Use cases and expectations:

- Convenient, on-a-whim data mining
- Ease of setting up dashboards
- Multiple users

Development times:

		Oracle & Python/Matplotlib	Splunk
Batch stats	CPU time	$2^1/_2$ days	2 hours
	Jobs	12 hours	5 minutes
	Waiting time	3 hours	1 minute
	Cumulative waiting time	3 hours	$^1/_2$ hour
Job info	Job info, related jobs, job error analysis	2 days	2 days
	Search by user name	Never got this far	10 minutes

Views on Splunk:

	What I like	What I dislike	Comment
Dashboards	Quick to set up simple ones	XML markup is clumsy	Wrote XML generator script
Multiuser	Authentication, authorisation & SSO for free	Free queries by default	Free queries are Splunk's strength
Storage	High access throughput to the data	$\varnothing$	What about non-sequential data, though?
Collection	Flexibility	Setting up Forwarders	Always managed without anyway
Querying	Interactive, very powerful	Not always obvious	Simple queries easily achievable
Doc	Complete, there are tutorials	Can't see the wood for the trees	I probably just need more time to read

LSF • Job Info for 291013757

Search by job ID:

Try for instance 270697568, or 270692821, or even 270697290. And here's a shoddy one: 270698221.

Or just search by user name.

Information for job 291013757

Index	0
User	jbl
Queue	8nm
LSF Code	
Job Status	

Other jobs from user jbl

	eventTime	jobId	queue
1	24 Aug 2012 16:17:50	291013757	8nm
	24 Aug 2012 16:17:21		8nm
	24 Aug 2012 15:38:03		
10	24 Aug 2012 15:32:50	290995[illegible]	

Other jobs in queue 8nm

	eventTime	jobId	userName
1	24 Aug 2012 16:17:50	291013757	jbl
2	24 Aug 2012 16:17:21		jbl
	[illegible]		[illegible]
	24 Aug 2012 16:13:24	28784[illegible]	

splunk> Batch Historical Usage

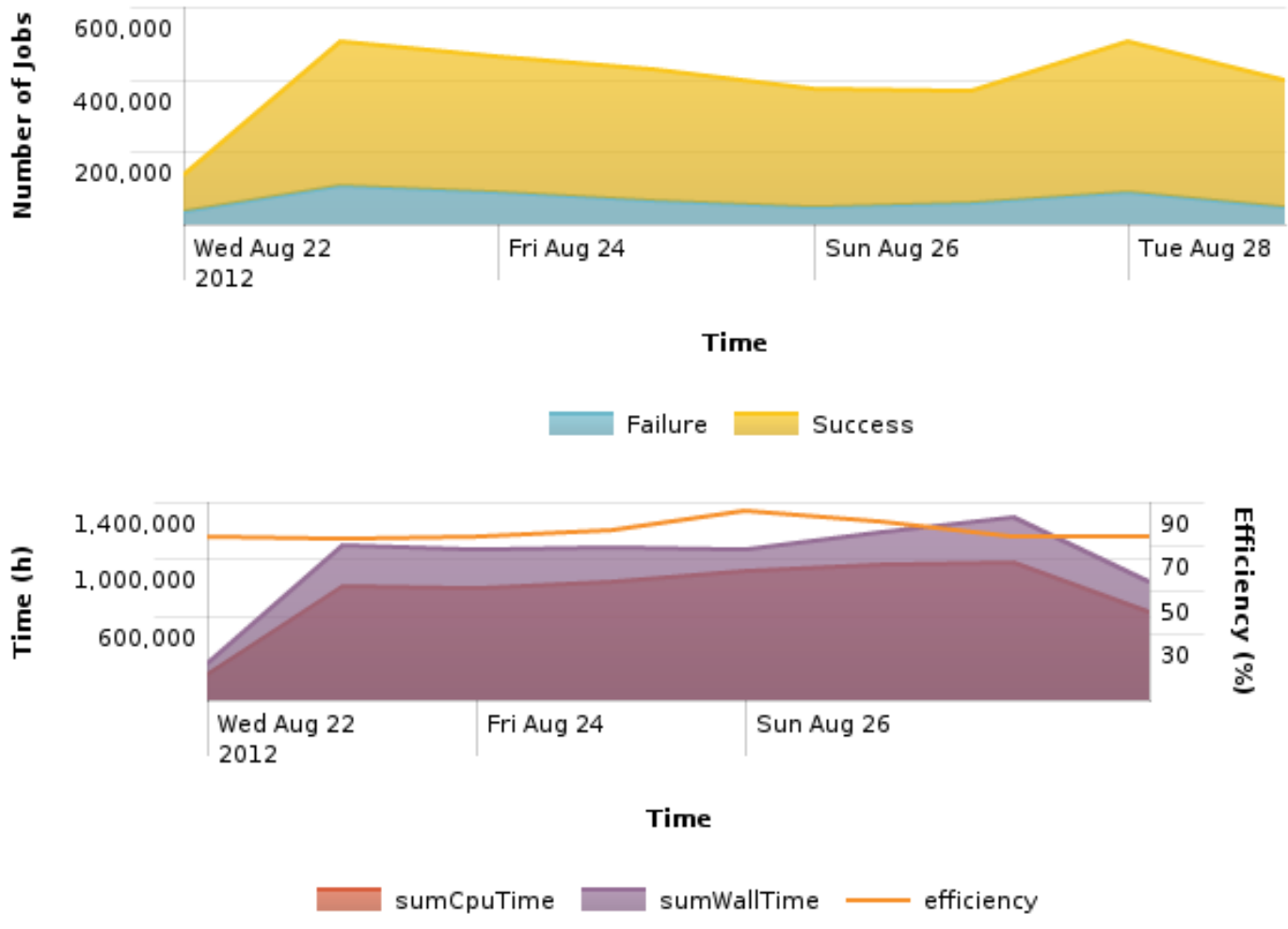
Administrator | App | Manager | Alerts | Jobs

Home Groups Histograms

Help | About

Home

In the Last Week



Splunk and the future of batch service monitoring:

- Historical Usage
- Consumer-side monitoring
- Existing batch monitoring to Splunk
- Even more advanced dashboards