Parallel Spider

Analyze the World

A Business Plan

(Updated: Mar 2013)

[caveat emptor: all plans are subject to change since the only things certain in life are death and software bugs.]

TABLE OF CONTENTS

Executive Summary

Parallel Spider is Golden Hill's first product. The system is designed to be a tool for students, parents, journalists, marketers, and anyone else interested in viewing the Internet through a unique statistical lens that focuses on aggregate details. Tools exist to scrape websites, to analyze data, and to analyze websites, but nothing currently available combines all three into a service for "external" website analysis.

The system is constructed entirely of open source components and runs in the cloud. Users interact with Parallel Spider through a client-side web interface. They initiate crawls that launch dozens of computers, or spiders, which download one site, or multiple sites, and analyze those sites according to the inputs provided by the user. The results are presented in multiple formats with tools for visualization that help users to both understand the data and to present their findings to others in a professional manner.

The initial types of analysis performed by Parallel Spider are text, link, context, and synonym ring analysis. Users can target particular sections of a site or specific parts of a page. Later enhancements will include time series analysis, an application program interface for web scraping, and the incorporation of other data sources and analysis methods to expand the system into a general data analysis tool.

Parallel Spider will initially target the low-cost, niche markets of academia and journalism. While limited, these markets will enable use case development and product refinement in a more protected ecosystem. These initial markets will then serve as the staging ground for future moves into the more competitive markets of web scraping, online listening, and data analysis. These follow on areas represent billion dollar markets with enormous growth potential.

An implementation plan targeted initially toward students will allow Golden Hill to develop the system with minimal resources and limited risk. Small marketing campaigns will be initiated at little cost to acquire test users for product development. Local user studies will also be conducted. After a few development iterations, Parallel Spider will be targeted toward journalists. This latter group of users could then provide a credible source of free marketing as Parallel Spider expands into the more competitive follow on markets.

Parallel Spider is both a data acquisition and a data analysis system. It can be implemented and developed for limited resources, yet has the potential to expand into some of the largest and fastest growing markets around. Parallel Spider is a tool that can help users from vastly different areas perform research into the largest data source available, the web. It can help students with homework, journalists with stories, and marketers with business intelligence. Parallel Spider is a tool for people to analyze the world around them.

Product Description

Parallel Spider is a platform for external website analysis and comparison. The software is designed to provide individuals in less technical fields the ability to easily perform data analysis on websites: What words are used? What sites are linked to? What ads appear? Additionally, it provides the ability to compare websites based upon these features.

Parallel Spider has three primary components, the web client (SpiderWeb), the scraping and analysis portion (SpiderEngine), and the glue between them (SpiderServer). The web component is built with Angular. The server is currently running Twisted with multiple Redis data-stores. The engine is Starcluster running Hadoop with a Redis data-store for maintaining state between the mappers, or spiders, which parse the sites with lxml. The system is designed to rapidly download and analyze websites, to present the results in a visually appealing manner, and to scale easily with an increasing user base.

Initially, the analysis will consist primarily of word counts and other simple MapReduce scripts. These word counts will be used to analyze text in various ways:

Text Analysis – Users can see what are the most used visible words, the most used headline words, and the most used hidden words on the site.

Link Analysis – Users can see what are the prominent links, what they say, and where they point, both internally and externally.

Context Analysis – User can find out which words appear in context with another word and how often those words appear.

Synonym Analysis – Users can determine how often a group of related words are used on a site, either predetermined lists, WordNet lists, or user created lists, and how this usage compares to the Internet as a whole.

Selector Analysis – Users can enter their own path qualifiers, XPATH selectors, or CSS selectors, allowing them to analyze the parts of a site that they are interested in.

Site Comparison – All the above analyses can be performed on multiple sites allowing the user to compare the results.

Visualization – All results will have simple to use visualization tools based on D3.js. These tools will help users understand the data and present that data to others in a professional manner.

Future enhancements will depend upon usage patterns, but the following are areas of interest for expansion:

Public API – By opening up the application program interface (API), Parallel Spider can charge other companies who wish to use the parallel scraping technology for their own needs.

IPython Library – An easy to use python library that allows users to integrate Parallel Spider's results into an IPython client on their local computer could greatly enhance the tools appeal to data scientists.

Starcluster Plugin – Allowing advanced users to operate the analysis engine on their own cluster could generate open source contributions to the core Parallel Spider technology as well as provide leads to potential developer talent. Also, open sourcing the engine would be a goodwill gesture and could help the company toward constructing a more general PAAS / data analysis system built upon open source components.

Social Feeds – By incorporating the Facebook and Twitter streams, Parallel Spider could provide new data sources relevant to academia and journalists, as well as begin an expansion into the online listening market.

Power Parent – While Parallel Spider is designed to be a more general tool, a specific version allowing parents to analyze sites, rate sites, and monitor their children's social feeds my generate significant revenue by helping parents protect their children online

Expanded Data Sources – Web content should be expanded to include PDF, DOCX, and so on. Users should be able to upload text files or analyze digital books already online. Log files should be incorporated as the system moves beyond external data acquisition and analysis and into more general data analysis.

General Statistics – With more data sources, the system could branch into an online data acquisition tool: Parallel Spider; and a more general data analysis tool: Mr. Feynman. This offshoot could target the markets of traditional desktop clients such as SPSS with an online system where people pay by usage and have additional features that can only be provide in the cloud.

Time Series Analysis – The ability to analyze the trends of words on websites would be beneficial to all use cases. Time series analysis would also greatly enhance the products evolution into a more general data analysis tool.

Machine Learning – With new data sources and the expansion into more general data analysis, allowing users to use ML techniques to analyze desired outcomes would be a logical step. SPSS currently offers this ability, but the advantage of an online system is that an API could be provided so that a company could acquire the data, train the algorithm, and then connect the algorithm to their system with ease.

Analytics – Although a competitive area, by targeting analytics, Golden Hill could connect the data triangle: external data acquisition, internal data acquisition, and data analysis.

Platform as a Service – External and internal data acquisition and analysis would allow Golden Hill to make a strong push into PAAS. After helping companies acquire and analyze data, the logical next step is data storage. Following that, enabling companies (and other developers) to build tools for the system so that employees can generate new data and interact with the current data in new ways may allow Golden Hill to take substantial market share from the currently predominant platform providers.

These future enhancements are tentative and subject to change as Parallel Spider develops. Additionally, as the service grows, new areas for expansion may open up. Parallel Spider will initially stay focused on "external" website analysis, all the while, keeping eight eyes open toward becoming a more general data analysis tool and growing into any other markets that show potential.

Market Analysis

The exact market size for Parallel Spider is difficult to analyze since there are no easily comparable companies performing a similar function of crawling and analysis. Scraping, Online Listening, and Data Analysis all share similarities with the service, but target markets and use-cases that Parallel Spider will initially avoid. The market for the system within academia and journalism could vary greatly. Following are some possible estimates with various use cases (See Appendix 3 for data sources.)

The student and academic market is probably about $10 million a year. 20 million college students in the US, assuming only 10% have an applicable assignment in a given month, would represent 2 million potential users. Optimistically, assuming that of those 2 million the company is able to get 10% usage, this would represent 200,000 monthly visitors. With a price range of about $3 per month, the total revenue would be $600k per month. Adding in non-college students, international students, and other academics, a market estimate of about $10 million seems reasonable.

As an example use case for academia, consider a Sociology major, Steve, who is studying bias in the news media for a paper. Steve identifies leading journalism outlets: NBC News, CNN, Fox News, The New York Times; and a handful of controversial issues: abortion, taxes, gun control, war. He then uses Parallel Spider to perform a context analysis for each issue on all the sites. Visualizations display the words each site uses most in regards to each issue. Steve downloads pictures summing up the data and attaches them to his paper. A+.

Or consider a health care researcher, Savannah, studying emerging diseases such as Celiac's. She identifies a handful of leading health sites: Yahoo Health, WebMD, Celiac.com and uses Parallel Spider to analyze the comment sections to determine what words are used most in regards to the disease. She gains insights into the most common symptoms according to the sites' users. She documents her discoveries in a research paper targeted to healthcare providers, helping them to more ably diagnose the disease.

The journalism market is probably a million dollars a year. 60 thousand journalists in the US, with an optimistic 5% usage rate and a slightly higher price of $20, would represent monthly revenue of $60k, totaling to $720k a year. Adding in international, $1 million seems a plausible estimate. This market may be larger.

As a use case, consider Matt. He runs a news agglomeration service with a slightly libertarian slant, but also does his own stories. He is convinced that the country's fourth central bank, the Federal Reserve, is up to no good. He users Parallel Spider to perform a textual and link analysis of the FED's website. He discovers that the most used word on the site is "red herring" and all the external links point to J.P. Morgan Chase's website. Now that's news.

The market for parents is likely $300 million, but already very competitive. Seventy-six million children in the US, with half in an age appropriate range and 2 children per couple, represents

about 19 million potential users. Assuming only a quarter use tools to help monitor computer activities leaves about 5 million potential customers. At an average $5 per month this equates to a possible 300 million dollar market.

As an example user, consider Martha. She wants a better understanding of the sites her son Stewart is visiting. She inputs the links of all the tumblr blogs, twitter feeds, and other sites he visits most often. Parallel Spider lets her see what words are most used, how often words associated with violence, sex, and drugs are used, and how this usage compares with other sites on the Internet. She sees that the rating of the site *insidertrading.org* is only a "G". She votes that it should be "X". Worried about Stewart's behavior, she enters his Facebook information into the listening system so as to receive daily reports with an overview of his online discussions. She flags the word "insider trading" to be tracked specifically.

The tangential markets of web scraping, online listening, and data analysis are much more significant and growing rapidly. The online listening platform market is currently estimated to be around $800 million. Web scraping itself is probably a fraction of this at $200 million. The true target though is data analysis, which is currently estimated to be $5 billion with the potential to grow to $50 billion by 2025. The markets for students, journalists, and parents are the staging grounds to execute flanking attacks into these much more lucrative business markets.

Competition for Parallel Spider is non-existent in the student and journalism markets. Parental monitoring software has some competition such as Safetyweb and SocialShield. In web scraping, competition exists from companies such as 80legs and ScrapingHub. Pure web scraping's big brother online listening has talented companies such as SalesForce wth Radian, NM Incite, and Lithium Technologies. In data analysis, competition is the fiercest with computer giants such as HP with Vertica and SAS; and hot startups such as Palantir and Continuuity all striving for the data scientist crown. Within data analysis, textual analysis companies such as Saplo are doing interesting things yet missing the data acquisition component. (See Appendix 1 for a more thorough list of potential competitors.)

Despite fierce completion in the larger markets, by initially focusing on the low-cost, non-competitive markets of students and journalists, Parallel Spider should be able to fly below the radar of the companies in the larger markets. When the product is refined and ready for a larger ecosystem, depending upon user experiences and system development, moves can be made into the more competitive markets of web scraping, online listening, and data analysis. All combined, these markets have both incredible size (billions) and incredible growth (10%+).

Strategy

Parallel Spider threads the needle between Scraping, Listening, and Analysis. It differs from scraping in that it performs data analysis. It differs from listening by taking a more narrow focus (at least initially) and offering a more general tool. It differs from data analysis in that it provides data acquisition via scraping and the analysis (although limited initially). This situation presents the possibility to grow in an unencumbered market that is tangential to more significant, but increasingly saturated markets. If Parallel Spider is able to succeed as a service in this hybrid market, solving the problems of important but oft ignored individuals such as students and journalists, it can use that position to push hard into the other three areas.

Business Model:

Parallel Spider will likely either be sold in a manner similar to phone services or Amazon Web Services (AWS). As a phone service, people can pay for one time uses or monthly contracts, with prices dependent upon the number of users, the pre-allotted usage plan, customer support services, overage rates, and other special features. As a service similar to AWS, users receive monthly charges depending upon their usage of the system and often have an initial free tier.

Development Strategy:

To gain acceptance, Parallel Spider must give users the ability to gain valuable insights from websites. It must provide the ability to gather actionable "external" intelligence for people and businesses. Since a general tool for the current task doesn't really exist, the system's growth must be more organic, with the interested domain experts helping to guide its evolution. It must blend the ease of use of a web applications such as Gmail with the options of desktop applications such as Excel and SPSS.

Students, primarily in the liberal arts, will be the first group offered the system with incentives to help it gain traction such as Amazon gift certificates. Helping students with homework is not only a problem worth solving, but also when combined with the opportunity to win a little spending money for what has to be done anyway, is a strong motivator to adopt for a group that is predisposed to explore new technologies. Also, since many students are collocated, they can be exposed relatively easy through both inexpensive advertising such as flyers and word of mouth. By focusing on students the system can be refined for various use cases prior to being offered to a larger audience.

The next steps will be to see if journalist and parents can use the system. Again, like students, helping journalist research stories not only solves a problem worth solving, but also provides a free form of credible advertising. Helping parents analyze the sites their children visit is a less glamorous use case but may provide a significant revenue stream.

Growth Strategy:

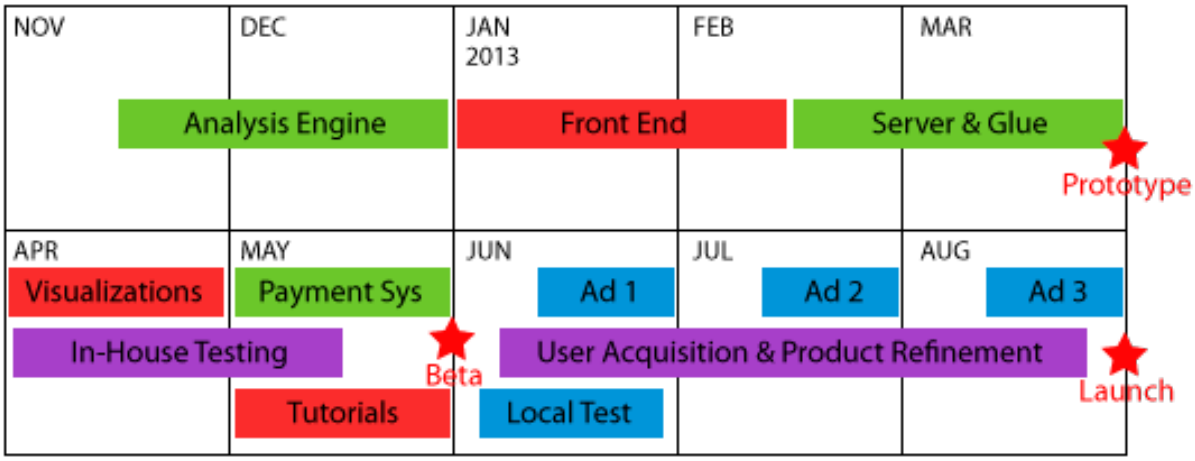
The ultimate target group is business. Assuming the service has been refined and proved successful through the previous use cases, Parallel Spider will be marketed to companies interested in performing competitive analysis, gathering customer intelligence, and ascertaining market insights. Rather then define the finished content in a report, the system will allow the domain experts to perform the analysis themselves. Marketing departments can use the tool to perform their own analysis rather than paying millions for reports and systems that limit their ability to analyze the areas they know better than anyone else.

Additionally, during this progression, expansion opportunities may be seized in the other tangential markets. An API will be offered for people and companies to use the parallel scraping capabilities. The analysis portion will be enabled for books, log files and other textual information beyond websites. The sites internal analytics may also be offered as a separate service.

By starting with niche markets, growing the system in steps, and validating use cases along the way, Parallel Spider can be developed for less cost and with less risk then would otherwise be required. Additionally, by helping students and journalists, Parallel Spider gains two additional benefits: training the next generation of business users and obtaining one of the best possible types of marketing. Finally, with the significance of the adjacent markets, a successfully product has an extremely strong possibility of a favorable exit.

Implementation

The first step is the completion of a working system. A prototype should be available at the end of March. The Alpha period, between prototype and Beta, will primarily focus on conducting in-house usability testing and making refinements to the system based on the results of these tests. Three other focal points during this period will be: one, the addition of visualizations in order to aid users in both understanding the data and presenting it; two, creating easy to use tutorials to help people not only use the system but also understand what it can do; and three, adding an online payment system. While many users will be offered $20 of free usage, the payment system is necessary to deter malicious use of the system, to determine people's need for the system, and to try to actually generate revenue.



Calendar 2012-2013

By the end of May the Beta version should be complete. The following three months will be dedicated to user acquisition and product refinement. Initially, usability tests will be conducted with local college students to determine possible use cases and quickly determine roadblock issues. Two weekend tests will be conducted with students from different majors, likely including Anthropology, Sociology, and Political Science. These tests will be spaced two weeks apart so that an adequate time for system refinement is available.

Upon completion of the first usability test, the first of three ad campaigns will begin. Each campaign will target 10 schools throughout the country, offering users the ability to "get paid to do your homework." All students who sign up for the system will be given $20 of usage free and the opportunity to win a $50 Amazon gift certificate by submitting any homework they do that involves Parallel Spider. Once a week, a winner will be chosen, and their homework posted in the help section to demonstrate a possible use case of the system. As in the usability testing, a two week window will be preserved between campaigns to absorb lessons learned and make steering changes.

If the first two ad campaigns show promise, the third will target journalists in addition to students. Free accounts along with sample use cases will be offered to journalists who may be enthusiastic to either using the system themselves or posting about results generated from it. This campaign will focus not only on exploring a new market, but also generating publicity for the official launch at the end of August.

Financial

Profits for the Beta period will be minimal to non-existent. Free $20 accounts will be distributed freely in order to attract initial users and refine the site based upon their feedback. Fortunately, costs during the Beta period should also be minimal. System, advertising, usability test, design, and coding costs should not exceed a total of $11,000 over the planned three month Beta period (Table 1).

Advertising: $3500
- 3 Ad Campaigns x $1500

Usability Testing: $3000
- 2 Local Tests x $1500

Site Design: $1500
- 1 Site Overhaul x $600
- 3 Illustrations x $300

Code Audits: $1000
- 4 Audits x $250

Amazon Web Service: $2000
- Compute x $1000
- Storage x $500
- I/O x $500

TOTAL: $11,000

Table 1: Expected Costs for the 3 month Beta period

The system itself should have limited fixed costs. The web servers and the master will have usage purchased ahead of time, but these instances should initially cost no more than $100 a month. The analysis engine will take advantage of the Amazon Spot market, using micro instances at $0.003 per hour. In this manner the system can grow and shrink depending upon usage. 100 spot instances would run about $200 a month. To deal with possible price spikes, a small backup cluster will be held in a stopped state, only paying for the EBS images. $2000 should be sufficient for the first few months of operations on AWS (See Appendix 2: AWS Costs).

When scaling, one micro instance should be able to handle about 5000 pages in an hour (later optimizations could certainly increase the processing rate.) When including input and output for both the crawl and user downloads, the cost per 10000 pages will probably be around a penny. If we conservatively charge $0.0001 per page, then 10000 pages would bring in a dollar for a cost of a penny. Upon stabilizing the cost model, if these margins are accurate, discounts could easily be offered for heavier users, and prices could be reduced across the board to suffocate competition.

Initial ad campaign costs will derive from hiring students to post flyers, flyer design and printing, and amazon gift certificates. Gift certificates will cost $50 dollars a week. Flyer printing and posting will cost about $800 per 10 schools, assuming $60 dollars per student to post the flyers and $200 to print and deliver them. $500 should be sufficient to design two flyers for A/B testing. In total, a three month ad campaign targeting 30 schools could be conducted for about $3500.

Local usability testing costs will consist of hiring student testers, purchasing some food, and renting a location. $100 per student for 4 hours on a weekend afternoon is a fair price. Assuming 5 students on Saturday and 5 students on Sunday total pay costs for a weekend

would be $1000. Students will be required to have their own laptops. Setting aside $250 for food and room rental each day, total costs for a weekend of testing would be $1500. Two usability tests would therefor cost about $3000.

Other initial expenses will include site design and code audits. $600 will be used for professional help on the overall site design, along with $300 each for site illustrations. A site security audit can be performed for $50. Additionally, $250 will be offered for code reviews for 4 parts of the code base: Angular, Twisted, Hadoop, and Lxml. Design and coding costs should total about $2500.

Technical

Parallel Spider is constructed entirely from open source components. The analysis engine is built on Starcluster, Hadoop & Dumbo, Lxml, and Redis. The front end is built with Angular and D3.js. The backend server architecture is Nginix, Twisted, Redis and PostgreSql. All components are deployed on Amazon Web Service (AWS) but the only dependencies are storage (S3) and compute (EC2).

Clients interact with the service through HTTPS. Angular (from Google) is used to provide a web application for end users. A native mobile application may be constructed later, but the current software will be unlikely to target phones due to screen size. However, tablet computers are a feasible use case, but an HTML5, CSS3, JavaScript solution with a responsive design should be sufficient for the foreseeable future.

Initial page requests are served from Amazon's Content Delivery Network (CDN) since the main page, login page, and information pages are all static (HTML, CSS, JS, PNG.) All subsequent requests are sent to a RESTful interface over HTTPS. An NginX server acting as a load balancer receives a request and passes it to a Twisted server for processing. On login, Twisted queries PostgreSql for the user's information. Should the encrypted password match the stored one for that user, two session cookies are created and placed into Redis. One is short term for purchases (expiring after 15 minutes) and one is for analysis-data access (expiring only on logout).

Subsequent requests are either to retrieve user analysis data, all of which is stored on S3, or to initiate a crawl. For the latter action, request data is saved in a Redis queue. A Twisted instance running on the master of the analysis engine periodically checks the queue and places requests in the Sun Grid Engine attached to StarCluster. This second queuing system enables an auto-scaling of resources using Spot instances.

When compute resources are available, the crawl is executed over multiple mappers in Hadoop. These processes use another instance of Redis to maintain state, so pages are only crawled once. These mappers download and parse the page using lxml, and then perform the analysis. The initial results are sent through another MapReduce operation to create a JSON response, which is then saved to S3. During the analysis, the engine updates Redis, which is used to keep the client informed of the crawl's progress and the moment of completion, at which point the client retrieves the JSON results from S3.

Legal

One of the key issues regarding Parallel Spider is the legality of web scraping. Many sites have terms of service agreements that prevent such behavior; however, these terms usually provide exceptions for Google and other search engines. These web-bots have the advantage that their parent platforms generate enormous traffic for the scraped website. While Parallel Spider can't rely on this commercial advantage, the company isn't reproducing copyrighted material either, and is simply providing users of a site the ability to analyze it. However, Parallel Spider may be perceived as no economic advantage to certain site owners, perhaps even a threat, and they may even demand that Parallel Spider stop scraping their site.

The company will follow a three-sided strategy to avoid the possibility of legal action:

One: achieve legitimacy by helping students and journalists research the world around them. Focus on the fact that freedom of speech requires the freedom to listen, read, and understand the words that are being created. Position a legal attack against Parallel Spider not as one of company against company, but company against openness.

Two: be considerate and a good "netizen". The analysis engine will only hit sites so hard and so often. Download rate will be throttled and caching will be extensively employed. Help pages will be displayed prominently to aide in the quick resolution of issues. Areas of the site excluded by robots.txt or passwords will be honored.

Three: blend in with the masses. Parallel Spider will not request permission before visiting a site. It will allow users to go were they please. The system implementation should strive to prevent a site analysis from appearing as anything more than another search engine. As the analysis engine grows and shrinks, computers will start and stop, causing the assignment of new IP addresses within the AWS infrastructure. Therefore, no small range of IP addresses will be responsible for site traffic.

Appendix 1: Possible Competitors

Computer Giants:
Google
Amazon
Microsoft
IBM
Hewlet Packard
Oracle

Data Analysis:
Vertica (HP)
Opera Solutions
Mu Sigma
Aster Data
Splunk
Greenplum (EMC)
Cloudier
Calpont
Fractal Analytics
Think Big Analytics
Digital Reasoning
Datameer
SAS Institute
SAP

Newer Data Analysis:
Palantir
Platfora
Cloudera
Continuuity
Domo
Metamarkets
Odiago
Karmasphere
Saplo
Fivetran

Online Listening:
Sales Force
Converseon
NM Incite
Visible Technologies
SDL
Cymfony
Evolve24
Dow Jones
Collective Intellect
Attensity
Clarabridge
Overtone
Networked Insights
Synthesio
Lithium Technologies
Semantics3
Connotate

Web Scraping:
Promptcloud
80legs
Scrapinghub
Grepsr
Mozenda
Screen-Scraper
Spiderman
Bobik

PowerParent:
Safetyweb
SocialSheild
TrueCare
Minormonitor

Appendix 2: AWS Costs

Back of the Hand Calculations for Scaling Costs

Compute:

- An analysis takes about 15 micro instances 5 minutes to handle 5000 pages. So 10 users (crawls) would require 150 micros. Throughput would be 120 users an hour or 2640 users a day. 150 micros at $0.003 an hour would cost about $11 a day, or about $330 a month.

Storage:

- The average page is 50kb (text only) and the average analysis is 5000 pages, so the average analysis requires a cache of 250 MB. For 120 users an hour a total cache would require 30 GB. At $0.095 a GB per month, hourly caching would cost $3 a month.

- Assuming the average analysis is 10 MB and an average user performs 10 analyses, then each account would have 100 MB of storage. 50,000 users would require 5000 GB. At $0.095 a GB this storage would cost about $475 a month.

I/O

- Assume users look at their results and 3 other results totaling 40 MB during an analysis. 2640 users in a day would download around 105 GB. At 0.120 per GB to download from S3 to the Internet, the cost would be about $13 a day or $390 a month.

- Downloading the pages from the Internet into EC2 is currently free. Additionally, transferring data from EC2 to S3 is free if both services are hosted in the Northern Virginia Region.

- Assuming the web app is approximately 1MB, with 2640 users a day, 2.6 GB will be downloaded from CloudFront at $0.120 a GB. This cost would be about $10 a month.

Appendix 3: Market Data

Big Data:
http://wikibon.org/wiki/v/Big_Data_Market_Size_and_Vendor_Revenues

Online Listening:
http://www.networkedinsights.com/pdfs/The_Forrester_Wave_Enterp.pdf

Web Scraping:
http://online.wsj.com/article/SB10001424052748703358504575544381288117888.html

Text Analytics:
http://www.informationweek.com/software/business-intelligence/text-analytics-demand-approaches-1-billi/229500096

College Students:
http://www.census.gov/newsroom/releases/archives/facts_for_features_special_editions/cb11-ff15.html

Journalists:
http://www.bls.gov/ooh/media-and-communication/reporters-correspondents-and-broadcast-news-analysts.htm