Web Analytics Project
Documentation

Project Overview

This project began 5/27 with a request to do a research review of issues and questions related to comparing anonymous web traffic with portal traffic. Loosely put, the question at hand was something like "Is there an existing framework for comparing anonymous web traffic vs. known-user portal traffic? Or is it always an apples-to-oranges comparison?"

On 5/28 a bibliography and description of research findings was handed back. The short answer was "no, there is no research about analyzing anonymous and known-user web traffic side-by-side, although there is a lot of research about both topics individually."

By 5/29 the project expanded and changed direction to include working with the existing NetTracker web analytics software implementation. The new goals included exploring the NT product, and to find the answer to 3 questions about the web traffic of the anonymous, public-facing Welch webserver:

1. Of the traffic what portion is domestic vs. outside the country?
2. Of the domestic traffic what portion is internal to JHU and what portion is external?
3. Of the internal traffic what portion comes from what organizational unit and/or physical location?

On 6/2 and 6/4 Shubin Wang and Dongming Zhang were interviewed with a set of questions designed to establish the scope and requirements of the project. The goal of these questions was to understand how the data related to Questions 1-3 was going to be consumed in order to develop requirements for the new system/process. For example, one question was how web traffic (which is archived daily) relates to the business calendar and any administrative process for comparing traffic from period to period (i.e. calendar year, academic year, etc.). Other questions included who was going to use the reports, what kinds of decisions would they be used to evaluate.

For the next week NetTracker was tested and evaluated for suitability to answer the 3 questions. Due to the relative inflexibility of the product, it looked like it would be a lot of work to customize NT to answer the questions, and it might not be possible at all.

Up until 6/12, the goal of the project was find the ideal configuration and customization of NetTracker to show the answers to questions 1-3, however on that date Dongming proposed the building of an external system/process that would start with a NetTracker export, but would otherwise be handled outside of NT. Note: This new architecture would still leave the door open to doing host resolution and filtering in NT.

Between 6/12 and 6/19 data storage structures were developed in MySQL to store datasets which are needed to answer the 3 questions in a single database which could be used in many ways to query and publish the data, and which would be well positioned to migrate to a different commercial database if needed. These datasets are JHMI IP address ranges (including building and floors), country IP address ranges, and the NetTracker “visitor report” dataset. On 6/19 Dongming got a paper output showing a histogram of the Question 1 data (based on the working sample, which at that time was a single day’s traffic). During this period, data transformation processes were also being developed and tested.

By 6/24 the data had all been normalized and imported in MySQL and the first set of complete data that shows the answers to Questions 1 and 3 was developed. Unfortunately, Question 2 has remained impossible to answer conclusively both because the complete range of JHU IP addresses is not known, and not all external addresses can be resolved to a particular country.

These are the question 2 caveats: Regarding JHU IP addresses, it is an important premise that all non-JHMI web traffic with an IP address starting with "10." is somewhere in JHU, however some addresses in that range are used for things like servers, etc. which are not the traffic we are interested in. Furthermore, there will probably also be some portion of internal human-generated traffic which can't be resolved to location/organization. Regarding domestic web traffic which is thought to be external to JHU (i.e. IPs starting with other than "10."), not all traffic can be resolved to a country, and some traffic coming from outside is likely really coming from within, through external facing jhu hosts, domains, and address ranges which make a trip outside the firewall for some reason.

On 6/25 the project again shifted and became almost exclusively focused on documentation and preparations to hand it off to Shubin Wang.

The next steps in the project are:
1. Develop a reporting process to chart or report the answers to questions 1 and 3. This might be as simple as tabled data presentation straight from MySQL, or more complicated like reporting or charting.

2. Implement filtering in NT to eliminate known non-human agents and improve the quality of the data. To start with, all the top visitors are bots or applications of some kind. The parsed host and huser fields in tblVisitor are available to aid in constructing regular expression-based NetTracker filters to remove these users' traffic from the data.

3. Enlarge the sample from a single day to a representative sample. This should be done in stages, starting with 3-days, 7-days, and so on, and steps should be taken to get an appropriate range of weekend days, etc. Note: with current technology the import process into NetTracker of a large sample will eventually take many hours for a single instance. Likewise, the various data transformation processes and queries will also take a long time as well, and sometimes these require a dedicated computer time to run. There may be a need to migrate to a commercial database on a dedicated server, and to develop indexes at this stage.

4. Take a significant sample of data to Pat Kowalski, the Space Systems Administrator at the JHMI Space Inventory Office (pkowals3@jhmi.edu, 410-614-5625), and try to construct a way to get space database organization data and location data to match our location data. To do this, there will need to be a mapping made between locations in both systems. Since the space database is authoritative, it would be best to use the space inventory location data throughout the application. Since it is just the organization data that is really secret and controversial, it may be possible to get a full campus/building/floor superset of data. The agreement with Ms. Kowalski is that we will bring her a dataset of locations that we want to know ownership of and we will discuss whether it will be possible to get ownership data for those locations. If we get it, one of our obligations is to make sure that we don't disclose ownership/location organization data to the university community.

5. Implement the new organization and location data into the existing database and reports.

6. Finally, developing some data structures for meta-information about the imports themselves with foreign keys in the data would allow the MySQL database to have a large sample in it, and allow the sub-setting of imported records to run various queries against it. This approach would allow new questions to be more rapidly modeled and answered (like “what is our traffic like on Tuesday’s”). The practice of not storing information about which log files have been aggregated together to make a given sample makes the data very opaque once it has passed through NetTracker.

Architecture Overview

The architecture of the system is divided into 4 parts: data source layer, data transformation layer, database layer, reporting layer.

Data Source Layer

The data source layer is different for each of the 3 datasets.

The IP Country data comes from MaxMind (GeoLite Country database):
http://www.maxmind.com/app/geolitecountry
http://geolite.maxmind.com/download/geoip/database/GeoIPCountryCSV.zip
Contents: GeoIPCountryWhois.csv
C:\pepper\20090527_PortalVsWebUsers\hostAnalysis\maxMind\GeoIPCountryWhois.csv

The JHMI IP data ranges come from Joanne Wroblewski (actually Robert Gelder sent the file):
jwroble@jhmi.edu; RGELDER@JHMI.EDU
C:\pepper\20090527_PortalVsWebUsers\interfaceAndApp\import\IP address per building.xls

The NetTracker datasets come from importing the data into NetTracker and then exporting it twice, once with host resolution applied and one without host resolution.
The names of the 2 most recently used test/sample imports are:
C:\pepper\20090527_PortalVsWebUsers\interfaceAndApp\import\
nt_visitor_1245705152_18557_20090622_host.csv
nt_visitor_1245705373_18606_20090622_ip.csv

Data Transformation Layer

Each of the 3 datasets has an access database containing tables to temporarily hold the data and a linked table to the MySQL Analytics database. The data transformation process involves several phases: first the data is imported into a less-constrained table, which has the same structure as the imported table. Often this is done with an import specification, though sometimes it is not. Then the data is transferred to a more constrained table, which matches the structure of the MySQL Analytics database table. Finally it is exported into the linked table. At any point in the process transformations may happen to the data with queries and/or with code, depending on the need.

Source Data File	Access File	MySQL Analytics Table
IP address per building.xls	jhuIP.mdb	tbljhuip
GeoIPCountryWhois.csv	IPCountry.mdb	tblipcountry
nt_visitor_1245705152_18557_20090622_host.csv nt_visitor_1245705373_18606_20090622_ip.csv	visitor.mdb	tblvisitor

Database Layer

The MySQL Analytics database stores the three datasets in different tables: tblipcountry, tbljhuip, and tblvisitor. It also has some views. FYI: MySQL has built-in functions to convert back and forth between "IP Address" (a string, that can't be sorted into octet order) and "IP Number" (a very large number, specifically an unsigned integer). These function are called inet_aton and inet_ntoa (aton means "alph-to-numeric" and ntoa means "numeric to alpha"). Using the IP Number allows briefer, simpler SQL logic to build associations between the records in tblvisitor (which has an IP address) and the the other 2 tables, which each contain a range of addresses.

Reporting Layer

The reporting layer hasn't been finished yet. It might not need to evolve beyond tabled data. If reports are needed, then the easiest way to do this will probably be to use Access's excellent report environment to generate reports. One exception to that theory would be if there are needs to do charting, in which case it is really an open question of how to design reporting. Luckily MySQL is well-positioned to allow development of charting/reporting in a web technology if that's what makes sense.

Data Import Process

The data transformation layer consists of 3 access databases, one for each of 3 types of datasets that are imported into the MySQL Analytics database.

IP Country

The IP country data will probably change the least and may never need to be imported again. Perform these steps to repeat the import:

1. download the file from:
2. http://geolite.maxmind.com/download/geoip/database/GeoIPCountryCSV.zip

3. Open the IPCountry.mdb database and get an empty tblGeoIPCountryWhois table. This can be done by making a backup of the existing table and deleting all the records, etc.

4. Import the GeoIPCountryCSV.csv file using the import specification: specTblGeoIPCountryWhois_Import. The way this is done is to follow the following steps exactly:
   a. right-click nothing import
   b. change file type to text
   c. select file GeoIPCountryCSV.csv
   d. advanced: specs: specTblGeoIPCountryWhois_Import: ok: finish
   e. save yourself some trouble and get the data from the new table "GeoIPCountryWhois" into the old table "tblGeoIPCountryWhois" using any strategy you want to copy just ip1, ip2, and countryCode tblGeoIPCountryWhois.

      fyi: many of the fields in tblGeoIPCountryWhois are not needed (such as both versions of the start/end IPNumber).

5. go to frmIPCountryImport and perform the 3 steps there:
6. Update the 4 start and end octets
7. Export the data into tblIPCountry (don't forget to delete the old data first)
8. Update the tblipcountry IP Numbers (this is an Access Pass-Through query, so it could be done in a Query Browser as well.)

As far as I know the other queries in the database are all left over from experiments with other IP country datasets, like signed vs. unsigned storage for IP Number (which wasn't needed).

JHU IP

The JHU IP address ranges come from the IP address per building.xls file. The data in the current version of the file has been normalized and contains records for every range of IP addresses (first 3 octets) within JHMI. The process of converting from the source dataset to the normalized data took about a day and a half, and made more sense to do in Excel because the data was so dirty. In the future if there is a need to repeat the process, it would probably make more sense to approach the changes as a “differential” rather than full update.

1. Populated fields for JhuIpID (a MySQL requirement).

2. The Excel-to-Access import process is done manually (unless there is an import specification in the Access file, in which case use it).

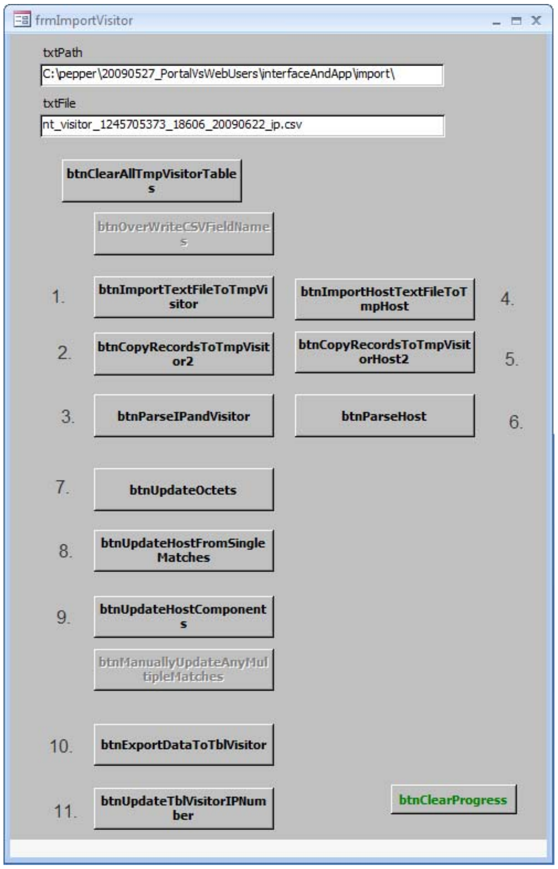
3. Do frmJhuIPImport: btnUpdateTmpJHUIPOctets

4. Use query qryExport_tbljhuIP to load the data into MySQL.

5. frmJhuIPImport: btnUpdate2NewSubnetNumbers will run the pass-through query qryPostImport1_UpdateIPNumbers that populates the IP numbers. This can obviously be done more directly in a Query Browser.

NetTracker Visitor

The visitor.mdb Access file contains the data transformation logic that can be used to prepare the 2 visitor report export (CSV) files to be processed into a single import into MySQL. Unlike the other 2 datasets/databases, this will have to be done over and over again, and almost all the business logic is automated by frmImportVisitor:



frmImportVisitor

txtPath: C:\pepper\20090527_PortalVsWebUsers\interfaceAndApp\import\

txtFile: nt_visitor_1245705373_18606_20090622_ip.csv

btnClearAllTmpVisitorTables
btnOverWriteCSVFieldNames
1. btnImportTextFileToTmpVisitor
2. btnCopyRecordsToTmpVisitor2
3. btnParseIPandVisitor
4. btnImportHostTextFileToTmpHost
5. btnCopyRecordsToTmpVisitorHost2
6. btnParseHost
7. btnUpdateOctets
8. btnUpdateHostFromSingleMatches
9. btnUpdateHostComponents
btnManuallyUpdateAnyMultipleMatches
10. btnExportDataToTblVisitor
11. btnUpdateTblVisitorIPNumber
btnClearProgress

The first thing you should do is populate a new set of VisitorIDs in the ip csv file (see an old file for an example). These IDs are used as part of the "btnUpdateHostFromSingleMatches" process, and then MySQL imports them.

The btnClearAllTmpVisitorTables clears 4 temp tables that are part of the import process. They will need to be cleared the first time a new import is run, but may not need to be cleared in order to start a particular step over.

Perform all the steps in order. btnImportTextFile toTmpVisitor (1-3) assumes that the file is the "IP" csv file. btnImportHostTextFileToTmpHost (4-6) assumes that the file is the "host" file.

Note that step 10 loads the data into MySQL and step 11. runs the pass-through query that populates IP Numbers on the server.