Tracking a GIS – System Monitoring Tools for an Enterprise GIS

By Jaclyn Meade Cardillo
Advisor Jim Detwiler

Outline

Introduction & Research

Project Process

ArcGIS Desktop

Web Applications

ArcGIS Server

Geodatabase

Outcomes

Introduction & Research

Top

The goal of my project was to create a resource where users can go to find out how to deploy tracking within their GIS. In working with multiple government agencies, I have found that tracking of GIS usage was very limited, if at all deployed. There are many questions we can ask about our systems that tracking can answer. For instance, what is the most popular application, what are the most frequently used feature classes or which users are the most active? Besides popularity, we can also track performance and answer questions like what is my map service average request time and is that meeting the appropriate level of service for my end users? From a system planning standpoint, can an existing system handle a new application or five or ten? Without tracking in place there is no way to knowledgeably answer or quantify these questions. The answers can drive investment decisions, resource allocation and strategic planning for any organization administering a GIS. Since my organization uses Esri technologies, that was the focus for tracking with this project.

In the beginning of the research phase there were three main components of a GIS that came up consistently when investigating different monitoring tools: web applications, map services and geodatabases. I was also interested in tracking the desktop usage of ArcGIS, so that was added as a fourth component to tracking for this project. The plan was that for each component I would develop or deploy a tracking solution, create an easy way to understand the tracking data, and lastly take actions to improve my environment depending on what the analysis results shed light on. All of the resources created from the project would then be uploaded to a wiki page that any system administrator could access and have enough details to understand the benefits of each solution and deploy them in their organization.

During the research phase, I found more ArcGIS-specific monitoring tools than I thought existed. I also found tools to design, test, and analyze a GIS system. As shown in Figure 1, Esri recommends that all of these tools be used holistically in an ongoing process of planning, deploying and maintaining a GIS (Sakowicz 2013). I found that these tools overlapped very easily in that once you start asking questions about monitoring other questions come up in regards to system performance, system design, system capacity and application/load testing. I found myself a few times diverging into Esri's capacity planning tool or load testing tools like JMeter or System Test, but those tools were out of the scope of this project.

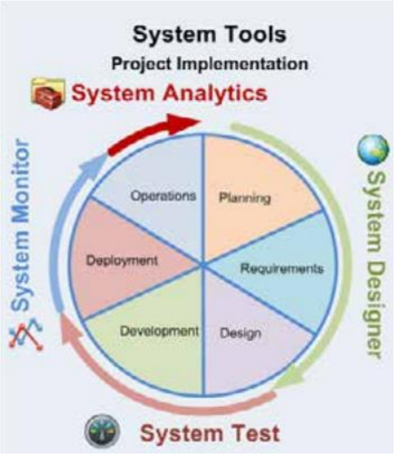


Figure 1. Esri recommendation for project lifecycle. (Sakowicz 2013)

For tracking tools I found a combination of free monitoring tools, pay-for tools, tools developed by Esri and tools developed by third parties. For this project I thought it was important to identify all of the monitoring tools available, but ultimately to deploy only freely available options. Esri has an adequate amount of free monitoring tools available and the best resources I found came from the ArcGIS Online content of Esri's professional services group. Out of that group I found the best free holistic tracking tool, Esri's System Monitor application. This solution provides a web-based interface where you can configure tracking on servers, applications, ArcGIS Server services and databases. In addition to tracking, you are able to set up alerts as well as configure actions to take place before alerts are sent out. For example, if a service becomes unresponsive, you can program System Monitor to try to restart the service and check if the service returns before sending an alert. System Monitor also allows you to set up dashboards to easily visualize your environment health at a glance or configure reports that provide deeper insight into the system (Enterpriselmp 2015).

Besides Esri solutions, I also found pay-for monitoring systems developed by Vestra and Latitude Geographics. Both organizations created similar web applications to Esri's, which have the ability to track servers, databases and applications. Vestra's application, called GeoSystem Monitoring, features the additional capability of tracking ArcGIS Online-specific services, web maps and hosted layers, as well as Esri's Enterprise License Manager (Vestra). Latitude Geographic's product, Geocortex Insight, appeared to have all of the same functionality as Esri's System Monitor tool (Geocortex). I included this information in the wiki in case an organization had the resources to purchase a commercial off-the-shelf solution.

Overall, during the research phase I found a lot of great monitoring tools that could have been utilized for this project. I did not find many resources in terms of user experiences with deploying monitoring tools, so I made sure that the step-by-step deployment documentation I developed was robust. I was concerned about venturing into a project that was not well documented in the online community, but I had a strong GIS team at work that I could lean on if I hit snags along the way.

Project Process

Top

The project consisted of three phases; a tracking phase, analysis phase and action phase (See Figure 2.). All of the information, documentation and solutions developed during the three phases was uploaded to the GIS wiki. The wiki included detailed how-to's, templates to use for easy analysis of the tracking data and possible actions that could be taken based on the tracking results. The technical level of the online resources was low enough that any GIS system administrator comfortable enough to deploy ArcGIS Server would be able to follow the instructions to deploy these solutions. The tracking wiki page was added to the established GIS wiki, http://wiki.gis.com. I chose this site because the GIS community is already familiar using it as a technical resource and I felt that it would be a great addition to that body of knowledge since I did not see any other tracking resources documented there.

Diving deeper into the three phases of the project, the tracking phase consisted of developing and deploying the different tools that I found during the research period. I focused on deploying the free software options and where one did not exist I developed a solution. The analysis phase consisted of taking the output data from the tracking phase and plugging it into templates that easily display the information into charts and graphs that answered targeted questions about that component. The last phase, the action phase, included taking the analysis information and developing an action strategy to enhance the existing component, plan for future infrastructure, or guide the system wide strategic plan.

Track
Analyze
Action

Figure 2. Three phases of the project.

The track-analyze-action lifecycle outlined above was applied to each of the four GIS system components discussed earlier: ArcGIS Desktop, web applications, ArcGIS Server and geodatabases. In the next section I will outline the project lifecycle for each of these four components.

ArcGIS Desktop

Top

The Desktop element was all about tracking the license manager usage and understanding how our community leverages the Esri Desktop products. Details of interest included how many licenses desktop users are consuming on average, what extensions they use, when users are in the system most and what the maximum concurrent usage is of licenses. This was a component where tracking software did not exist, so I developed a batch process that was scheduled to run at any interval. For my deployment, this interval was every fifteen minutes between 6am and 6pm. The batch process runs an executable that comes with every installation of Esri's License Manager lmutil.exe. When run with the lmstat parameter, the utility provides the computer names of all the users that are in the system at that time, along with the products being used, and writes those details to a text file. When run during business hours, the script gives fifteen-minute snapshots of who is using ArcGIS Desktop.

From there the analysis phase included the creation of an Excel template that took the text file outputted by the batch process, and processed it into record-level details. I then added pivot tables that provided information like usage by department over time (see Figure 3), average of the number of licenses in use throughout the day, maximum number of licenses ever used, and extension usage. In using the template, I was able to derive the following statistics about my organization by tracking over an 8-month period of time:

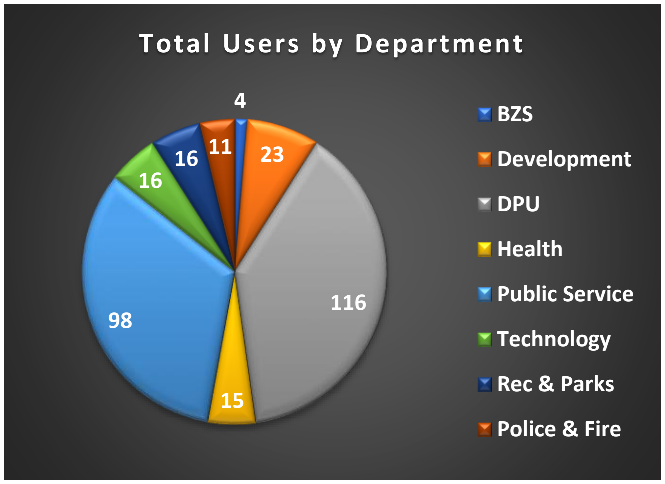


Figure 3. ArcGIS Desktop User Count by Department

- Average license usage = 66 licenses
- Maximum concurrent license usage = 184
- Total hours of usage = 211,419 Hours
- User counts by department
- Extension usage by department

I was also able to compare usage over time and understand when the busy and slow seasons existed within our organization. As Figure 4 shows, GIS is heavily used in the summer and declines around the holidays with the beginning of the year being the slowest.

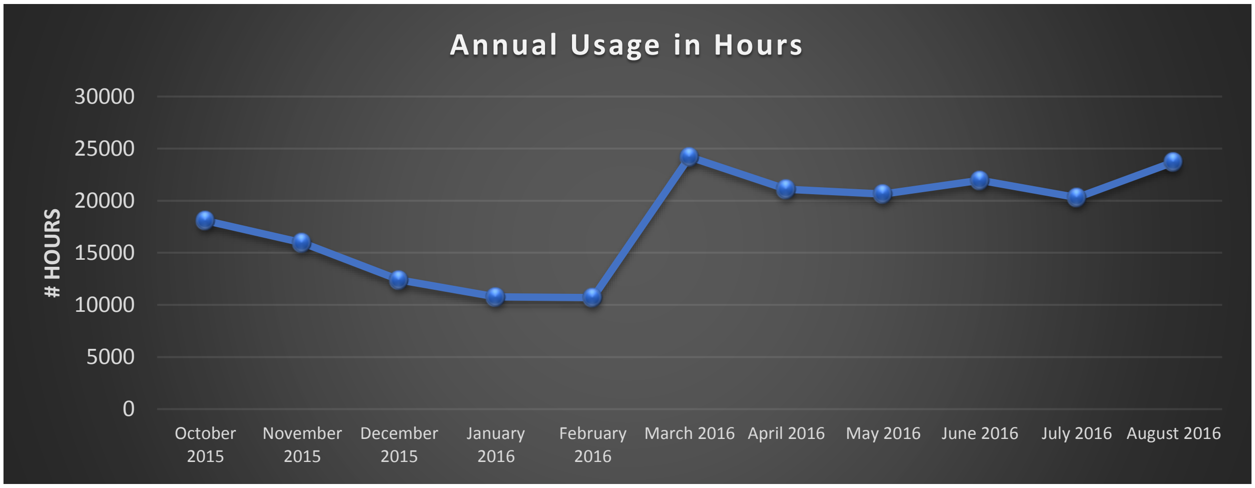


Figure 4. ArcGIS Desktop Usage in hours from October to August

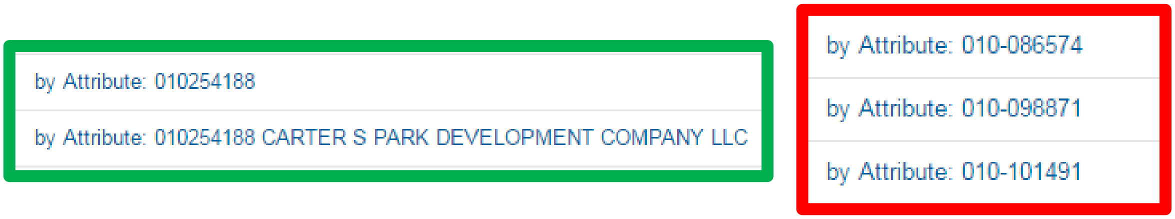
For the action phase I created a report with the completed analysis and provided it to my upper management. It assisted the division in being able to understand our GIS community better. We now know how many users we have per department, how large our GIS Desktop community is, when they work and what components they use. This year we were able to negotiate a better deal with our Esri licensing because we were able to provide details on the level of license that is used (mainly ArcGIS basic) and exactly how many extensions our enterprise consumes. In the future we will also plan to deploy major upgrades or plan outages around the slow season so that we cause as little interruption to our end users as possible.

Web Applications Top

The web application element was geared toward tracking web application usage down to the widget level. For this element I was interested in tracking how many people use an application, what parts of the application they are clicking on, what device they are using to get to the application and overall, highlight what the most/least popular applications were in our organization. The original solution selected was a third-party tool called Heatmaptracker that would actually play back user sessions and give you a heatmap of where users clicked, scrolled or navigated to on your application. However, the tool did not respond to JavaScript actions like a map scrolling in and out or panels opening and closing. It would basically give you the initial image of the GIS application and from there you couldn't tell what the user was clicking on. The alternative solution that could give us the same data to accomplish the tracking goals was Google Analytics. Google Analytics is a hosted solution where you create a login, register your application and are in turn provided an HTML tracking snippet to add to your application code. Event triggers can also be configured to enable widget-level usage by associating a specific named event per widget, which then Google Analytics is able to track.

Once all of my applications were registered I collected data for about 6 months before beginning the analysis phase. Google Analytics has built in, robust visualization capabilities and also allows you to directly export all of the tracking data. I chose the export method because the way I deployed the tracking codes did not allow me to compare different applications or my entire system holistically. Instead, I created a second Excel template where I loaded the data and had pivot tables showing different indicators. From there I was able to understand what my most popular applications were, what widgets were most used and even what search terms people were using when running the search widgets in the applications.

There were a lot of actions that came out of the web tracking, including a couple of unanticipated benefits. The first action that I used this tracking for was an issue we found with one of our locators. We identified that if end users used a dash in the parcel id look-up instead of all numerical characters that the locator would not return any results. I was able to export the search terms being used to understand how many users were experiencing this issue. Figure 5 shows a couple examples of what I found.



by Attribute: 010254188

by Attribute: 010254188 CARTER S PARK DEVELOPMENT COMPANY LLC

by Attribute: 010-086574

by Attribute: 010-098871

by Attribute: 010-101491

Figure 5. Correct address search on left and search examples on the right that would yield no results

I found that more than half of search strings submitted were using a dash. Since the problem was affecting half of the searches completed, we updated the widget to have a large warning message instructing users to not use a dash. From there we were able to monitor as users started to use the dash less and less.

A second, unintended way I used this tracking was with the widget-level usage. We purchased and implemented the Pictometry widget during this project, expecting it to be a hit in our applications. We even gave it a special location separate from our other widgets, but in watching the event trigger for the widget found that it was rarely used. We thought maybe users were not seeing it since it was in a new location where widgets were typically not found, so we gave it a brighter background color to stand out. After the color change the widget usage increased as end users could find the widget and start incorporating it into their workflows.

The third surprise use was for when we developed a new version of an application, using tracking to understand when the old application should be decommissioned. Typically, departments want a grace period where both a new and old version of an application can both be running while they train users and get their teams comfortable with the new way of completing a workflow or task. Google Analytics assisted in being able to give weekly updates on the progress of the end-user migration and helped us confidently decommission applications when we knew the majority of users had transitioned. Figure 6 is an example of one application where we provided these analytics on a weekly basis.

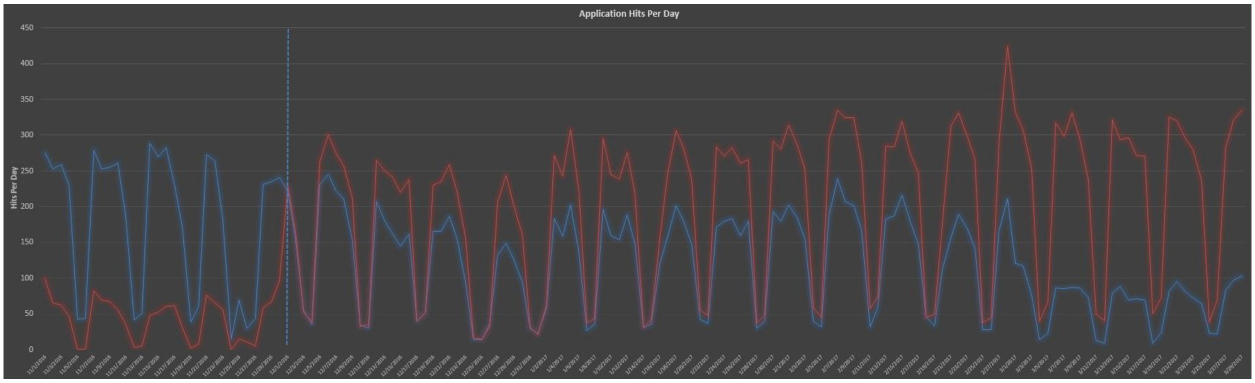


Figure 6. Graph showing 2 versions of an application running concurrently and the change in usage as end users migrate. The dotted line represents the release announcement of the new version.

Besides those surprise uses, we used the tracking from Google Analytics to accomplish the original goal of understanding our applications and widget usage. We put together a list of our applications from most to least popular which helps my division prioritize which applications to focus on enhancing or promoting within our organization and which we should consider decommissioning. In the future, we also plan to use these numbers for capacity planning and understanding how many more applications and demand can be placed on our current infrastructure. Figure 7 shows the top seven applications in the organization and their daily hit count. If we can measure the load on our infrastructure that each hit generates, then we can estimate the impact a new application will have.

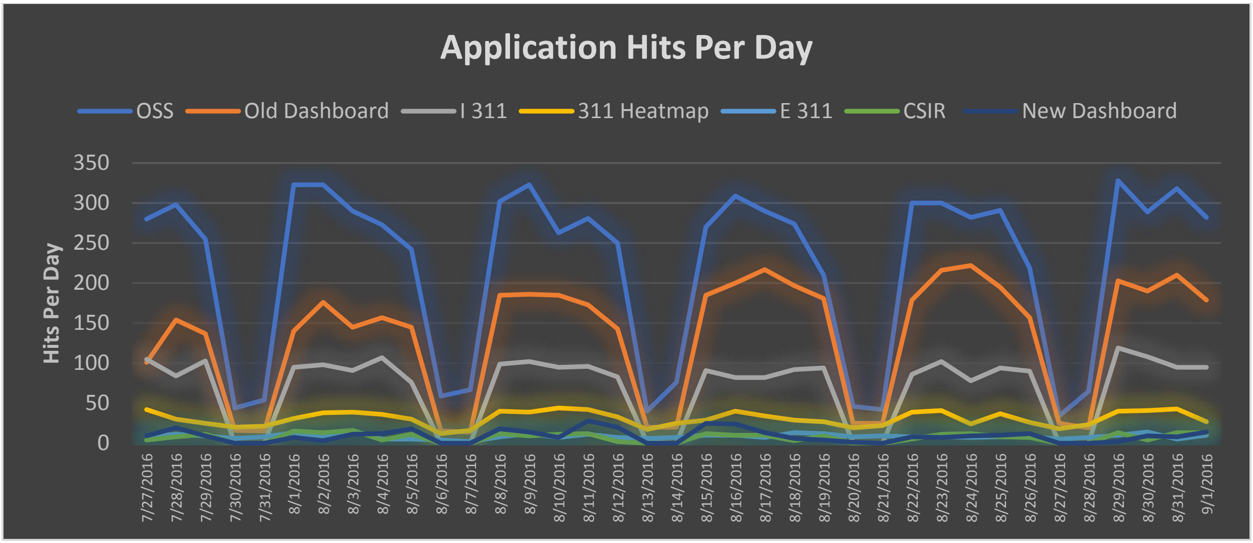


Figure 7. Most popular applications usage over a week period

ArcGIS Server

Top

The third tracking element, ArcGIS Server, involved implementing and configuring Esri's System Monitor (SM). SM can be configured to track individual map services in the ArcGIS Server environment and was implemented to track all eight production ArcGIS servers in my organization. SM is recommended by Esri to run on a staging server because it does consume a considerable amount of resources. It consists of both a web application that provides a graphical user interface (GUI) to configure tracking and an agent service that runs and collects different log files from ArcGIS Server. SM stores all of the log file and analytic data in a MongoDB that can be on the same staging server or in a different location. The servers to be tracked also required a separate client installation and a couple of small configurations to be put in place so the System Monitor server can connect successfully and pull log files.

The analysis phase was very simple for this tracking element because out-of-the-box SM has a lot of pre-canned queries tied to reports and charts that enables easy access and understanding of the tracking data. The results include the following which you can also see in Figure 8:

- Map service throughput (how many requests processed per second)
- Map service busy times per second
- Map service total requests
- Overall ArcGIS Server requests

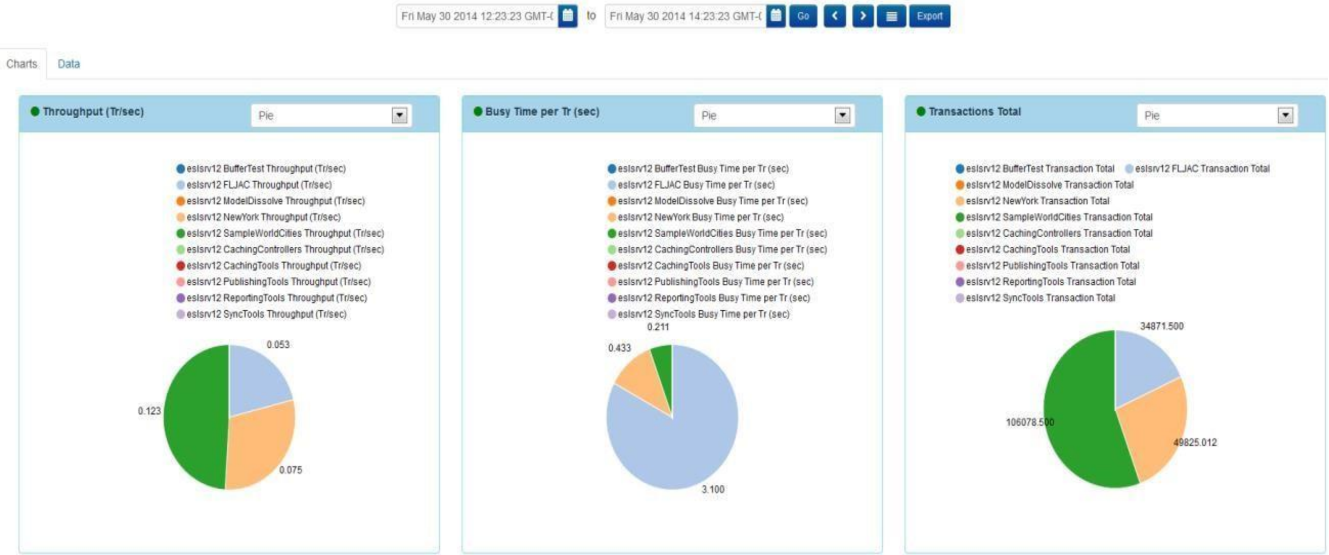

Fri May 30 2014 12:23:23 GMT-(
to
Fri May 30 2014 14:23:23 GMT-(
Go
Export
Charts
Data
Throughput (Tr/sec)
Pie
eslsrv12 BufferTest Throughput (Tr/sec)
eslsrv12 FLJAC Throughput (Tr/sec)
eslsrv12 ModelDissolve Throughput (Tr/sec)
eslsrv12 NewYork Throughput (Tr/sec)
eslsrv12 SampleWorldCities Throughput (Tr/sec)
eslsrv12 CachingControllers Throughput (Tr/sec)
eslsrv12 CachingTools Throughput (Tr/sec)
eslsrv12 PublishingTools Throughput (Tr/sec)
eslsrv12 ReportingTools Throughput (Tr/sec)
eslsrv12 SyncTools Throughput (Tr/sec)
0.053
0.123
0.075
Busy Time per Tr (sec)
Pie
eslsrv12 BufferTest Busy Time per Tr (sec)
eslsrv12 FLJAC Busy Time per Tr (sec)
eslsrv12 ModelDissolve Busy Time per Tr (sec)
eslsrv12 NewYork Busy Time per Tr (sec)
eslsrv12 SampleWorldCities Busy Time per Tr (sec)
eslsrv12 CachingControllers Busy Time per Tr (sec)
eslsrv12 CachingTools Busy Time per Tr (sec)
eslsrv12 PublishingTools Busy Time per Tr (sec)
eslsrv12 ReportingTools Busy Time per Tr (sec)
eslsrv12 SyncTools Busy Time per Tr (sec)
0.211
0.433
3.100
Transactions Total
Pie
eslsrv12 BufferTest Transaction Total
eslsrv12 FLJAC Transaction Total
eslsrv12 ModelDissolve Transaction Total
eslsrv12 NewYork Transaction Total
eslsrv12 SampleWorldCities Transaction Total
eslsrv12 CachingControllers Transaction Total
eslsrv12 CachingTools Transaction Total
eslsrv12 PublishingTools Transaction Total
eslsrv12 ReportingTools Transaction Total
eslsrv12 SyncTools Transaction Total
34871.500
106078.500
49825.012

Figure 8. Esri's System Monitor analytics for one ArcGIS Server

I learned a lot about how the load was distributed on our servers and which map services were popular and unpopular.

The actions were fairly simple, mainly focused on optimizing the services depending on their usage. For services that were heavily used, I increased their minimum number of instances (resources) allocated to them so that their response time would be faster. Then for services that were not used often, I allocated them no running resources. Through this tracking I also found that one ArcGIS Server was carrying a much larger load than the others as you can see in Figure 9. The second action that I took included migrating some of those services to other machines so that the overall load could be more evenly distributed across the environment. In the future we will use this tracking in conjunction with Esri’s capacity planning tool to understand how many more services can be added to our environment before we need to budget for a new ArcGIS Server.

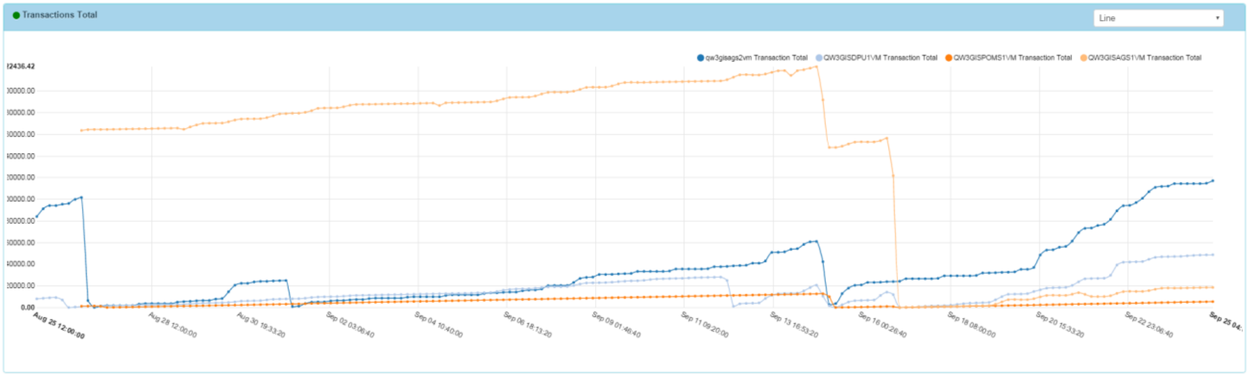
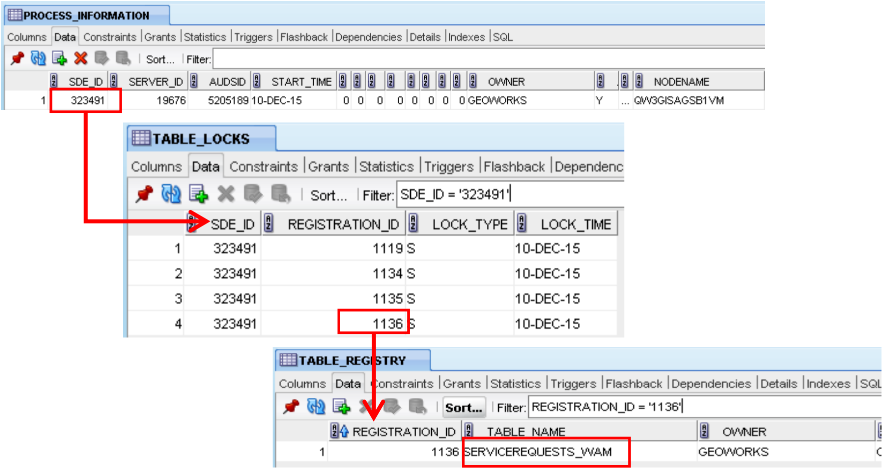


Figure 9. Transaction total across all ArcGIS Server machines

Geodatabase [Top]

The final tracking element was geodatabase usage and more specifically which feature classes are being used within the database and by whom. This tracking element also did not have an out-of-the-box solution, so I created and deployed a database trigger that wrote to a log table in the database. The solution works by running the trigger anytime there is a new record in the PROCESS_INFORMATION SDE table, which happens when a user first opens ArcMap and adds a table or has tables already loaded in a saved map document. The trigger executes a query that joins three tables within the database, the PROCESS_INFORMATION, TABLE_LOCKS, and TABLE_REGISTRY tables and reports the join results to the log table. By joining those three tables you are able to get the user making the connection and the tables that they have loaded in ArcMap. Figure 10 details an example of the join that ultimately shows the GEOWORKS user is using the SERVICEREQUESTS_WAM feature class.



PROCESS_INFORMATION

	SDE_ID	SERVER_ID	AUDSID	START_TIME									OWNER			NODENAME
1	323491	19676	5205189	10-DEC-15	0	0	0	0	0	0	0	0	GEOWORKS	Y	...	QW3GISAGSB1VM

TABLE_LOCKS

Filter: SDE_ID = '323491'

	SDE_ID	REGISTRATION_ID	LOCK_TYPE	LOCK_TIME
1	323491	1119	S	10-DEC-15
2	323491	1134	S	10-DEC-15
3	323491	1135	S	10-DEC-15
4	323491	1136	S	10-DEC-15

TABLE_REGISTRY

Filter: REGISTRATION_ID = '1136'

	REGISTRATION_ID	TABLE_NAME	OWNER
1	1136	SERVICEREQUESTS_WAM	GEOWORKS

Figure 10. Example of database trigger action to identify user and table

The analysis for this tracking was fairly simple. I loaded the contents of the tracking log table to a third Excel template and created pivot tables from there to provide the following insights:

- List inactive/active tables
- List most popular tables
- Exact list of who used what when
- Timestamp any FC was last used
- List of FC usage by Department
- Show database usage over time
- Cross reference with License Log by Computer Name to get ArcReader users/Single Use license users. *Since ArcReader users do not use the license manager, but do access the database, this tracking is a way to pick up those users

The geodatabase tracking actions also took a different direction than initially intended and proved to be the most frequently used tracking tool by my division. The intended actions that we followed through with included database optimization. With a list of non-used tables we were able to confidently archive and clear out a lot of space in the database. We also worked with the most active list and our database administrator to optimize the performance of those feature classes by running Analyze Datasets and updating their statistics more frequently. The third intended action that we followed through with was to create a layer directory of the frequently used feature classes to provide easier access to our end users. The layer files were pre-symbolized as well as field aliased so end users don't have to know much about the data structure to pull it into ArcMap and start interacting with it.

The major unexpected action that my division has taken as a result of geodatabase tracking has been with troubleshooting. With close to 350 datasets, we sometimes have problems with accessing a dataset. For example, a feature class may have been corrupted due to a data migration error. When these types of problems occur, we are able to query the table log to see who frequently uses the table having issues and email them directly to let them know of the problem. This targeted notification has helped us communicate more efficiently and effectively and has eliminated unnecessary confusion the end users may have had when using ArcMap and a layer is not functioning correctly. On the flip side, now when we plan for an outage or have scheduled maintenance on a feature class, we also notify the recent users so they can plan around the outage or report immediately if a maintenance action we took negatively affected the dataset.

Outcomes

Top

Once all of the solutions and documentation were developed for the project I moved on to creating the wiki page which can be found here. I documented the purpose of the project and detailed descriptions of each element, walking the reader through the deployment to the analysis to the actions that could be taken. I also linked all of the solution downloads to GitHub in case people enhanced any of the built-from-scratch tools then I had a mechanism to incorporate those changes.

I was also fortunate enough to present at two GIS conferences. The first was at the 2016 Ohio GIS Conference held in Columbus. The second presentation was in Jacksonville at the 2017 URISA GIS-Pro conference. Both had a great turnout and were well received by the attendees.

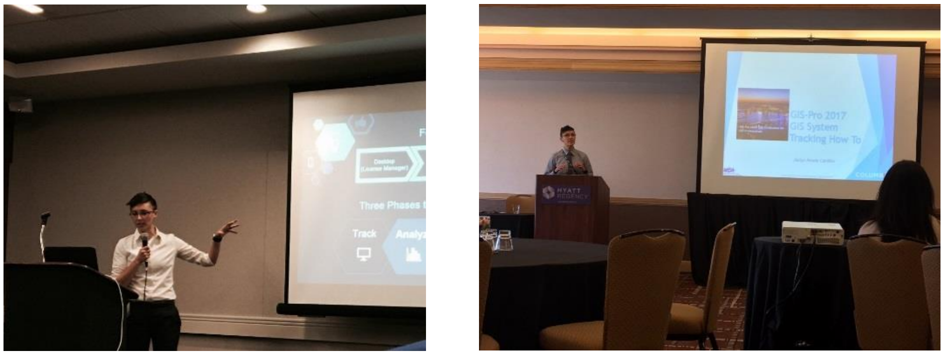


Figure 11. Project Presentations

Now that tracking is in place in my organization, I see it as an ongoing process of continuing to collect usage data, analyze it and take actions to improve our enterprise deployment. During this project there were a few surprise uses we benefitted from in addition to the insight we expected to gain. The knowledge and data gained (and continued to be gained) by this project has been invaluable to my organization and has provided necessary metrics that are used to make better decisions about our GIS infrastructure, the services we provide and applications we develop.

Resources

1. Capacity Planning Tool: http://www.wiki.gis.com/wiki/index.php/Capacity_Planning_Tool
2. JMeter: http://jmeter.apache.org/
3. System Test: http://www.arcgis.com/home/item.html?id=e8bac3559fd64352b799b6adf5721d81
4. System Monitor: http://www.arcgis.com/home/item.html?id=848f48b0f88e4de7a036377197453efe
5. Esri Professional Services Resources *Check to Show ArcGIS Desktop Content: http://www.arcgis.com/home/search.html?q=owner%3AEnterpriseImp&restrict=true
6. Vestra: http://products.vestra.com/geosystems-monitor-features/
7. Latitude Geographics, Geocortex Insight: http://go.geocortex.com/get-insight

References

EnterpriseImp. (2015, July 4). *System Monitor (1.1.6).* Retrieved November 15, 2015, from http://www.arcgis.com/home/item.html?id=848f48b0f88e4de7a036377197453efe

Geocortex. (n.d.). *GeoSystems Monitor Features | VESTRA GIS.* Retrieved November 18, 2015, from http://products.vestra.com/geosystems-monitor-features/

Sakowicz, A. (2013, March 25). *ArcGIS Server Performance and Scalability – Testing Methodologies*. Retrieved November 12, 2015, from http://proceedings.Esri.com/library/userconf/devsummit13/papers/devsummit-021.pdf

Vestra. (n.d.). *GeoSystems Monitor Features | VESTRA GIS.* Retrieved November 18, 2015, from http://products.vestra.com/geosystems-monitor-features/