Software Usage Collection and Analysis with ELK

Andrew Caird

acaird@umich.edu

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

All of this software! Why?

CAEN supports 100s of software packages in its student computing labs and in its HPC environment, and the list grows every year.
To help us make decisions about software usage patterns, we wanted to collect and analyze data on a per-title, per-computer, per-user basis.

Our Goals

By gathering software usage information for everything that CAEN installs, we hope to:

- get a better understanding of what software is used, so we can remove titles that we don't need to be installing
- correlate titles with courses, and try to suggest that similar courses use the same software to minimize what the students have to learn, the instructors have to teach, and CAEN has to support
- make purchasing and labor decisions based on more than just an educated guess

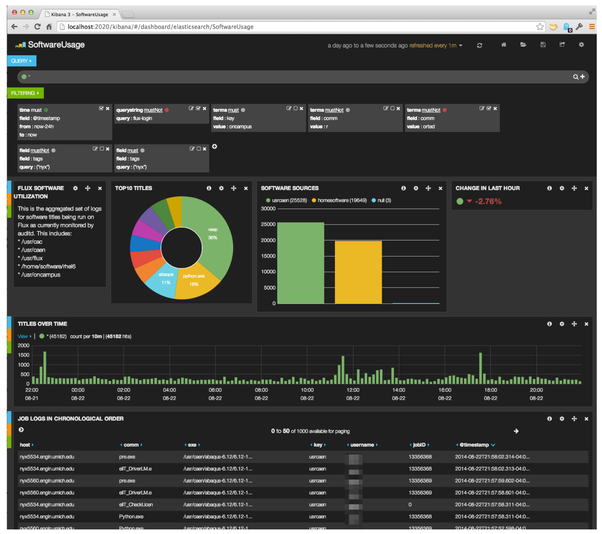
ELK

The open-source ELK stack (`http://www.elasticsearch.org/overview/`) allows us to process, aggregate, store, search, and analyze logs with a lot of metadata from Windows and Linux computers.

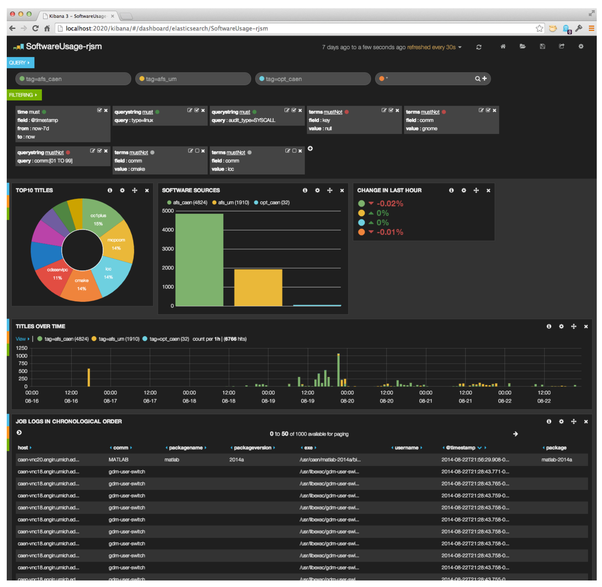
- **E**lasticsearch is based on Apache Lucene and is a very fast, distributed real-time search and analytics engine that offers a rich query DSL via a REST API
- **L**ogstash converts logs to JSON according to your rules and ships them to Elasticsearch
- **K**ibana is a real-time visualization engine for Elasticsearch

We also use the Beaver Python daemon (`http://beaver.readthedocs.org/`) to ship logs to Logstash from the Linux Lab computers.
Elasticsearch also allows us to modify log entries after the fact, so we are going to insert demographic and role data to augment the `username` field. This data will include things like class standing (Freshman, Sophomore, etc.), department, course enrollment, and affiliation (Alumni, Student, Staff, etc.).

HPC Cluster Software Logs



Lab Linux Software Logs



REST Interface

ElasticSearch has a rich REST interface that supports queries on date ranges, terms, and aggregations. The programming interface for custom queries or output is straightforward.

```
curl -XGET 'http://localhost:9200/\
  flux-software-*/_search?pretty' -d '{
 "aggs": {
  "comm_users": {
   "terms": {
    "field": "comm",
    "size": 10,
    "order": { "top_users" : "desc" }
   },
   "aggs": {
    "top_users": {
     "cardinality": {
      "field": "username"
     }
    },
    "users": {
     "terms": {
      "field": "username",
      "size": 5
     }
    }
   }
  }
 }
}'
```

returns the top 10 most frequently launched programs and the usernames of the top 5 users who launched each of them:

```
 "key" : "matlab",
 "doc_count" : 16301,
 "users" : {
   "buckets" : [ {
     "key" : "RedactedUser01",
     "doc_count" : 3491
   }, {
     "key" : "RedactedUser02",
     "doc_count" : 3330
   }, {
     "key" : "RedactedUser03",
     "doc_count" : 966
   }, {
     "key" : "RedactedUser04",
     "doc_count" : 669
   }, {
     "key" : "RedactedUser05",
     "doc_count" : 651
   } ]
 },
 "top_users" : {
   "value" : 78
 }
```

Our ELK Environment

- Two node ElasticSearch cluster and one Kibana web server
- About 1,000 Linux Logstash clients logging directly to Elasticsearch
- About 1,000 Linux Beaver clients sending logs to one Logstash client for insertion into Elasticsearch
- About 1,000 Windows Logstash clients logging to one Logstash client for insertion into Elasticsearch

Linux S/W Logging Config

The Linux config is `auditd` with `/etc/audit/audit.rules` containing lines similar to these:

```
-a always,exit -F dir=/usr/cac/  -F perm=x -k usrcac
-a always,exit -F dir=/usr/caen/ -F perm=x -k usrcaen
```

The LogStash configuration contains lines similar to these:

```
input {
 file {
  path => [ "/var/log/audit/audit.log" ]
  type => fluxsoftware
 }
}
```

Windows S/W Logging Config

In the Local Group Policy, enable **Audit Process Creation on Success** and **Audit Process Terminiation on Success**
The `input` section in the LogStash configuration is:

```
input {
 eventlog {
  logfile =>  ["Application", "Security", \
               "System"]
  type => "winevent"
 }
}
```

References

[1] Radu Gheorghe and Matthew Lee Hinman, *Elasticsearch in Action* `http://www.manning.com/hinman/`, Manning, early 2015
[2] Clinton Gormley and Zachary Tong *Elasticsearch: The Definitive Guide*, `http://www.oreilly.com/catalog/0636920028505`, O'Reilly Media, March 2014
[3] *The ELK Stack*, `http://www.elasticsearch.org/`
[4] *ELK at CAEN*, `http://caen.github.io/elk`

Thanks

Chris Berger, Matt Britt, Katelyn Findlay, Nicole Heffernan, Paul Killey, Tom Knox, Don Lambert, Dan Maletta, Steve Mattson, Ross Smith, and the rest of the CAEN staff.