

Loggly in Plain English Video Script

core value props to remember all the time

- cost effective log storage solution
 - lower maintenance fees
 - lower storage costs
 - reliability
 - retention control
- log file aggregation
 - all logs in one place
 - user access control and availability
- log file search
 - lower development costs through realtime app debugging
 - lower MTTR on application failures
 - proactive alerting to increase availability



overview (starts here, read out loud)

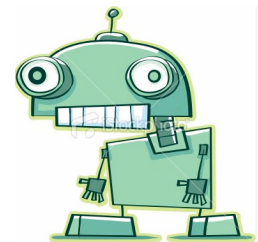
- logs are everywhere around you - no, not those logs (showing some tree logs) - log files - computer log files



- most computers generate logs - at home, in a colo, in the cloud, and yes - even in robots [log falls out of the robot]

- different types of software generates different types of logs. information in these logs includes stuff like all the people that emailed you yesterday,

security breaches in your router, and what IP addresses are accessing your website



- web servers, in particular, are important because they serve up applications like web mail, photo albums, or your favorite social networking site

- some people call the way these applications are delivered SaaS, which means software as a service. we just call them 'web apps', or 'apps', for short. we're cool that way.



market scope

- apps are all around us - salesforce, facebook, flickr, twitter, and thousands of other sites are web apps - all running on servers connected to the internet. lots of servers.

- web apps are a critical part of a company's revenue model and a lot of money is on the line when it comes to writing and running them.

- there are 10s of thousands of companies writing web apps today. in fact, there are so many, over five thousand of those web apps are written specifically for the iPhone. that's a lot of apps!
- there are so many web apps out there, over 10 programming languages and hundreds of web app frameworks exist for developing them; developers like writing their own frameworks for some reason; those guys are so hardcore

problem statement

- when someone uses a web app, data is communicated between the server and the client machine and some of that interaction is stored in a log file on that server



- to better illustrate, we'll take a look at a familiar photo sharing web app - flickr. everyone loves using flickr in their web coding examples. we love flickr too!



- on flickr, people look at photos. in fact, they look at a LOT of photos. serving so many photos requires lots of servers. LOTS of looking. LOTS of servers.

- every time someone looks at a photo, the web server serving the photo makes a short entry in it's own local log file.

- this entry looks something like this: a timestamp - when the event happened - the URL - what photo was looked at - size - how big it was - and ip address - who looked at it.

- this ends up happening on ALL the servers flickr runs - 24 hours a day, 7 days a week. even on national smores day. look it up. it's real people.

- anyway, flickr has lots of servers and lots of logs. LOTS.

- flickr also writes out other things in their logs, like how long it took to load a page, whether errors were encountered on the page, and crazy technical things like how long it took to run the database query for finding all the funny albanian cats photos uploaded last july.



- because of all this detailed logging, these log files get huge. HUGE. well, you get the point.

~~- sometimes log files get so big they fill up hard drives. this happens a lot to sys admins. it's a big problem. people get fired for this. sometimes they do. maybe.~~

- in short, it's a HUGE challenge to manage these log files. it's a pain to look across multiple apps and servers, and once you find the correct log file, an even BIGGER pain to try to search through them for issues.

- developers need to have the logs there in case something goes wrong with their software - like paypal barfing on a payment transaction when the prince of Morocco tries to upgrade his flickr account. boy that sucked.



- one of facebook's developers, at least we think it was facebook....anyway this guy, a developer, Carl Snavley, told us with a snort, "the process of retention and use of logs for troubleshooting and proactive maintenance is a critical part of our application availability strategy and is key in our ability to disambiguate interactive metadata!"

- let me translate. carl's saying "i looked in the logs and figured out why the server crashed this morning. i'm fixing my code now and it shouldn't happen again. maybe." ~~what a geek.~~

- writing and maintaining great web apps is demanding. companies like yahoo and google spend billions of yen making sure their logs are centrally located, backed up, and accessible to their developers so they can make great apps.



- most smaller web app companies don't have the time to develop a solution to their logging problem because they are too busy writing great software for their users. they didn't consider it to be a problem to solve in advance. they figured they would deal with it later.

- however, just like the good people at google and yahoo, they still care about making their apps reliable. they have to, or they'd go out of business. ~~it's later guys.~~

- most companies usually resort to doing things the old school way. spending lots of time digging around in logs.



-when a developer pushes new code to the servers that run his app he has to go from server to server, from log file to log file, looking for something - anything - in the logs that may indicate a problem is occurring on the production site.

-this process happens all the time, and this debugging process ends up wasting a bunch of time and money - and unfortunately - affects

the quality of a company's apps in the long run if not done right.

solution to the problem

- Loggly is a web app built from the ground up to do a really good job of storing, retaining, searching, alerting, and reporting on log files to increase web application availability and providing insight into a company's web app.

- instead of keeping its application logs locally, a company sends logs to Loggly to manage. this saves them time and money because they don't have to run the servers or pay for storage costs.

- to start using Loggly, a customer signs up for an account, and then makes some small changes to their logging setup to 'forward' the logs to the Loggly servers. these logs can be sent in via web POSTs or syslog. it's easy.

- once the logs arrive at Loggly, they are broken down into chunks by time, and then indexed. the indexes make the logs searchable later.

- Loggly takes care of the log file retention policies for the app too. retention policies allow a company to define exactly how long to keep a log file around before it is sent off to the big trash can in the sky.



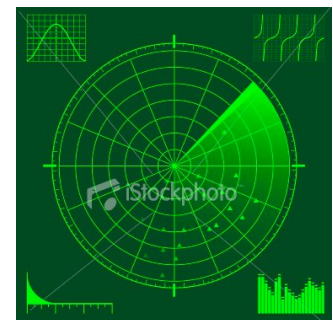
- Loggly utilizes fancy elastic cloud computing technologies, so it's easy for us to use as many servers as needed to complete a task - like when your server gets on Reddit, Slashdot and Digg all at once. We handle the log volume in an elastic way.

- users end up paying Loggly for the resources they use on the system, including bandwidth, storage space, and computing time.

- once the logs are stored in Loggly, a developer - or sys admin - can use the interface to quickly search for issues in his logs. The user can search his logs just like people use google to search for "how to survive a raptor attack". [asked off mic] Seriously?

- at the same time, the developer can watch - we call it "tail" - all his log files for new events - even from multiple servers and files, all right from his web browser. it's like a radar for your logs.

- Loggly's search interface looks and feels exactly like a server's command line interface, which is both familiar and functional for the developer to use.





-the interface provides high data density - literally filling the screen with log data, and allows you to filter by time, server, log type, keywords and more.

-by bringing all the logs together from multiple machines and sources, the developer can quickly find correlated events which may have contributed to an issue

-this lowers the time spent finding problems, increases reliability and quality of the web app, and saves the company tons of dough letting Loggly deal with their logs.

- because we're total web hipsters, Loggly provides a rich RESTful based API around the service. the API allows you to combine queries to your logs with other services.

- developers on the internet like to call these mashups. mashups are big business. developers love to create innovative apps doing goodness knows what.

- Loggly mashups could be used to create a trending graph of users, security monitoring dashboards to stop hackers, and dynamically updated things like a shiny spinning globe that will wow even the most hardened product manager. well, most of them.



- these types of mashups could provide instrumentation and VISIBILITY into a company's app ... into to it's business model. This translates into bigger business, and money, for the company

- we're working tirelessly on an alpha version of Loggly. if you are interested in learning more about the product, company, founders, or attacking raptors, give us a shout. thanks for listening. see you in your logs.

The Loggly Team