

Andrew's T-shirts
Andrew's Links
Videos
Panoramas
About Me
Pictures
Public Key
RSS Feed
E-mail Me

Site

2005-10-22 16:58:07

Some changes are coming to my website, but first a little history. Originally my website was just intended to explain my t-shirts to classmates. I didn't even have a domain yet, so you had to go to 24.20.153.67 on some funky port to even get there. A few months later I bought a domain name to use for e-mail, and then a few weeks after that I switched my website to my domain. I started adding more content and I used this blog area (I called it 'news' back then) to announce the new content. After a while I started posting boring life stories. I always liked the idea of a journal, but I could never stay up to date. I type a lot faster than I can write, and the website was an easy place to store the information, so ended up just using this as a public journal. Now, however, I find that my past e-mails and IM logs are doing a good job archiving my life and day to day activities. Plus, now that I have Spotlight, I can easily search through past IMs to find what I'm looking for (although, Spotlight could use snippets). Anyway, since my website was made from scratch, I have to write all the features I want. At one point I decided I needed RSS, so I wrote a small RSS script. That created another problem, because then I needed a way to link to posts. I was lazy, so on the huge page of past posts, I just added a little anchor text to each entry and linked to that. When someone would click to read more in their RSS reader, they would have to load a huge page which would then jump to their post. It wasn't ideal, but it worked for the time. Now, my quest for more readers and in-links has spurred me to make more site improvements. Tuesday, in preparation for the BigTable talk, I updated my script to allow links to individual blog posts. I had to remember how my big, uncommented index script worked, but that was easier than I anticipated. I also changed the links in my RSS feed to point to the new page. It worked. I've gotten quite a few links to my overview of the talk (including from the official [Google Reader blog](#)). I also got rid of the music albums that were on the right and the random quotes at the bottom of the page. Now, the page should be exactly the same every time someone visits... unless I have actually changed content. I figured I should give Google and other search engines a little hand by removing it (although, I'm sure they have ways to differentiate between content and excess stuff like that). Also, with less random data, their BigTable database should compress slightly better. I recently read about [the most common weblog mistakes](#) and so far I've fixed two of the five I was violating (that list prompted me to add the new About Me page). A few more violations will probably be fixed in time. I have a few more updates planned, but some of the ideas require a bigger overhaul of the site (which is probably a good idea). Keep an eye out for more small, evolutionary changes in the near future.

[Permanent Link](#)

2005-10-21 16:18:15

Okay, I take back some of the bad stuff I've said about Microsoft recently. Yesterday I attended a tech talk about Visual Studio 2005. Unfortunately, only about one quarter of the talk was about Visual Studio; the rest was the story of one guy's journey through Microsoft. Come on guys, if you are going to advertise a talk about Visual Studio, you need to spend more time talking about Visual Studio (however, I was lax from the free pizza, so I didn't protest at the time). They demonstrated the extensible build system, which didn't interest me that much (I don't have much use for it right now), but I was impressed by the unit testing system. After running a test, it shows not only whether the test passed, but how much of the code was touched during execution. You can then click on the unit test and it'll show you the code it didn't test. I haven't used a large number of developer environments (only Xcode, Visual Studio, and Eclipse), so I don't know whether it has been implemented before, but I thought it was really neat. At the Google open house in November 2004, I believe I overheard a Microsoft employee talking about the tools they were developing to automate testing (especially for games). I wonder if this is related to that. As for Office 12, the only features that have piqued my interest are the visualizations. For example, if you have a column of values, you can set them to display little bar graphs which change colors based on their values. I don't often use spreadsheets, but I imagine this would come in handy for scanning large amounts of data for peculiarities. The problem isn't that Microsoft isn't developing cool new features, they are, it is that they don't promote the stuff I find interesting. I hear about these cool Office and Visual Studio features through random blogs and tech talks, not through official Microsoft channels (not that I follow those closely). Microsoft makes a big push for tablet computers or media center PCs, but I don't really care. I want to know about little features that will make my life better or easier (and no, new toolbars and a new theme do not make my life better). Also, in addition to the free hard copy of Visual Studio 2003 Academic (an electronic version is available on the CSE website for students). They are also going to send me a copy of Visual Studio 2005 when it is released. Now I'm actually interested in testing out C#.

[Permanent Link](#)

2005-10-18 18:35:35

Today Jeff Dean gave a talk at the University of Washington about BigTable - their system for storing large amounts of data in a semi-structured manner. I was unable to find much info about BitTable on the internet, so I decided to take notes and write about it myself.

First an overview. BigTable has been in development since early 2004 and has been in active use for about eight months (about February 2005). There are currently around 100 cells for services such as Print, Search History, Maps, and Orkut. Following Google's philosophy, BigTable was an in-house development designed to run on commodity hardware. BigTable allows Google to have a very small incremental cost for new services and expanded computing power (they don't have to buy a license for every machine, for example). BigTable is built atop their other services, specifically GFS, Scheduler, Lock Service, and MapReduce.

Each table is a multi-dimensional sparse map. The table consists of rows and columns, and each cell has a time version. There can be multiple copies of each cell with different times, so they can keep track of changes over time. In his examples, the rows were URLs and the columns had names such as "contents:" (which would store the file data) or "language:" (which would contain a string such as "EN").

In order to make each manage the huge tables, the tables are split at row boundaries and saved as tablets. Tablets are each around 100-200 MB and each machine stores about 100 of them (they are stored in GFS). This setup allows fine grain load balancing (if one tablet is receiving lots of queries, it can shed other tablets or move the busy tablet to another machine) and fast rebuilding (when a machine goes down, other machines take one tablet from the downed machine, so 100 machines get new tablet, but the load on each machine to pick up the new tablet is fairly small).

Tablets are stored on systems as immutable SSTables and a tail of logs (one log per machine). When system memory is filled, it compacts some tablets. He went kind of fast through this, so I didn't have time to write everything down, but here is the overview: There are minor and major compactions. Minor compactions involve only a few tablets, while major ones involve the whole system. Major compactions can reclaim hard disk space. The location of the tablets are actually stored in special BigTable cells. The lookup is a three-level system. The clients get a pointer to the META0 tablet (there is only one). This tablet is heavily used, and so one machine usually ends up shedding all its other tablets to support the load. The META0 tablet keeps track of all the META1 tablets. These tables contain the location of the actual tablet being looked up. There is no big bottleneck in the system, because they make heavy use of pre-fetching and caching.

Back to columns. Columns are in the form of "family:optional_qualifier". In his example, the row "www.cnn.com" might have the columns "contents:" with the HTML of the page, "anchor:cnn.com/news" with the anchor text of that link ("CNN Homepage"), and "anchor:stanford.edu" with that anchor text ("CNN"). Columns have type information. Columns families can have attributes/rules that apply to their cells, such as "keep n time entries" or "keep entries less than n days old". When tablets are rebuilt, these rules are applied to get rid of any expired entries. Because of the design of the system, columns are easy to create (and are created implicitly), while column families are heavy to create (since you specify things like type and attributes). In order to optimize access, column families can be split into locality groups. Locality groups cause the columns to be split into different SSTables (or tablets?). This increases performance because small, frequently accessed columns can be stored in a different spot than the large, infrequent columns.

All the tablets on one machine share a log; otherwise, one million tablets in a cluster would result in way too many files opened for writing (there seems to be a discrepancy here, he said 100 tablets per machine and 1000 machines, but that doesn't equal one million tablets). New log chunks are created every so often (like 64 MB, which would correspond with the size of GFS chunks). When a machine goes down, the master redistributes its log chunks to other machines to process (and these machines store the processed results locally). The machines that pick up the tablets then query the master for the location of the processed results (to update their recently acquired tablet) and then go directly to the machine for their data.

There is a lot of redundant data in their system (especially through time), so they make heavy use of compression. He went kind of fast and I only followed part of it, so I'm just going to give an overview. Their compression looks for similar values along the rows, columns, and times. They use variations of BMDiff and Zippy. BMDiff gives them high write speeds (~100MB/s) and even faster read speeds (~1000MB/s). Zippy is similar to LZW. It doesn't compress as highly as LZW or gzip, but it is much faster. He gave an example of a web crawl they compressed with the system. The crawl contained 2.1B pages and the rows were named in the following form: "com.cnn.www/index.html:http". The size of the uncompressed web pages was 45.1 TB and the compressed size was 4.2 TB, yielding a compressed size of only 9.2%. The links data compressed to 13.9% and the anchors data compressed to 12.7% the original size.

They have their eye on the future with some features under consideration. 1. Expressive data manipulation, including having scripts sent to clients to modify data. 2. Multi-row transaction support. 3. General performance for larger cells. 4. BigTable as a service. It sounds like each service (such as Maps or Search History) have their own cluster running BigTable. They are considering running a Google-wide BigTable system, but that would require fairly splitting resources and compute time, etc.

The talk was very interesting and it contained lots new information I had never heard before (I haven't seen a BigTable paper yet). This information is accurate to the best of my knowledge, but if you see a mistake, please e-mail me. I wonder what the next Google talk will be about... I can't wait.

[Permanent Link](#)

2005-10-14 13:26:17

As I'm sure you have heard, Apple held a special event on Wednesday where they announced the iPod capable of playing video and a new video section of the iTunes Music Store. As an Apple, iPod, and iTunes user (as well as HD junky), I have some thoughts on this. First, what the hell where they thinking upping the version number again? iTunes 6? This is one of the worst examples of version number inflation ever. Wasn't that one of the complaints of AOL?

Anyway, I think this is a big step forward. Steve had already mentioned the \$1.99 price point of music videos when he said they were going to sell TV shows. I was very afraid that they would price the TV shows too high. Fortunately, TV shows have the same \$1.99 price. At this price you could buy seven shows per week and it would still be cheaper than cable (I believe). What about shows that air almost daily, like Late Night with Conan O'Brien and The Daily Show? At this price, it would take \$30-\$40/month to watch them. As they expand their service, I hope they take this into account.

One of the biggest advantages hasn't been widely published: no branding. These days, it is despicable how much content is lost in the audio and video. We have the permanent channel brand, the more and more common rating brand (aka, TV-14), and the annoying as all hell advertisements. The branded advertisements have gotten out of control; they can take up to 1/3 of the screen at any time with distracting animation and many have audio! What the hell is this? They are destroying our content. Finally, there are the occasional scrolling public service announcements. Branding is a big reason I'm not getting cable in my apartment. Even though I'm not getting cable, I don't plan on using the TV show store in its current form at all. Why? Resolution. I've been wishing for HD optical discs for a number of years. My mom mentioned that she tries to avoid SD content these days, because it looks so much worse than HD. With a huge shift to HD in the coming years, I don't plan on buying content that is 1/4 the resolution of SD content. These postage stamp sized videos are 1/27 the resolution of what I want to be watching. You can call me elitist, but I enjoy the HD. Hell, before most content was in HD, I'd watch boring shows just to enjoy the high definition. In the beginning, I even resorted to the local news!

Front Row looks pretty neat, but is somewhat limited on the computer screen. My parents (who will likely be buying an iMac this weekend) do not want to watch movies or TV shows on the computer. Since you can't burn your purchases to DVD, they can't watch it on TV either (without having the computer in an awkward spot near the home theatre). What Apple really needs now is a new Airport Express that supports video (high definition, of course) and can forward the remote control back to the computer in another room. I'm convinced my family would buy that.

I may seem critical, but I think this has lots of potential and I want to see Apple get it right from the beginning. I love the idea of buying and downloading 1 v shows, but the resolution just isn't there yet; I'm still sticking with DVDs. Finally, I'm really impressed that Disney has signed on to this; this is a show of good faith by Iger. Hopefully this is the first sign that Disney has reached a turning point after Eisner left.

[Permanent Link](#)

2005-10-11 18:22:29

Okay, it appears the new Search History does work in Safari. I don't know if it was some bad code on their end or a network error or what, because it wasn't working when I first tried (but it did later in the day). Also, I just tried out Yahoo's [My Web 2.0](#) and imported all my del.icio.us bookmarks. It seems pretty cool, I'm surprised I hadn't heard of it before. However, it doesn't seem like it is [taking off](#). In other news, no internet until Saturday :(I thought it might have come today, but I guess Speakeasy's e-mail was misleading.

[Permanent Link](#)

[Show all entries](#) | [RSS Feed](#)



All content on this website created by Andrew Hitchcock is licensed under a [Creative Commons License](#). Everything else is copyright its respective owner.