption, web based ontology generation, and many others.

And yet enterprise applications often outlast all these changes. Our own SAP R/3 was introduced more than a decade ago, and is just now getting past the peak of its lifecycle. Legacy systems still dominate IT landscapes and long-term maintenance devolves into islands of (expensive) expertise.

Abstractions help simplify design-times to some extent, but often reflect in creating their own associated run-time islands. Each specialization brings about its own sets of containers, littering IT landscapes with islands of computing, each one bringing its own TCO and lifecycle mgmt issues to the table.

We ask the question, then, of what the next great programming model for enterprise apps ought to be? What are its characteristics if it is to support construction and management of applications that outlast and exploit changes to the underlying technical dynamics? Can we have benefits of abstraction at design-time and at the same time have optimizations at run-time? In other words, how can we cross some of these divides that we have often presumed to be unbridgeable? We present some of these characteristics, and some early thoughts on what it might take to build such an enterprise programming model.

Tuesday 10:30AM – 11:00AM Bruce Lindsay IBM

Database Indices: Who Needs 'Em? All Queries Are Text!

Stored data is very heterogeneous. Valuable stored information increasingly includes diverse elements such as XML, plain text, presentation graphics, sparse attribute sets (e.g. product catalogue), as well as object specific (structured) meta-data. Extracting value from such diverse collections is not well supported by traditional database indexes over selected table columns. This is especially true for semi- and un-structured data elements for which deep analysis is needed extract valuable "meaning' and structure.

The text processing community has begun to address value extraction by incorporating the results of sophisticated text analytic algorithms into the text indexes. That's the good news. The bad news is that extracting value via text analytic tools is computationally complex and the index data may be much bigger than the raw data. However, text analysis and multi-dimensional indexing are both well suited to scale-out clustering on commodity hardware.

I suggest that many information storage and retrieval applications can be well supported by a (central) transactional storage system that is asynchronously coupled to scaled-out analytic and indexing systems. All application specific query services are implemented via a single (scaled-out) index and exploit general (e.g. words, meta-data) and specific (e.g. relationships, classified terms) analysis results stored in the index.

Tuesday 11:00AM – 11:30AM Margo Seltzer Oracle/Harvard

Erlang plus BDB: Disrupting the Conventional Web Wisdom

The conventional wisdom says that you implement a web site with web servers, application servers, and a relational database backend. However, this solution is neither scalable nor reliable. Next generation high-performance, highly reliable web sites are going to be implemented using Erlang and Berkeley DB.

Erlang is a language, developed by Ericsson, used in mission-critical telecom applications requiring nine 9's -- yes, that's 31 ms/year of downtime. Erlang applications achieve non-stop operation, live upgrades, and failure resilience, by taking the historic UNIX model of small, simple processes to the extreme. Applications consist of thousands or tens of thousands of isolated, light-weight processes, communicating through message-passing. The magic is that Erlang process creation and IPC are fast (a few microseconds), but there is no shared memory, no mutexes, and all Erlang data are immutable.

The only thing missing from Erlang is large-scale persistent storage. Enter Berkeley DB. Berkeley DB is an embedded database library providing ACID, replication, and automatic failover. The Berkeley DB Driver in the Erlang Driver Toolkit unites the scalable, fault-resilient, and highly concurrent Erlang environment with the high-performance, highly- available, flexible storage of Berkeley DB. This combination enables the development of enormous, high-performance web sites with the reliability profile of mission-critical telecom switches.

Tuesday 11:30AM – 12:00PM Ron Avnur Mark Logic

MarkLogic Server: Not Your Grandmother's XML Database

MarkLogic Server is a database management system designed for XML, 100s of terabytes of XML, 100s of terabytes of semi-structured XML. Building this system was quite an adventure:

- Transactions are important, but what traditional approaches make sense?
- Content bases (databases of semi-structured content) contain XML that varies from having well-defined schemas to having no schema whatsoever, yet people want to load and query it all.
- Moreover, people want to query over structure, values, and text together. What can we learn from the search engine world to build a new high-performance query processing engine for these queries?

Ron will talk about the experiences of building this commercially-proven, enterprise-grade DBMS from scratch: What turned out to be an implementation exercise? What did we engineer? And when did we find ourselves doing new science?

Tuesday 1:30PM – 2:00PM Shanku Niyogi Microsoft

Silverlight: Bringing .NET and RIA Everywhere

Silverlight is the recently announced browser-based and cross-platform version of .NET. Using Silverlight, applications can offer a RIA experience using the .NET language of their choice with the same framework, execution model, and high-performance runtime as offered in .NET on Windows. Web 2.0 applications demand a rich client experience and Silverlight offers LINQ (Language INtegrated Queries) as a common foundation for working with data. Silverlight offers a safe execution environment in which the browser-based application is sand-boxed and only able to update limited resources on the client machine.

This talk will describe the evolution of .NET into Silverlight, describe the approach to LINQ integration with client-side UI, and address the safety provided by the sandbox.

Tuesday 2:00PM – 2:30PM Charlton Barreto Adobe

Enterprise and Collaborative RIAs with AIR and LiveCycle ES

Discover how the Adobe Integrated Runtime (AIR) and LiveCycle ES (LCES) provide a powerful set of capabilities to the hands of developers of enterprise-class RIAs. This talk covers how they combine to enable high-volume, real-time and collaborative RIA applications, and how the embodied SOA programming model increases productivity, enhances performance, and simplifies maintenance.

Tuesday 2:30PM – 3:00PM Scott Dietzen Zimbra

Zimbra: Lessons Learned Crafting a 100+KLOC Ajax Application

The Zimbra Collaboration Suite may currently represent a high-water mark in terms of Ajax application size and complexity. In the talk, we will focus on the novel aspects of the Zimbra client architecture and the server that supports it, as well as discuss the associated challenges in RIA integration/mash-ups, performance, hardening, and security.

Tuesday 3:30PM – 4:00PM Aaron Brashears Second Life

Living without Transactions

Second Life is a 3-D virtual world entirely built and owned by its Residents. Since opening to the public in 2003, it has grown explosively and today has nearly 9 million registered users. At any particular time, between 35,000 and 50,000 unique people are simultaneously collaboratively producing and consuming content in a world which is managed completely on the grid and streamed to the clients.

With approximately 4000 hosts in two separate co-location facilities simulating well over 13,000 distinct regions of the world, overcoming network topology and reliability issues is paramount for attaining agreement of the state of the world.

This talk will discuss the current internal Second Life service infrastructure, what challenges we currently face, and our plan for addressing these challenges. We do not have a system capable of processing previously unheard of quantities of transactions, but we do have a simple plan for distributing agreement across an infinite number of hosts as long as each host can process a modest number of ACID transactions.

Tuesday 4:00PM – 4:30PM Adrian Cockcroft NetFilx

Millicomputing: the Coolest CPUs and the Flashiest Storage

Power consumption is a hot topic, however using CPU designs from the world of battery powered devices, and flash memory based storage we can make cool systems. A Millicomputer is defined as a computer that uses less than one watt, so its power is specified in milliWatts. Enterprise Millicomputer clusters provide large numbers of small computing units at an aggregate cost, performance and power level that redefines the limits of what is possible. Initial Millicomputer systems are being constructed as open source hardware by their end users. This presentation describes the architecture, capacity characteristics, and applications of Enterprise Millicomputers. The future is Open Hardware by the Milliwatt!

Tuesday 4:30PM – 5:00PM TBD

TBD

Tuesday 7:30PM – 9:30PM

Tribute to Jim Gray

As we all know, our dear friend, Jim Gray, has been missing at sea since January. We will gather over beer and wine and offer our tribute to Jim. In early, 2006, Jim was interviewed for a series called "Behind the Code" which explores the life and works of a notable software engineer. We will share this hour long video and then tell our own "Jim stories" of which there will be many!

Wednesday 8:30AM – 10:00AM Dave Patterson (UB Berkeley) and Burton Smith (Microsoft)

The Hardware Sea Change

The microprocessor industry is now at a crossroads due to hitting the limit of the power that a single integrated circuit can dissipate. To continue the pattern of increasing performance, semiconductor companies have been forced to replace the single large power-inefficient processor with several smaller power-efficient processors operating in parallel. This shift toward increasing parallelism is not a triumphant stride forward based on breakthroughs in novel software and architectures for parallelism; instead, this plunge into parallelism is actually a retreat from even greater challenges that thwart efficient silicon implementation of traditional single processor architectures. The new "Moore's Law" is now to double the number of processors per chip with each new technology generation, with individual processors going no faster.

In addition to this revolution in microprocessors, we'll talk about the technology trends in DRAM, Flash, Disk, networking, and data centers.

Dave will cover the inevitable changes in hardware. Burton will follow with the implications for software.

Wednesday 10:30AM – 11:00AM Pat Helland Microsoft

The Irresistible Forces Meet the ~~Im~~-Movable Objects

There are a number of economic forces being unleashed on our industry:
- The Looming Arrival of Many-Core Processor Chips
- The Cost Changes with Low-End Datacenters (no battery backup, no operators in attendance)
- Flash Is Disk; Disk Is Tape → Most Machines Will Use Flash as their Disk Durable Storage; Spinning Disks will Replace Tape for Archive; Impact on Power and on Management
- Pervasive Intermittent Connectivity (every client comes and goes with their connection to the network… there's no such thing as "online")
- Devices, phones, iPods, RFID, Sensors, Face Recognition, Speech Recognition… What the Heck Is the Client?
- With So Many Devices, Where the Heck Is My State?

These forces mean we need to program for:
- Parallelism running across the Many-Core processors
- Services and State in the Cloud AND Spread across the Devices/Clients I Use
  → What Is the True Answer with So Many Opinions?
- Data and State in the Cloud Running with Multiple Replicas (on Crappy Datacenters) Ensuring Highly-Available Access to you Service
- Browser (whatever that means) Access to My Data and Computational Resources
- Configuration Organized by Business Function and Individual More than Machine

Today, we have notions of components and composeability of "applications". Tomorrow, it will be difficult to tell where one application ends and the next one begins. We will need to focus on large-scale "objects" (i.e. think of components/services/new-code) which will move in their behavior. First, it is essential to say that the changes will necessarily involve:

- Integrate: the new code must be allowed to be a part of the existing code and have clear mechanism by which part of an existing application comprises new code, and
- Entice: the new code must offer so much business value that customers will want to create parts of their systems using the newer model of components and, over time, move more and more of their functionality forward.

How can we create applications that are approachable to implementers, composeable in their deployments, and responsive to these economic and technical forces bearing down on us?

Wednesday	*11:00AM – 11:30AM*	*Mike Carey*	*BEA*

SOA What? The Line between Data and Processes is Blurry!

In the world of SOA today, data is too often overlooked or forgotten. This speaker will start by raising a few questions regarding the role and positioning of data in SOA-land. I will then explain the positioning of data in BEA's approach to SOA and what BEA is doing to simplify the problems of service-enabling and integrating SOA data. Finally, I will raise some questions that our community ought to be thinking about in terms of how SOA applications (a.k.a. composite applications) are designed and built and where the line is (and/or should be) between "data" and "process".

Wednesday	*11:30AM – 12:00PM*	*Al Spector*	*Independent*

Lessons for High Performance Services

Transaction processing as a field of engineering and science has a 50-year long record: Thus, due consideration of the engineering lessons of systems like Saber and the academic work, say, on atomicity or replication, should now be yielding high quality, non-speculative design principles. Perhaps some elements would be marked as understood and codified; some as explored, but not yet fully understood; and some as emerging and still unexplored. But, there would at least be the beginnings of an unambiguous, engineering backbone to the field.

But, the reality is that we have many too many unpooled experiences and too many proposed techniques, structures, and algorithms on which no critical consensus has been reached. For example, I have my own opinions on the value of our teams' work on transaction management (done 10-20 years ago at Carnegie Mellon and Transarc), but I've yet to write (or see) a broad critical analysis that attempts to put the relevant topics conclusively to bed. There are many more examples like this.

My motivation for desiring to begin this classification is a feeling that our field (and computer science as a whole) does not sufficiently and clearly document what we know. We don't really address the reduction large aspects of software/systems project to a more traditional type of engineering. Due to the breadth of my recent management experiences, I recognize I lack qualifications to contribute too much to this effort, but indeed no one can see the whole elephant. Perhaps, the right approach is to come up with shared publication mechanisms (and needed structural elements) that can leverage the great successes of social networking technologies to harness the collective wisdom of the community.

I will present this critique, illustrate it with categorized examples (some of which may generate debate), and propose some conceptual structures that could lead to a better codification of the science and engineering principles of transaction processing.