Sitaram Iyer

Principal Engineer
(current projects: Livegraph, Alexandria)

My snippets | okrs

Significant Project Accomplishments at Google

I joined Google on Aug 26, 2003. Over most of the past 8 years, I led indexing-team to build two generations of Google's web search indexing system. I found myself in 2 month-long code yellows, 2 spells (8 months + 3 months) in warrooms, and sustained, multiple-year spells of intense development. Fun times were had by all.

Livegraph (May 2013 - present)

Livegraph is an online system meant to replace the end-to-end knowledge graph building pipeline. http://go/livegraph

Tech Lead: Alexandria (Oct 2009 - Oct 2011)

1. **Overall:** Alexandria is Google's incremental indexing system for web search. I led the team to build and launch it in 2009. Since then I have worked on making it deliver on its promise and potential. This involved:

- a lot of non-obvious and technically hard details that we figured out,
- a bunch of lateral/creative thinking that led to radically simple and powerful designs,
- iterations (on average our components are on their third rewrite), and
- developing the team's skill set to cover key areas needed to keep the project moving forward.

I established our team's primary focus as improving index quality, which helped channel our efforts during the post-launch slack. The ranking code yellow happened just then; it was about improving Google search quality given recent competition. I pushed indexing team headlong into it, and jointly coordinated the effort to debug 1300 queries. Based on that, I carved out 40 indexing projects, and drove them really hard for 2 quarters with ours and a few adjacent teams. As a result index quality went up, and most low-hanging bugs in the entire crawl/indexing pipeline got fixed.

Following that, as the pace started easing off, I restored momentum and focus by getting the team more organized. There were a lot of important but un-sexy projects that were not happening, and the team was getting increasingly silo-ed. I pushed people to think about these projects, and started tracking them and running weekly meetings (with the help of a project manager). We finished ~30 of these projects in each of two quarters, and have ~75 currently in progress. This resulted in two of the strongest "peace-time" quarters that we have ever seen, with hopefully more to come.

2. **Dup Clustering:** Dup clustering and canonical url selection is possibly the most complex component of Alexandria. Web pages are clustered based on dup rules, redirects, and checksums, and a canonical url is chosen based on quality signals and webmaster guidance.

   1) The web is a complex place, and this system mirrors the complexity. Hundreds of broken cases came up during the ranking code yellow, and a handful get reported every week, which the Dups TL and I debug every day, discuss, and attempt to convert to CLs.

   - These can be high profile: e.g., once vatican.va got hijacked by pedofilo.com. After various short term fixes, we worked with the pageranker team on a year-long effort to implement the "perfect" solution.
   - There was a rash of Appspot proxies hijacking good sites, for which I contributed fixes.
   - Wikipedia pages often get vandalized in dramatic ways (e.g. [sergey rapist]). We worked around it in Dups, then I worked with a team in Kirkland for the long-term solution.

   2) Forwarding dups: This system also selects a few dups to collect signals from, and thus has a direct effect on ranking. There were always a few occasional but serious problems, and ongoing pressure from Search Quality. The usual answers were expensive and imperfect, and we even ran an analysis project that came to nothing. I came up with an idea to nail the problem by keeping the top 5 dups with navboost. Simple change, thorough eval, and haven't seen a single problem since.

   3) We stabilize dup clusters and canonicals. I proposed two of the four strategies currently in use, and worked with someone to implement them.

   4) I tried to push the envelope by creating 3 projects around second-guessing webmasters with misconfigured websites, but we get blamed anyway.

   5) Webmirror-dups is a sister system that computes dup rules, and sometimes gets it wrong. We dealt with the consequences many times, once in a very major way that I took the lead to recover from. I have worked with them, and put pressure on them for about a year, in these ways:

   - I helped them design an online validation system
   - I co-designed and co-wrote an offline eval system

- I wrote a tool to isolate the top 5000 broken dup rules, which also serves as a tracking metric
- I suggested many improvements to their logic.

6) I conceptualized dups V4, which is to become the next and much simpler implementation of this system.

3. **Anchors:** Alexandria inverts links on the web, forwards them across dups, down-samples them, and provides them for ranking purposes. I helped the team chalk out a series of simple, concrete, and inexpensive solutions for:

   - penalizing bad anchors "at the last minute" to reduce spam
   - removing anchors from dead hosts
   - applying these penalties in Teragoogle index tiers
   - keeping anchor sources fresh by crawling important ones often
   - pushing anchor quality signals in weeks instead of months
   - sampling anchors to keep fewer but more diverse ones

4. **Crawl Interface:** During the ranking code yellow we found a large class of problems ultimately traced back to our failure to crawl a page. Also, about a year ago, we would see a few cases a week due to the same set of problems.

   - The Dups TL and I were the main channels from query losses to crawl scheduler bug fixes or features, and this process eventually converged to our satisfaction.
   - I advocated out-of-band, low-latency crawls from all parts of indexing to resolve any material doubts about the actual state of the web.
   - I drove a month-long case study of zazzle.com to see how well we are doing. The results were impressively positive.
   - I actively participate in bi-weekly meetings with the crawl teams where we iron out crawl problems and cross-system issues.
   - I redesigned the crawl scheduler interface from managed clusters to repeated one-offs which simplifies it. (yet to be implemented).

   There are still huge inefficiencies (for which there is a new project to revamp the entire workflow, which I'm participating in), but I'm glad to say that glaring quality bugs due to crawl failures are no longer a concern.

5. **Misc components:**

   - I wrote the initial version of the "Gatekeeper", which blocks serious problems from reaching the index, and has saved us a lot of bother.
   - I wrote the initial version of "delete-docs", the application-level garbage collector.
   - I wrote the "unprocessed url monitor" which surfaces a new set of bugs every week. In response, I co-wrote a change to ensure forward progress for urls stuck in a loop for weeks/months.
   - I refactored the entire codebase to simplify and make it harder to introduce bugs.

6. **APIs for indexing users:** The old indexing system was a black box. I wanted Alexandria to be "open" in safe ways to other teams at Google, and make it bidirectional by sending notifications to teams that need filtered access.

   - I came up with the idea of an arbitrarily extensible "Alexandria proxy service" with a read/write hook into our system. I helped docjoin-access team develop it, and now it has many users.
   - On a request from Moonshine and ContentAds to index private content, I came up a RunPipeline interface to convert a fetchlog to a docjoin without writing anything down.
   - I worked with Social teams (e.g. +1 service) to be notified of dup clustering changes.

- I notify Webspam folks about homepages that move between IP address as an input to suspected spam.
- I came up with new way of extracting "swap" docjoins for Search Quality experiments, and helped implement it. This method is far less painful and more correct than before.

7. **Integration with other systems:** One of the (stretch) goals of Alexandria was to be _the_ indexing system for all of Google. While this is too idealistic in the extreme, I have worked with many teams to come up with the right customized solution for them.

   - Freshdocs piggybacks off of Alexandria data, but remains our insurance policy. More recently, I had the idea of pushing "interesting transitions" to them for force reindexing. I'm also in discussions about sharing more components.
   - Ocean is integrating with Alexandria at the output stage, and I'm advising them on that.
   - Ramsey-team is about Smartphone optimized indexing. I have been guiding and helping them integrate.
   - I regularly help Image and Video teams get higher quality and more efficient.

8. **Release process, testing, backup, etc.:** Cos asked me to take the lead on beating the release process into shape, with solid testing behind it. Paul Haahr asked me to cut weekly releases, so that Quality engineers don't have to worry about coordination. We achieved both, and with minimum drama.

   - I closely mentored the testing team, setting goals, suggesting ideas, optimizing their pipelines, and writing some code.
   - I wrote and maintain the script that cuts releases and automates the process.
   - I play benevolent dictator for the production schema.

   Related point, I am working with Matt Snelham to figure out the details for Alexandria backup and disaster recovery.

9. **Performance:** I help with Alexandria performance debugging, and I wrote tools, improved the dashboard, etc.

   - I worked with the Dups TL to optimize incremental dup clustering, which was _hard_.
   - I came up with a simple fix for anchor table hotspots, by limiting concurrency.
   - I proposed two ideas (a year apart) to make the input side of Alexandria more robust, which led to corresponding rewrites.
   - While system throughput was okay, "goodput" was on a decline. I got a couple folks from Index Metrics team to tally crawl/indexing numbers end-to-end, and slice and dice them. This is an ongoing investigation.

- ## Tech Lead: Alexandria (up to launch) (Feb 2007 - Oct 2009)

Blurb: I led a 30-strong team in developing Alexandria, the next generation indexing system for web search, from concept to launch.

I led a team of 30 engineers over 3 years, and built Alexandria, a.k.a. Google's new incremental indexing system. Now it is a full-featured system almost on par with Google on quality, and scale that puts it at the largest and fastest data crunching system in Google by far (50 PB, 15M bigtable ops/sec). We launched a preview (called "Caffeine") and are due to roll out to all users starting in a month. I have been actively involved in every part of the process.

Public blog post: http://googlewebmastercentral.blogspot.com/2009/08/help-test-some-next-generation.html

Design and implementation

- I initially structured Alexandria application code into components, fleshed out the consistency model, and put the data model / schema together.
- I worked closely with each component-level team to build their piece. Over 3 years I wrote over 600 CLs, or 50% more than the next contributor on the team. I reviewed over 1000 CLs, or twice as much as the next reviewer.
- I conceptualized and wrote the two system tests that are being used today: a micro-level system test and a regression test.
- I redesigned and rewrote the converter component as a state machine, for greater clarity and correctness.
- The "dups" component (most complex part of the system) is launching version 3. I proposed the v2 and v3 designs, and convinced the team. I wrote a lot of v2, and helped write v3. I helped write their design doc.
- I closely mentored the Alexandria index selection team, helping them solve tough problems on maintaining index composition in an incremental system. They are now close to evaluating then launching.
- I helped the Alexandria signals and anchors teams solve difficult technical problems and iterate on their designs.
- I worked with several external teams to design their system for Alexandria, or to interface with it.
  - librarian for teragoogle
  - image search (moosedog)
  - docjoins for search quality
  - freshdocs/instant indexing
  - custom search (prose)
  - mobile search (in progress)
  - webspam
  - pageranker
  - crawl scheduler.
- I wrote a side-by-sides tool called "deathmatch". With eval-team and index-metrics team, I got the quality triage process started.
- After a big data loss incident, with some others I went through the entire application codebase and made it robust to corruption.

Performance

Alexandria is the most challenging performance problem any of us have worked with. There are some in our team who took the lead in getting through the layers. My focus was the application and its performance.

- I spend a good chunk of my time debugging performance. I have made a lot of optimizations to the code, e.g. jointly designing a priority scheme, throttling certain operations, improving locality, avoiding hotspots, etc.
- People tend to introduce performance bugs often (such as hotspots, poor cache use, excess I/Os). I keep Alexandria code on a tight leash, and frequently teach people how to write efficient application code.
- I wrote the initial versions of two tools, "performance table" and "progress monitor", that we use today.
- I inverted the execution model once, restructured the application, and got a 2x speedup. I have ideas for a few more such big improvements.

Leadership stuff

- I defined Alexandria OKRs every quarter for 3 years, and helped decide what each quarter should be about. I feel we have kept the project on track the whole time, despite being ambitious.

- I wrote the Alexandria design doc, and maintain the project wiki and PDB.
- I presented Alexandria on a dozen occasions: L&S reviews, quality team, quality leads, Bill's ext-staff, etc.
- Matt Cutts and I authored the webmaster blog post about Caffeine.
- On a daily basis, for 3 years, I assigned / delegated / prioritized / pushed back efforts with team members.
- I teach simple and efficient design, and clear and practical thinking, a LOT. With ~100 engineers rapidly contributing code, this is necessary.
- I am effectively the release engineer: every 2-3 weeks I cut releases, after rapidly reviewing ~200 CLs. I wrote our code management tools. I am also the point person for our actual release engineer, who helps out.
- I am the point person for our team's project manager. We meet and organize tasks for others to do.
- I met SRE several times to plan and negotiate machines, constraints, and services, for the serving colo.
- I called a bug triage week, a quality code yellow week, and a code deletion week, for the team.
- Every week I meet with the alexandria test eng team, and guide them towards building useful tools for us.
- I fought a handful of major fires (the kind with post mortems) and dozens of minor ones, and pulled together groups of people as necessary to tackle the crisis of the occasion.
- I met with several prospective entrants to the team to discuss the project.

Joint Tech Lead: The Indexing Pipeline (Nov 2004 - Feb 2007)

In Nov 05, Kekoa and I independently initiated the design and development the current Indexing system. I have helped take it from concept to production, and have contibuted to all aspects of it. I have personally invented, fleshed out, and generally implemented many strangely named concepts in current usage:

- repository queues,
- multi-tier collapses,
- garbage collection via drain,
- siblingmap,
- serving-elsewhere filter,
- name probes,
- queueflagparser,
- tracers,
- janitor,
- data browser,
- dataflow analyzer.

The team grew from 2 to 25 in the last almost 2 years, and I helped bring a number of newcomers up to speed. I did a ridiculous number of code reviews.

In the past six months, these were my main contributions:

- The biggest chunk of my time went into collaborating with other (sometimes remote) teams as we helped them modify the indexing system to incorporate their stuff. These were teams that did
    1. **Premium Content** (Dale Neal and Anurag Acharya),
    2. **Frames** (Jeff Cox in New York),
    3. **History** (Douwe Osinga in Zurich),
    4. **Blimpie++** (Hao Wu and Jun Xu),
    5. **Images team** (Chuck Rosenberg),
    6. **PerfectCrawl** (Max Ibel, Frederik Schaffalitzky, and Ralph Keller, in Zurich).

Often I have been the main Indexing contact person for these teams, and I helped them out in terms of education, advice, code reviews, and actual coding. There was a spell of two months when I did nothing but code reviews and discussions with these teams. It was immensely rewarding, but it was also an exorbitant amount of time designing, teaching, and code reviewing. I believe I was partly instrumental in Jeff Cox's intention to work with our team going forward, and while at Zurich, I am spending a lot of time working with Frederik.

- I gave a Tech Talk in Google Zurich on "The Indexing System", covering its architecture and rationale, Index Selection, and a demo of our tools.

- With Eisar, David, and the rest of the team, I have fought innumerable fires at all hours of the day or night, and have written random code in all parts of the system.
- Jointly with David Ziegler, I have been working on restructuring our storage from 2 to 12 GFS cells, and accessing everything through namespaces. This has been a long and tedious task, and I'm still on it.
- I wrote a slightly unusual regression test to validate the mechanics of the indexing pipeline.
- I am writing a document called "The Mechanics of the Indexing Pipeline", to explain most of what's "black magic" about it.
- I have done some performance work with Borg-team (latency measurements) and Mapreduce-team (finding stragglers, raising alerts).
- Over several meetings with WebTable-Team, we designed incremental duplicate elimination for the next generation indexing system.

In 2005 and late 2004, these were some of my specific contributions:

- the pipeline's operating framework, along with the controlling mechanisms,
- migration of the entire pipeline from bootstrap to continuously rolling mode,
- focus on Pipeline performance,
- building tools like the Pipeline Data Browser, Pipeline Dataflow Analyzer, GFS Space Auditor, etc.
- designing/building the urlmap, global-repmap, global-namemap, etc.,
- initial version of refresh selection with name selection using indyranks,
- various aspects of index selection, including several filters, working with Eric Jackson, and one major revamp,
- implementing drift evaluation (not yet deployed),
- status pages and invariants checks,
- building combined summaries, and liaison with the crawl diagnostics team,
- liaison between our pipeline and most user groups (image crawl, chinese crawl, perfect crawl, indian web),
- fixes to the content converter's redirects processing,
- some aspects of refreshing Blimpie using this pipeline,

• Tech Lead: Indexing Service Tools (Apr 2006 - Feb 2007)

I am also the Tech Lead for "Indexing Service Tools", where our team of three (Mike Treaster, Andy Hochhaus, and myself) design infrastructure for use by the whole company, to manage or analyze large production jobs and bodies of data (both of which the indexing system has). We have built four tools in the last six months.

- **Dataflow Analyzer:** described separately.
- **Job Controller:** Generalizing on the work of numerous job controlling utilities throughout the company, I have been working with Andy Hochhaus to design and build what we hope to be a simple, powerful, and super-general dependency management tool that Indexing and other teams with production batch jobs may use. After a lot of work we have it used in production by the indexing system, but we are a couple months away from a general release.

- **Janitor:** I invented the idea of this garbage collection tool that watches and deletes GFS files based on a complex set of policies and interdependencies. I built a prototype late last year, handed it off to Mike Treaster, and continued mentoring/advising him on the design. He has since reimplemented it into a brilliantly useful and dependable tool, an essential piece to the automation of the system. I code-reviewed it, and we plan to release it to the company in a few weeks.
- **Auditor:** I wrote this utility to be the counterpart of the Janitor. While the Janitor manages known space, the Auditor reports what are the big fish that have not been accounted for by the Janitor. It has been useful in space crises, which happen every few days.

The Dataflow Analyzer (Feb 2006 - Feb 2007)

I conceptualized and fully implemented this tool that does set operations on very large sets, in order to answer a variety of high-level questions about the dynamics of the indexing system. I spent a lot of time on this, and it has been described as "my baby". For example, it compares discovery rates between crawlers, measures index churn, tracks perfectcrawl urls through our pipeline, explains the sizes of some indices, etc. To get the expressiveness I envisioned for such a generic tool, after building and scrapping a GUI, I decided to design a custom programming language. Using the tool, I have uncovered many interesting/disturbing facts about the pipeline, challenged our early assumptions, and worked closely with teams like PerfectCrawl, Blimpie++, and Premium Content to analyze their data passing through our system.

Index Selection (Tech lead, May 2004 - Jan 2005, Team member, Jan 2005 - Feb 2007)

In the first half of 2006, I have worked with Sanjay, Aaron, Lars, Frederik, and Jun, on approxdups, hostlimits, host growth, segment skew management, index selection for Perfectcrawl, index selection for Blimpie++, and index selection evaluation.

Old news:

Given a set of documents, this project is about selecting the 4B documents of highest quality for indexing in Base. In the 10B webmirror crawl world, index selection is quality critical; now we are selecting from Base+Blimpie and trying to improve Base quality.

This project originated from notions that Fay, Daniel, Abhay, Anurag and I maintained about pagerank being a poor indicator of quality, and moreover, Base is only loosely selected by pagerank. Other signals like navclient, spam signals, and query-dependent document selection hold promise. So I pulled Oscar and Piaw into a team, and from May to mid-Aug, we systematically explored the space of selection criteria. We concluded this experimental phase with confidence in query-driven index selection (using either query results or a query-document matching approach that Oscar is developing), with a fraction of the index reserved for pagerank type signals to cover for query drift.

- Description of current state and long term visions.
- Wrote up experiment reports (chronological): pr/indyrank/random scorers and language/contenttype breakup, uniqueified logs, new eval metric, random500, quality evals and analyses of a serving 1.78B index.
- Presented index selection at three engineering reviews: 4/21, 7/14, 8/31, and at a quality meeting: 9/1.
- Held meetings every other week to discuss these and plan our next steps with Sanjay, Gomes, Amit, Anurag and Bill, and posted meeting notes on the project page.

We formulated and followed an evaluation plan, of selecting using these criteria for a 2B base index, running various evals and analyses on it, and refining the selection criteria. We served them using my idea of a mixer filter that searches all of Base+Blimpie and filters by looking up an sstable_server of docids that we select for this index. This gives a turnaround time of half an hour for selection+eval rather than half a month.

If all goes well, by mid/end October we should replace Base by an index of higher quality, greater coverage, fewer useless pages, and less spam.

- ## Blimpie++ Code Yellow (Oct 22 - Nov 10, 2004)

Launched a 20B Blimpie++ index, bringing Google to 30B (or in Sergey's words, 3 times as big as Yahoo's 20B index). I worked on the indexing pipeline part of it, and was operator for many nights.

- ## PhD (May 2005, part of Nov 2005)

After 2.5 years of being with the company I finished my PhD. I took three weeks off (as comp leave for leading the *last* (Helium) code yellow), worked like crazy and defended it. In Nov 2005 I worked some ridiculous 18 to 22 hours a day for 3-4 weeks, doing Google work in the day and PhD thesis at night. It was mind-numbing, but I cannot describe how glad I am to see it done.

- ## I'm Feeling Lost (with intern Mario Tapia) (Jul 2005 - Sep 2005)

I mentored an intern (Mario Tapia), and worked with him in my 20% time --- partly through a Code Yellow that I was critically part of. (Mario now has an offer from Google and is very likely joining). He did something very interesting -- he built a system to find users who are lost in the middle of long search sessions because they are being unable to phrase their query properly, and to such users, recommend the results of others who have been lost in similar long sessions but found something at the end of it. Despite my own lack of time, I mentored him effectively and ensured that he had a fulfilling and productive internship.

- ## Code Yellow Lead: Helium: 10B index (Oct 22 - Nov 10, 2004)

Worked with a team of about 30 wonderful people to double ("blow up") the index size to 10B documents. We quietly announced this on the eve of MSN Search's launch, and got some beautiful press coverage in addition to making Microsoft look stupid.

This project lasted three consuming weeks. I worked literally 20 hours (and slept 4 hours) every day including weekends, but it was thrilling.

- ## Blimpie (Aug 2003 - Aug 2005)

I used to handle the indexing and various other aspects of Blimpie, and have indexed and pushed three cycles. These days Blimpie has grown into a testbed for the New Crawl/Indexing Pipeline, and as such, I am automating Blimpie and transferring its operations to index-admin and SRE.

In earlier Blimpie cycles, I did most of the menial chores myself, which took up WAY too much of my time. In particular, I have:

- written mapsyncer.cc: code that integrates Bart with Blimpie crawl and is required for the latter,
- indexed and pushed three cycles of Blimpie segments,
- worked with Debra about Blimpie machine allocation,
- managed Blimpie serving machines (setup, kernels, manual replacement, etc.),
- administered the Blimpie GFS and GWQ in HE,
- folded in Deepcrawl after obtaining pageranks for these documents,
- always run the segindexer from the main branch, which has proved to be a useful canary for the production segindexer,
- coached Ken Ashcraft in Blimpie resharding and push.

- **Segment Indexer (Feb 2003 - Apr 2004, Consultant from May 2004 - Jun 2005)**

I have been part of the segindexing team, done index admin firefighting for a month, and have generally helped their team with bugfixes and late night debugging sessions. I have made 20 changes to segindexer. The major ones are:

- fixed set_machines(2*machines) which made indexers on QJ run rampant,
- wrote additional-collapsed-doclogs to add Deepcrawl to Blimpie,
- implemented external duptables for adding Base duptables to Blimpie,
- wrote a duptable-filter utility,
- did a big and nasty SETI downintegrate for Hyung-Jin,
- added iscrawled() to urlfptourlmap for Christophe's crawl coverage work.
- tied Alexis' linkmapextractor into segindexer,
- modified docjoin to perform seti pagerank sortedmap lookups,
- allow --linklogs=default,file,file...
- a few segindexer and indexmaster bugfixes.

- **Warroom (SETI) Pageranker (20% project during Apr 2004 - Sep 2004)**

I worked with Alexis and Corey, and later with Yonatan and Abhay, to integrate warroom pageranks into the segindexer. This is work I did partly while I was in the warroom. Corey wrote a design doc.

Initially I implemented it in a mode where the segindexer waited for the SETI pageranker to converge. Then we deployed some of this code, did experiments to discover that a day old pageranks are good enough with fallback to crawl pageranks, and I rewrote my code to be nonblocking for the segindexer, with lots of failure modes. I made five changes to the pageranker code, including adding distance to the crawl_checkpoint_prmaster and fixing an old normalization constant mess. I wrote a CGI that plots pageranker distance graphs, and a script that monitors pagerank misses. This effort finished and got deployed in sep04.

- **Docservers (Feb - Mar 2004)**

Worked on improving docserver utilization, with a pre-parsed repository format, resurrecting Daniel's idea of docserver rescoring, with loadtests and such. Turned out moot due to Mustang :(

- # Mapreduce, IndexMaster, etc.

I have submitted some changes to Mapreduce: internal event counters, setrlimit fix, making /map and /reduce tasks clickable, and HandleSerialFailures. One more in the queue, to specify mappers on the command line.

I made a number of improvements to the IndexMaster, and have done some giant code reviews for Yonatan.

- # Miscellaneous

After discussions with Sanjay Ghemawat, I have put together a list of some major problems with running systems at Google, and have taken the trouble to fix them or see that they got fixed.

Significant Non-Project Accomplishments

- Filed a patent on Joint money accounts in a social network such as Orkut.
- After procrastinating for nearly two years, I took three weeks off, went back to school, and finished my Ph.D. in May and Nov 2005.
- Submitted a paper to SOSP 2005 based on my Ph.D. thesis work.
- Mentored noogler Fred Quintana and intern Mario Tapia. Tech-mentored Mike Treaster.
- Referred ~20 candidates.

Education

- Ph.D. in Computer Science, Rice University, Nov 2005 (while working for Google)
  Dissertation: Application-assisted physical memory management
  Advisor: Prof. Peter Druschel

- M.S. in Computer Science, Rice University, Jan 2001
  Thesis: Anticipatory disk scheduling
  Advisor: Prof. Peter Druschel

- B.Tech. in Computer Science and Engineering, IIT Bombay, May 1998
  Thesis: Xority: A measure of separability of training sets for neural network size estimation
  Advisor: Prof. Pushpak Bhattacharyya

Papers

Peer-reviewed publications:

1. Practical, transparent operating system support for superpages. *Juan Navarro, Sitaram Iyer, Peter Druschel, Alan Cox*. Published in the Symposium on Operating Systems Design and Implementation (OSDI), Dec 2002, Boston, MA. [pdf]

2. Squirrel: A decentralized peer-to-peer web cache. *Sitaram Iyer, Ant Rowstron, Peter Druschel*. Published in the Symposium on the Principles of Distributed Computing (PODC), July 2002, Monterey, CA. [pdf]

3. Anticipatory scheduling: A disk scheduling framework to overcome deceptive idleness in synchronous I/O. *Sitaram Iyer, Peter Druschel*. Published in the Symposium on Operating Systems Principles (SOSP), Sep 2001, Chateau Lake Louise, Banff, Canada. [pdf] -- **Now part of Linux 2.6!**

4. Xority: A measure of separability of training sets to estimate hidden layer size in neural networks. *Sitaram Iyer, Pushpak Bhattacharyya*. Published in the Intl. Conference of Knowledge Based Computer Systems (KBCS), Dec 1998, Bombay, India. [pdf]

In preparation:

1. Application-assisted physical memory management. *Sitaram Iyer, Juan Navarro, Peter Druschel*. [pdf]

2. A resource management framework for predictable quality of service in web servers. *Mohit Aron, Sitaram Iyer, Peter Druschel*. [pdf]

Recent Talks

1. Some internal talks listed on http://go/alexandria

2. Tech Talk: The Indexing System. *20 July 2006, Google Zurich*.

3. Squirrel: A peer-to-peer web cache. *PODC '02, 23 Jul 2002, Monterey, CA*. [pdf]

4. Anticipatory disk scheduling.*SOSP '02, 23 Oct 2001, Chateau Lake Louise, Banff, CA*. [html] [pdf]

5. Invited talk at IIT Bombay *5 Jun 2002, Bombay, India*. [ppt] [html].

Research Experience

1. Graduate Student, Rice University, Aug 1998 - Aug 2003. Focussed on Operating Systems research, especially on problems in resource management.

   - Application-assisted physical memory management:

     Leveraged the ability of many modern software applications to trade off memory consumption for performance. Designed an OS facility that notifies elastic applications about the severity of memory pressure, allows them to adapt to changing memory availability, and enables a powerful mechanism of user control over memory allocations.

   - Transparent superpage support in operating systems:

     Collaborated with Juan Navarro in developing a practical solution to the decade-old problem of superpage support in operating systems. Achieved sustained speedups often exceeding 30% with our FreeBSD/Alpha implementation.

   - Anticipatory disk scheduling:

Identified a long-standing problem in OS disk subsystems, where synchronous I/O induces a state of `deceptive idleness'. Solved it with a novel disk scheduling paradigm that deliberately delays disk requests before service. Implemented this scheduler in FreeBSD, and experimented with file servers, web servers and databases to observe 30-60% performance improvements in many cases. Results of this work are being implemented and tested by independent parties for inclusion in Linux-2.6.

- Resource management for server QoS:

  Collaborated with Mohit Aron in designing a system that enables web and proxy server operators to ensure a probabilistic QoS level for web services co-hosted on servers. Applied methods such as admission control, feedback-based scheduling for multiple resource classes, and resource usage monitoring and readjustment.

- Contributed to Dr. Peter Druschel's Pastry research project on peer-to-peer systems, by designing the Squirrel web cache, designing and coding parts of the FreePastry software base, reviewing research papers, and setting up experimental hardware and software.

- Participated in systems-related course projects and minor research projects:

  - memory management in the IO-Lite unified buffering system;
  - enabling architecture heterogeneity in the TreadMarks DSM;
  - multipath routing for network QoS;
  - parallel profiler for programs written in pthreads;
  - DSR performance under statistical channel fading models;
  - transparently enabling SSL for conventional applications.

2. Summer Intern, Microsoft Research Lab, Cambridge, England, July-Sep 2001 *Project:* Squirrel: A decentralized peer-to-peer web cache
   Mentor: Dr. Antony Rowstron

   Designed, implemented, experimented with, and authored a paper on a peer-to-peer web cache named Squirrel (published in PODC 2002). Thus contributed to the ongoing peer-to-peer applications effort by exploiting a novel design space tradeoff.

3. Visitor, Massachusetts Institute of Technology, Cambridge, MA, Feb-May 2001 Conducted part of the anticipatory scheduling research at MIT, while accompanying my advisor during his sabbatical.

4. Summer Intern, Bhabha Atomic Research Center, Bombay, India, Aug-Sep 1997 *Project:* Parallelization of neural networks for handwritten character recognition
   Mentor: Dr. S. M. Mahajan

   Devised and evaluated two novel parallelization schemes on the ANUPAM parallel processing system, for distributing neural nets that implement modern handwriting recognition techniques.

5. Undergraduate, Indian Institute of Technology, Bombay, Aug 1994-May 1998 Took up several, mostly extra-curricular, mostly OS-related projects:
   - implemented a basic microkernel for the i386 architecture;
   - wrote a Linux file system to perform package management;
   - explored new parallel disk I/O techniques;
   - network-booted DOS on diskless workstations.

Honors

- Rice University Fellowship for graduate study
- Ranked 49th among 100,000 candidates over India in the IIT entrance exam (9th in the Western Zone)

- Indian National Talent Merit Scholarship awardee, 16th in state
- High School Merit Scholarship awardee

Professional Service

Refereed 17 papers for the following conferences, symposia and journals:

USENIX Security 1999, Sigmetrics 2000, OSDI 2000, USENIX 2001, SOSP 2001, Infocom 2002, ISCA 2002, OSDI 2002, PODC 2003, TOCS Journal (2004).

Other Contributions

- Created and maintain a popular poetry anthology web site (www.minstrels.org) for the past four years; the associated poem-a-day mailing list has nearly 2000 subscribers.
- Wrote and released open-source software applications (SVNCviewer, slash2mail, cluster-tools, cvs-exp, etc.)
- Created a script-driven documentation repository at IIT Bombay, which is actively being used and maintained by current students.
- Developed an accounts management web site to keep track of joint finances among groups of friends; it currently hosts 30 users.
- Maintain the technical report archive in the Rice CS department.
- Administered server clusters in the Rice CS systems group, and assisted many colleagues with many technical difficulties.

Personal

- Interests: Rock climbing, hiking, science fiction.
- Personal webpage at Rice University, and my resume from that time: (text), (pdf).