Sběr záznamů a událostí využívající virtualizační služby MetaCentra

Radoslav Bodó, Daniel Kouřil
CESNET

Konference Internet a Technologie 13.2

Agenda

- Úvod
- Sběr dat
- Zpracování logů
- Analýza
- Závěr

Úvod

- Současný stav a prostředí
  - Národní gridová infrastruktura -- MetaCentrum.cz
    - přibližně 700 výpočetních uzlů
    - web servery
    - služební stroje a služby

- Motivace
  - centrální logování
    - bezpečnost
    - provoz

Cíle

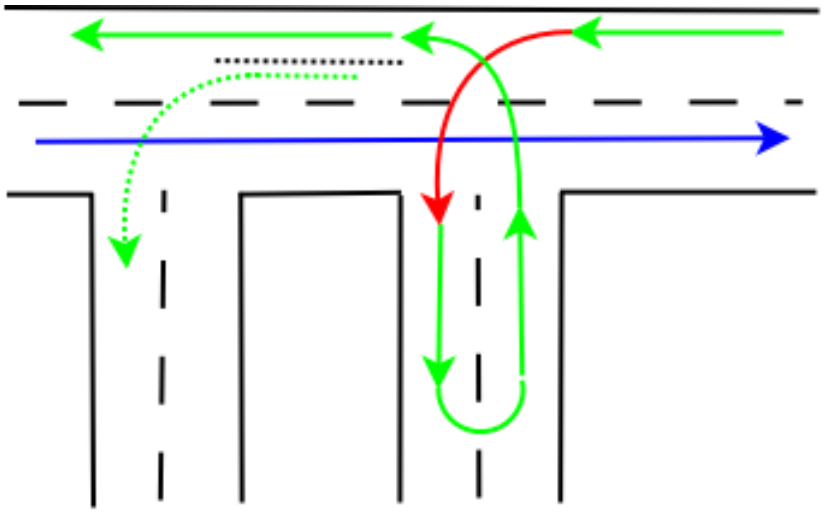
- zabezpečené a spolehlivé doručování
  - šifrovaný, autentizovaný kanál
- škálování
  - systém schopný pojmout mnoho logů
  - škálování nahoru, ale i dolů
- pružnost
  - systém který umí zpracovat “jakákoliv” data ...

Sběr dat

- linux + logování = syslog
  - přeposílání logů přes syslog protokol
    - UDP, TCP, RELP
    - TLS, GSS-API

- NGI Metacentrum
  - prostředí založené na GNU Linux/Debian
  - autentizační mechanismy založené na sys. Kerberos
    - přeposílání logů z výpočetních uzlů přes kanál chráněný GSS-API

rsyslogd shipper



- **omgssapi.so -- klient**
  - přeposílání (forwarding) je akce (action)
    - fronta akce (actionqueue) musí být nepřímá (non-direct)
    - fronta musí být omezena
    - plná fronta nesmí zablokovat hlavní frontu

```
$ActionQueueType LinkedList          # use asynchronous processing
$ActionQueueFileName srvrfwd1        # set file name, also enables disk mode
$ActionResumeRetryCount -1           # infinite retries on insert failure
$ActionQueueSaveOnShutdown on        # save in-memory data if rsyslog shuts down
$ActionQueueMaxDiskSpace 100m        # limit disk cache
$ActionQueueTimeoutEnqueue 100       # dont block worker indefinitely when cache fills up
*.* :omgssapi:<server_name>:<port>   # deliver all messages to central server using GSS-API protection
```

rsyslogd server

- **imgssapi.so -- server**
  - nic zvláštního
    - listener
    - logování do adresářů per IP
    - logování do adresářů per služba

```
$ModLoad imgssapi
$InputGSSServerServiceName host
$InputGSSServerPermitPlainTCP off
$InputGSSServerRun 515
$InputGSSServerMaxSessions 2000
```

```
$template PerHostLogsSyslog,"/var/log/hosts/%$YEAR%/%$MONTH%/%fromhost-ip%/syslog"
$template PerHostLogsAuthlog,"/var/log/hosts/%$YEAR%/%$MONTH%/%fromhost-ip%/auth.log"
$template PerHostLogsKernlog,"/var/log/hosts/%$YEAR%/%$MONTH%/%fromhost-ip%/kern.log"

auth.*,authpriv.* -?PerHostLogsAuthlog
kern.* -?PerHostLogsKernlog
*.*;kern,auth,authpriv.none -?PerHostLogsSyslog

$template PerServiceLogsSyslog,"/var/log/hosts/auth/%$YEAR%/%$MONTH%/auth.log.%$YEAR%%$MONTH%%$DAY%"
auth.*,authpriv.* -?PerServiceLogsSyslog

$template PbsService,"/var/log/hosts/pbs/%$YEAR%/%$MONTH%/log.%$YEAR%%$MONTH%%$DAY%"
:programname, contains, "pbs_mom" -?PbsService
```

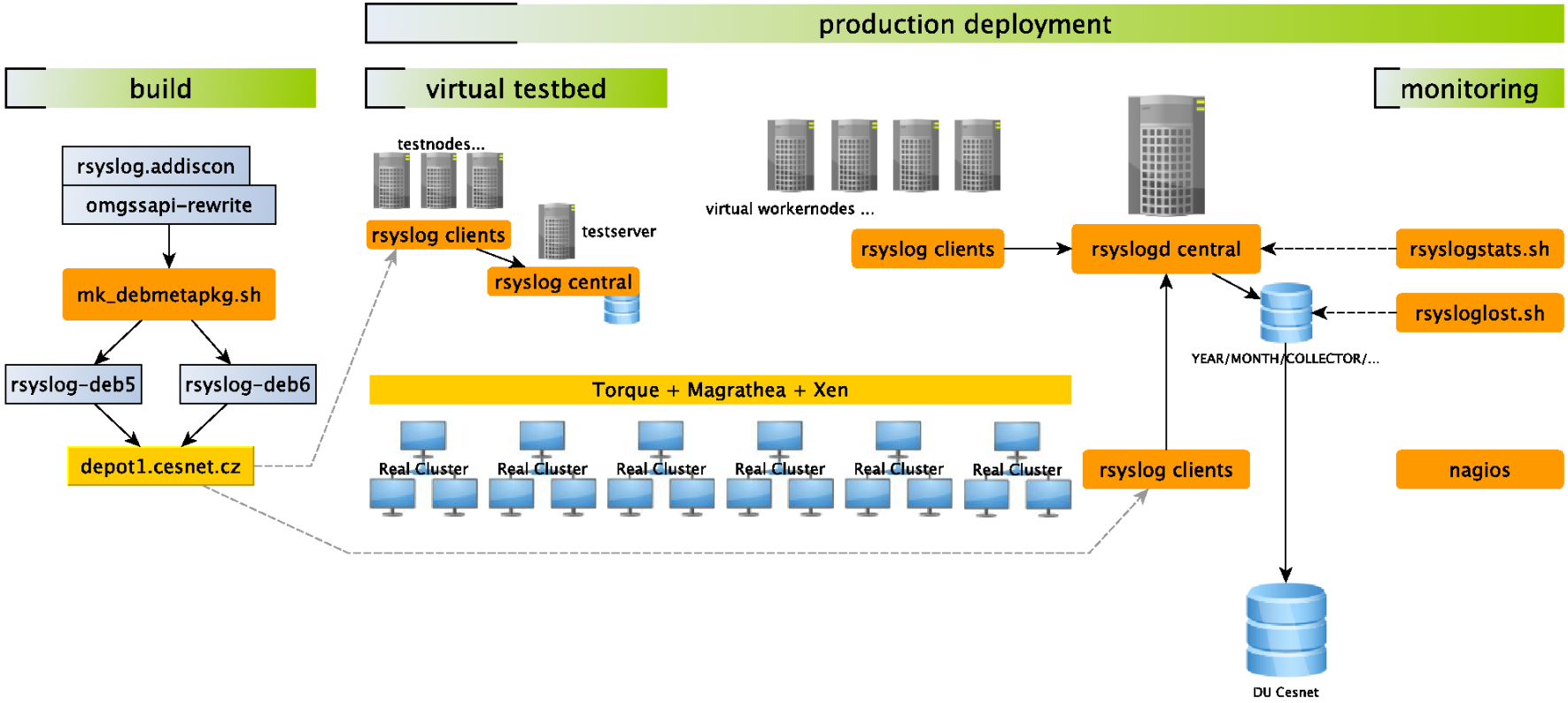
rsyslogd GSS záplaty

- původní GSS-API plugin není od verze 3.x udržován
  - plugin nereflektuje změny v jádře rsyslogu >> umí to velmi nehezky spadnout
    - snažili jsme se to opravit ve spolupráci s upstreamem
      - již to nepadá ale generuje to bouře SYN paketů (v5,v6,?v7, ?v8)

- nový plugin omgssapi založený na
  - starém pluginu + současném omfwd (tcp forward)
  - uveřejněno pro public domain, ale do hlavního stromu zatím nepřijato
    - zkusíme to znovu pro v7, nebo dnes spíše pro v8

rsyslogd testbed

- vývoj vícevláknové aplikace pracující s řetězci a sítí je vždycky …
  - ... velmi obtížné
  - je nutné dělat testy a CI pomocí virtuálního testbedu



rsyslogd souhrn

- nasazeno v ostrém provozu necelé 2 roky
- přibližně 90% pokrytí (700 úzlů)
- 50 - 100GB měsíčně
  - 2GB komprimováno 7zipem
- dohled
  - nagios
  - cron skripty

Zpracování logů

- proč centrální syslog ?
  - mít logy na jednom místě umožňuje dělat centralizovanou magii (do_magic_here)

- standardní přístup
  - grep, perl, cron, tail -f

Zpracování logů

- standardní přístup
  - grep, perl, cron, tail -f
  - alerting from PBS logs
    - jobs_too_long

```
# du -sh .
105G .
# time grep -R "realuser" * > search.txt
real 18m39.447s
user 1m7.796s
sys 1m24.565s
# wc search.txt
   81636  1773549 21777400 search.txt
```

- perl je dobrý, ale ne úplně rychlý při zpracování 100GB
  - příklad:
    - vyhledej všechna přihlášení ze zlobivých IP

- pro analýzy je nutné použít databázi
  - ale nejdřív trošku plánování ...

Velikost

- gridy/cloudy škálují
  - logy typicky rostou ještě rychleji
    - je nutné použít databázi která je schopná škálovat

- clustering, partitioning
  - MySQL, PostgreSQL, ...

Struktura vrací úder

- logy nejsou jenom řádky textu, ale typicky datová struktura

  LOG ::= TIMESTAMP DATA

  DATA ::= LOGSOURCE PROGRAM PID MESSAGE

  MESSAGE ::= M1 | M2

- liší se program od programu
  - kernel, mta, httpd, ssh, kdc, …

- a to úplně přesně do klasických RDBMS nezapadá (statické tabulkové struktury)
  - bude se měnit … postgres9.3

Světýlko v dálce ?

- NoSQL databáze
  - nová technologie (skutečně ? ;)
  - cloudová technologie
  - technologie která škáluje
  - a vůbec je to ultra hustá technologie

- v našem projektu jsme se soustředili na
  - ElasticSearch
  - MongoDB

elasticsearch.

- ElasticSearch je full-textový vyhledávací stroj postavený na knihovně Lucene

  - je vymyšlený tak, aby byl distribuovaný
    - autodiscovery
    - automatický sharding/partitioning,
    - dynamická (re)alokace replik,

    - k dispozici jsou různí klienti

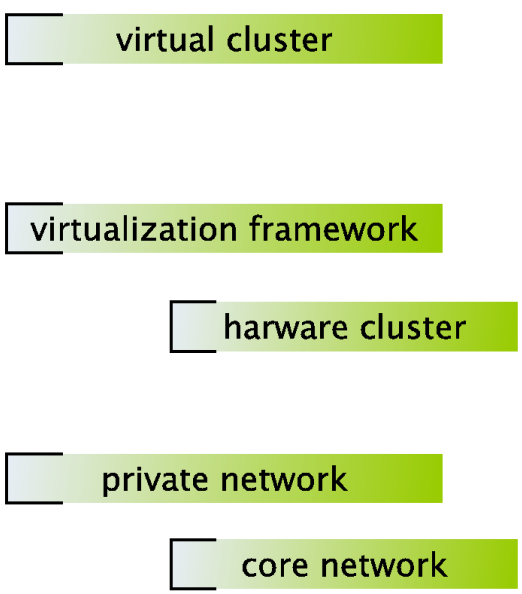
elasticsearch.

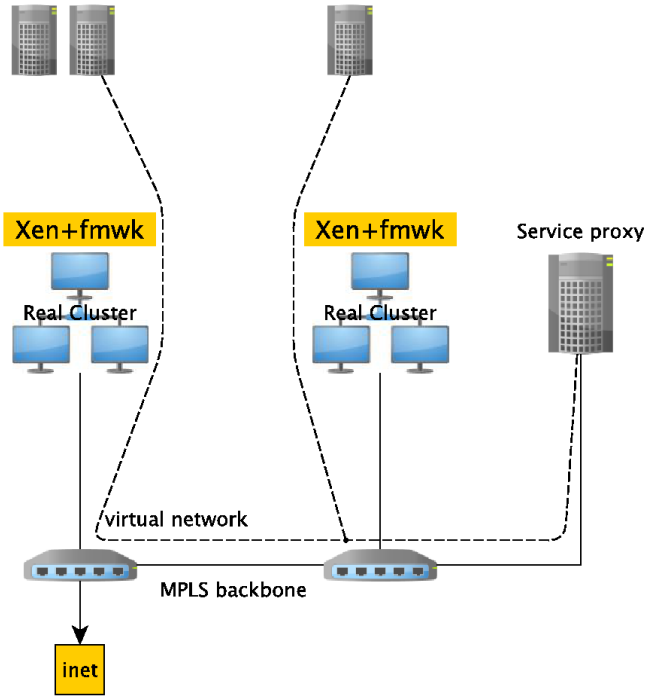
- REST, Thrift nebo nativní protokol
  - PUT indexname&data (json dokumenty)
  - GET _search?DSL_query...
    - index urychlí dotazy
- ElasticSearch by neměl být přímo dostupný veřejnosti
  - bez autentizace
  - bez šifrování
  - bez problému !!

~~rsyslog testbed~~ Privátní cloud

- místo testbedu si vytvoříme privátní cloud
  - členové cloudu jsou spouštěni podobně jako gridové úlohy
  - cloud je propojen dedikovanou privátní VLAN
  - komunikaci dovnitř a ven zajišťuje proxy

- reklama^2
  - Magrathea
  - OpenNebula





Rozklad řádků na struktury ...

- rsyslogd
  - omelasticsearch, ommongodb

LOG ::= TIMESTAMP DATA

DATA ::= LOGSOURCE PROGRAM PID MESSAGE

MESSAGE ::= **M1** | **M2** | **...**

- Logstash
  - grok
  - flexibilní architektura

logstash -- libgrok

- knihovna pro zpracování textu “vylepšenými” regulárními výrazy od Jordana Sissela

```
Nov  1 21:14:23 scorn kernel: pid 84558 (expect), uid 30206: exited on signal 3
```

In order, your brain reads a timestamp, a hostname, a process or other identifying name, a number, a program name, a uid, and an exit message. You might represent this in words as:

```
TIMESTAMP HOST PROGRAM: pid NUMBER (PROGRAM), uid NUMBER: exited on signal NUMBER
```

All of these can be represented by regular expressions. Grok comes with a bunch of pre-defined patterns to make getting started easier, including syslog patterns that help with the above. In grok, this pattern looks like:

```
%{SYSLOGBASE} pid %{NUMBER:pid} \(%{WORD:program}\), uid %{NUMBER:uid}: exited on signal %{NUMBER:signal}
```

All of the base grok patterns are in uppercase for style consistency. Each thing in %{} is evaluated and replaced with the regular expression it represents.

Grokked syslog


```
{
    _index: "logstash-2013.01.13",
    _type: "syslog",
    _id: "jTo_ymGdSaWluwom332bvw",
    _version: 1,
    _score: 1,
      _source: {
          @tags: [ ],
          @fields: {
              coll: [
                "160.217.209.82"
           ],
             logsource: [
                "hildor23-1.prf.jcu.cz"
           ],
             program: [
                "CRON"
           ],
             pid: [
                "3849"
           ],
             message: [
                "pam_unix(cron:session): session closed for user root"
           ]
        },
        @timestamp: "2013-01-13T23:20:01.000Z",
        @type: "syslog"
    }
}
```

logstash -- arch

- trubka na zpracování událostí
  - input | filter | output
- mnoho pluginů ...
- … což je velmi flexibilní

inputs

- amqp
- drupal_dblog
- eventlog
- exec
- file
- ganglia
- gelf
- gemfire
- generator
- heroku
- irc
- log4j
- lumberjack
- pipe
- redis
- relp
- sqs
- stdin
- stomp
- syslog
- tcp
- twitter
- udp
- xmpp
- zenoss
- zeromq

filters

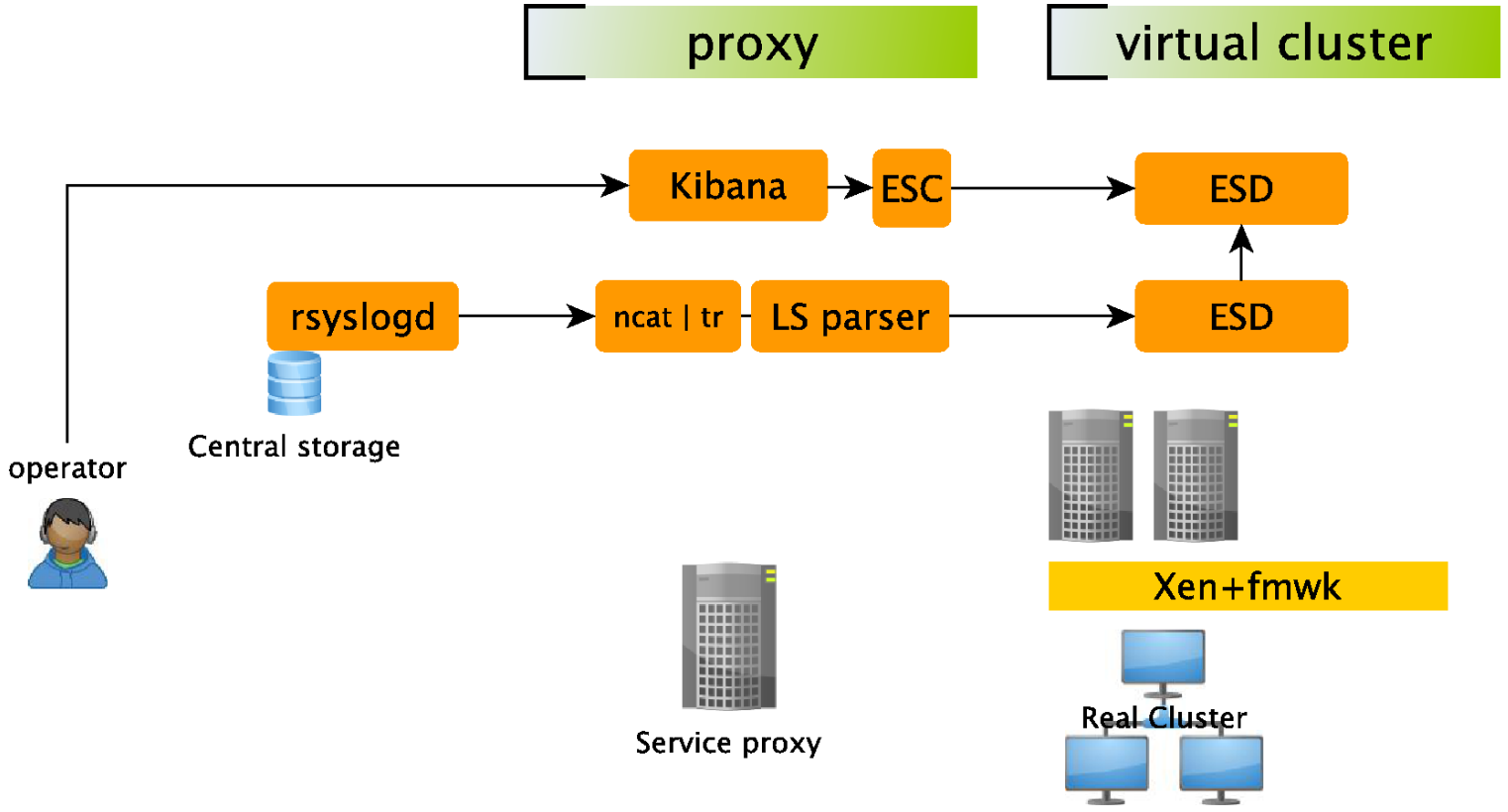
- alter
- anonymize
- checksum
- csv
- date
- dns
- environment
- gelfify
- geoip
- grep
- grok
- grokdiscovery
- json
- kv
- metrics
- multiline
- mutate
- noop
- split
- syslog_pri
- urldecode
- xml
- zeromq

outputs

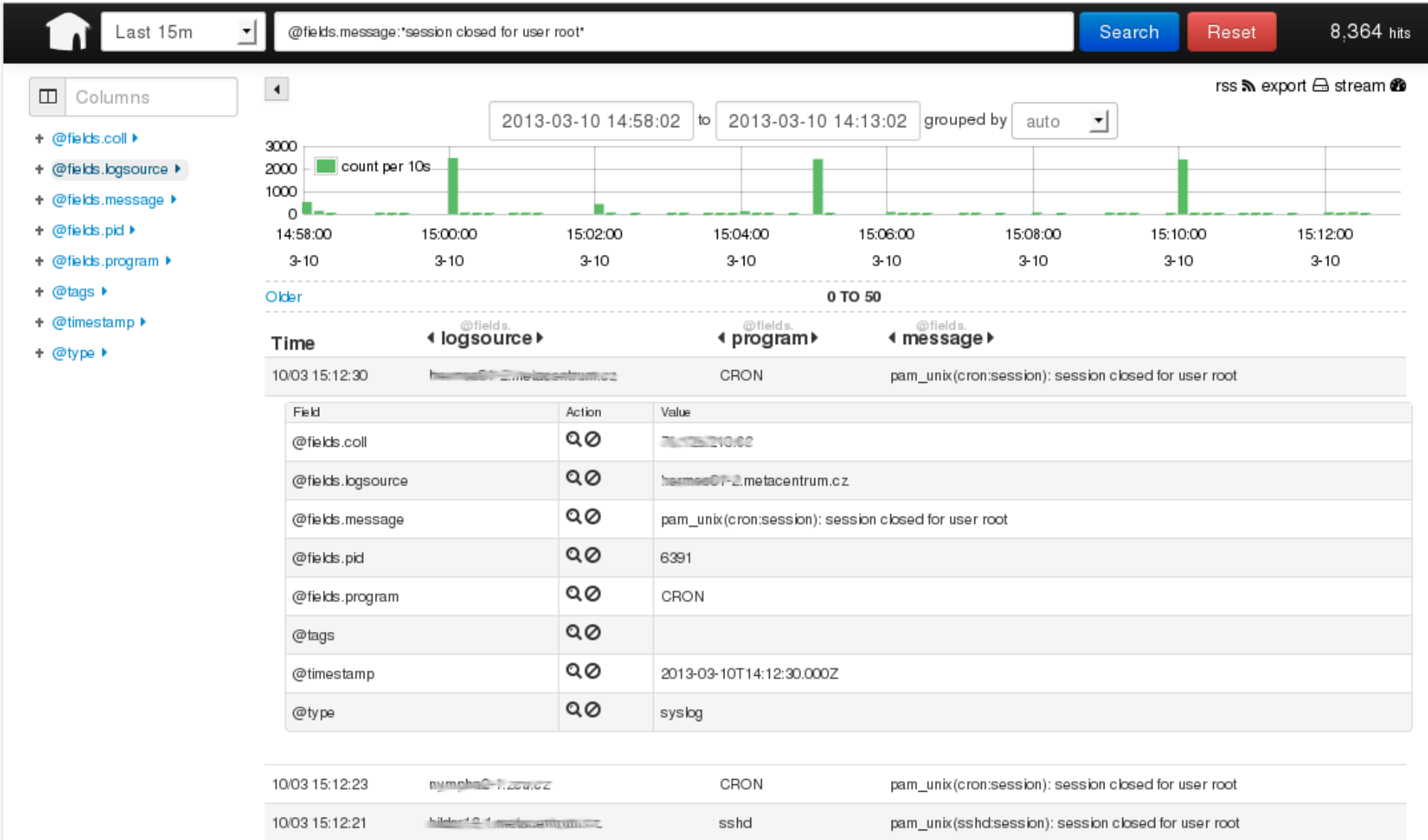
- amqp
- boundary
- circonus
- cloudwatch
- datadog
- elasticsearch
- elasticsearch_http
- elasticsearch_river
- email
- exec
- file
- ganglia
- gelf
- gemfire
- graphite
- graphtastic
- http
- internal
- irc
- juggernaut
- librato
- loggly
- lumberjack
- metriccatcher
- mongodb
- nagios
- nagios_nsca
- null
- opentsdb
- pagerduty
- pipe
- redis
- riak
- riemann
- sns
- sqs
- statsd
- stdout
- stomp
- syslog
- tcp
- websocket

Proxy pro zpracování logů

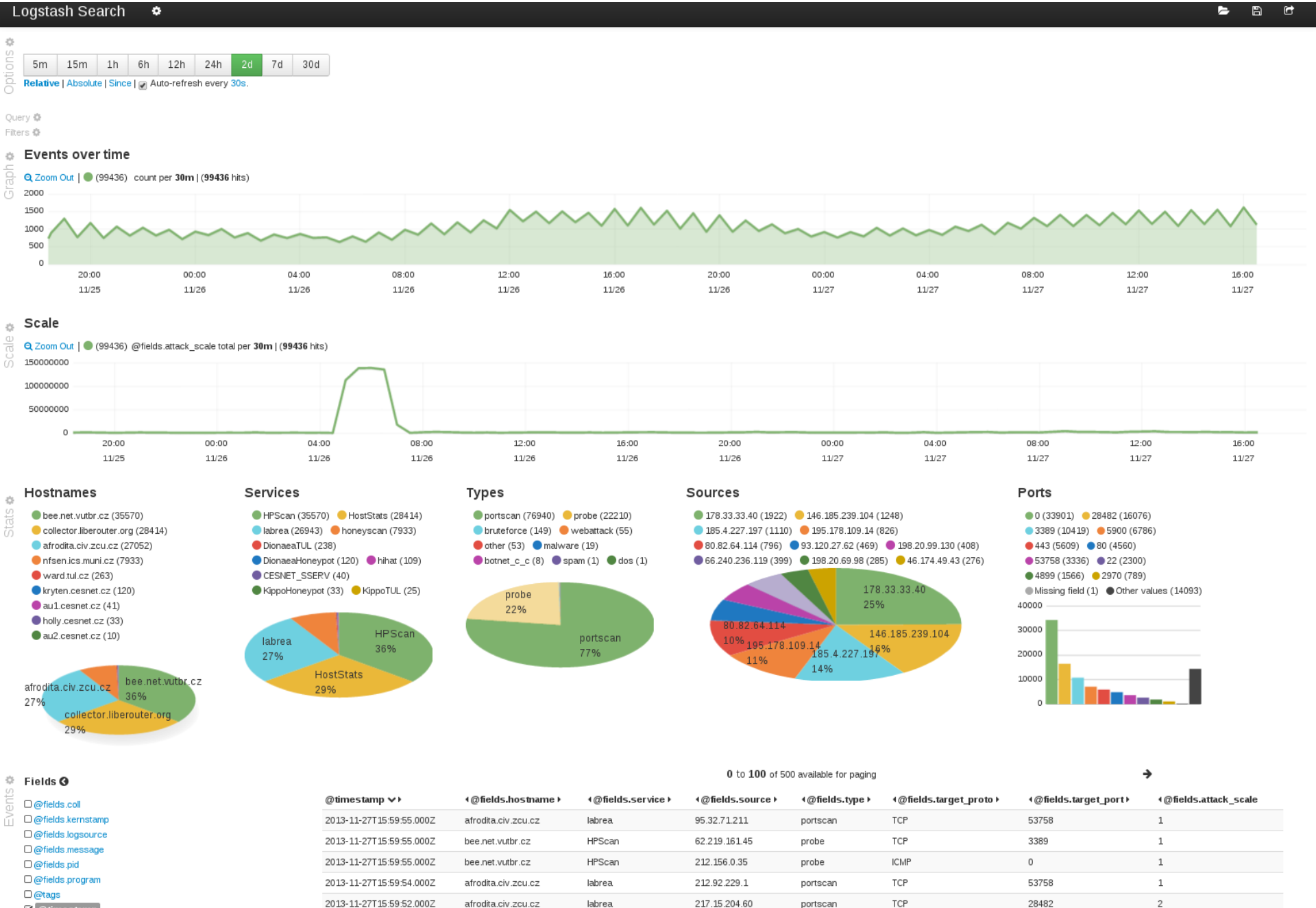
- ES + LS + Kibana
  - ... nebo ještě jednoduššeji (ES uvnitř LS)



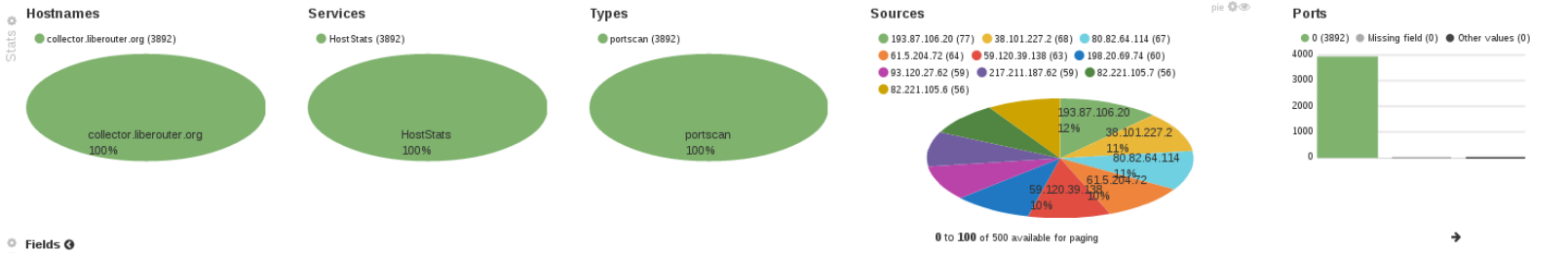
mimochodem ... Kibana

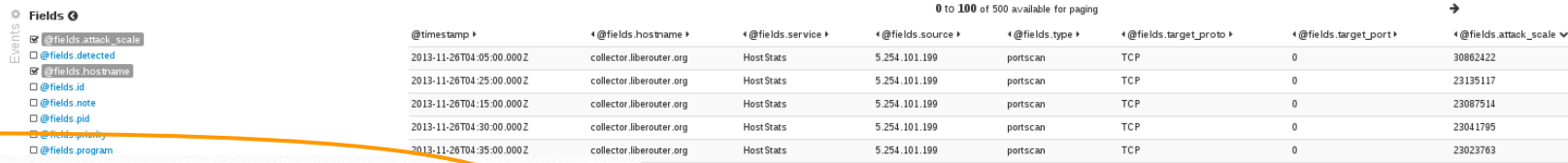


Kibana 3 pro Warden



mimochodem Kibana 3





@timestamp	@fields.hostname	@fields.service	@fields.source	@fields.type	@fields.target_proto	@fields.target_port	@fields.attack_scale
2013-11-26T04:05:00.000Z	collector.liberouter.org	HostStats	5.254.101.199	portscan	TCP	0	30862422
2013-11-26T04:25:00.000Z	collector.liberouter.org	HostStats	5.254.101.199	portscan	TCP	0	23135117
2013-11-26T04:15:00.000Z	collector.liberouter.org	HostStats	5.254.101.199	portscan	TCP	0	23087514
2013-11-26T04:30:00.000Z	collector.liberouter.org	HostStats	5.254.101.199	portscan	TCP	0	23041795
2013-11-26T04:35:00.000Z	collector.liberouter.org	HostStats	5.254.101.199	portscan	TCP	0	23023763
	erouter.org	HostStats	5.254.101.199	portscan	TCP	0	22976472
	erouter.org	HostStats	5.254.101.199	portscan	TCP	0	22945957
	erouter.org	HostStats	5.254.101.199	portscan	TCP	0	22917450
	erouter.org	HostStats	5.254.101.199	portscan	TCP	0	22889780
	erouter.org	HostStats	5.254.101.199	portscan	TCP	0	22880455
	erouter.org	HostStats	5.254.101.199	portscan	TCP	0	22834519
	erouter.org	HostStats	5.254.101.199	portscan	TCP	0	22828325
	erouter.org	HostStats	5.254.101.199	portscan	TCP	0	22806558
	erouter.org	HostStats	5.254.101.199	portscan	TCP	0	22800232
	erouter.org	HostStats	5.254.101.199	portscan	TCP	0	22779083
	erouter.org	HostStats	5.254.101.199	portscan	TCP	0	22742593
	erouter.org	HostStats	5.254.101.199	portscan	TCP	0	22703578
	erouter.org	HostStats	5.254.101.199	portscan	TCP	0	22701039
	erouter.org	HostStats	5.254.101.199	portscan	TCP	0	22682977
	erouter.org	HostStats	5.254.101.199	portscan	TCP	0	22631864
	erouter.org	HostStats	5.254.101.199	portscan	TCP	0	22571509
	erouter.org	HostStats	5.254.101.199	portscan	TCP	0	20783827
	erouter.org	HostStats	5.254.101.199	portscan	TCP	0	16915881
	erouter.org	HostStats	5.254.101.199	portscan	TCP	0	12319164
	erouter.org	HostStats	121.56.218.182	portscan	TCP	0	456607
	erouter.org	HostStats	61.147.113.66	portscan	TCP	0	325534
	erouter.org	HostStats	58.215.241.135	portscan	TCP	0	325527
	erouter.org	HostStats	85.248.123.130	portscan	TCP	0	325409
	erouter.org	HostStats	192.96.206.139	portscan	TCP	0	324450
	erouter.org	HostStats	113.240.245.243	portscan	TCP	0	323055
	erouter.org	HostStats	61.147.116.57	portscan	TCP	0	305469

More Bad Port 0 Traffic

Published: 2013-11-25,
Last Updated: 2013-11-25 20:57:57 UTC
by Johannes Ullrich (Version: 1)

3 comment(s)

Thanks to an alert reader for sending us a few odd packets with "port 0" traffic. In this case, we got full packet captures, and the packets just don't make sense.

The TTL of the packet changes with source IP address, making spoofing less likely. The TCP headers overall don't make much sense. There are packets with a TCP
0, or packets with odd flag combinations. This could be an attempt to fingerprint, but even compared to nmap, this is very noisy. The packets arrive rather slow, far fro

Here are a couple samples (I anonymised the target IP). Any hints as to what could cause this are welcome.

```
IP truncated-ip - 4 bytes missing! (tos 0x0, ttl 52, id 766, offset 0, flags [DF], proto TCP (6), length 88)
    94.102.63.55.0 > 10.10.10.10.0:  tcp 68 [bad hdr length 0 - too short, < 20]

0x0000:  4500 0058 02fe 4000 3406 91f1 5e66 3f37
0x0010:  0a0a 0a0a 0000 0000 55c3 7203 0000 0000
0x0020:  0c00 0050 418b 0000 6e82 ef01 0000 0000
0x0030:  25b0 ce4b 0000 0000 a002 3cb0 9a8b 0000
0x0040:  0204 0f2c 0402 080a 0005 272d 0005 272d
0x0050:  0103 0300


IP truncated-ip - 4 bytes missing! (tos 0x10, ttl 47, id 28629, offset 0, flags [DF], proto TCP (6), length 60)
    46.137.48.107.0 > 10.10.10.10.0: Flags [P.UW] [bad hdr length 56 - too long, > 40]

0x0000:  4510 003c 6fd5 4000 2f06 68cf 2e89 306b
0x0010:  0a0a 0a0a 0000 0000 51a9 89b8 0000 0000
0x0020:  e6b8 0050 b315 0000 ec67 0d66 0000 0000
0x0030:  0000 0000 0000 0000


IP truncated-ip - 4 bytes missing! (tos 0x80, ttl 51, id 45284, offset 0, flags [DF], proto TCP (6), length 60)
    186.202.179.99.0 > 10.10.10.10.0: Flags [SUW], seq 1603085765, win 27016, urg 0, options [[bad opt]

0x0000:  4580 003c b0e4 4000 3306 1416 baca b363
0x0010:  0a0a 0a0a 0000 0000 5f8d 25c5 0000 0000
0x0020:  aba2 6988 23fa 0000 f271 af2a 0000 0000
0x0030:  0000 0000 0000 0000
```

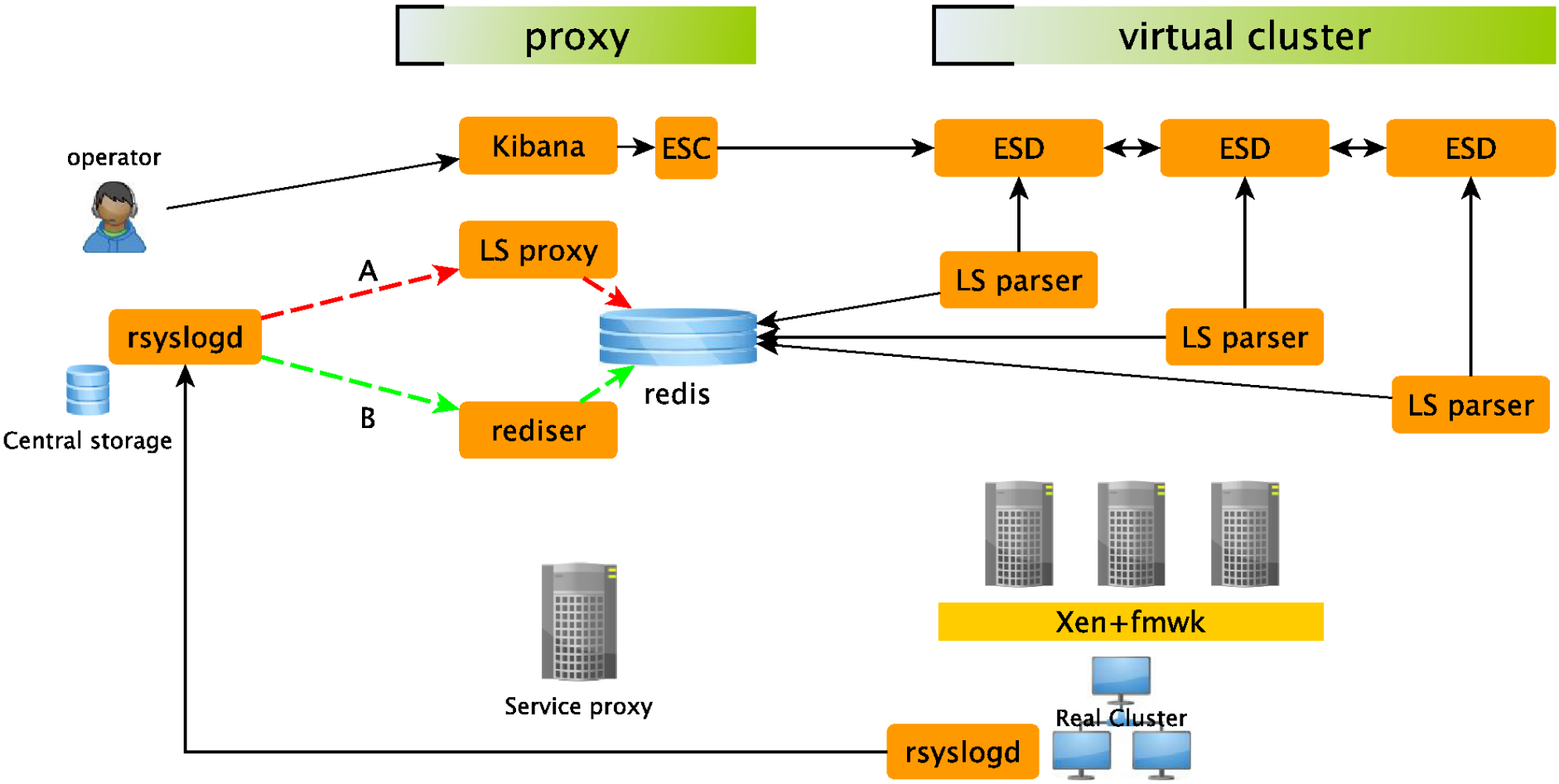
Johannes B. Ullrich, Ph.D.
SANS Technology Institute

Výkon

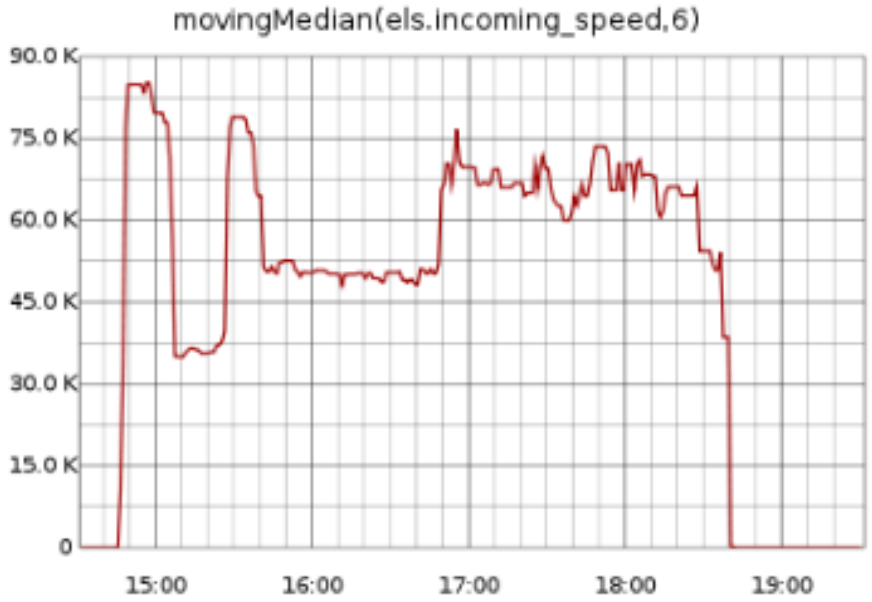
- Zpracování logů pouze na proxy nemusí pro gridy nebo jiná velká prostředí stačit ..
  - vytvoření cloudové služby je s LS snadné. To co potřebujeme navíc je fronta >> redis
  - použít modul input redis místo tcp

- Zrychlení procesu
  - batching, bulk indexing
  - rediser
    - výměna generického logstashe za speciální mikroskript

Více cloudu, více Adidas



Souhrn LS + ES



- upload
  - testdata
    - logy z ledna 2013
    - 105GB -- cca 800M událostí
  - zaindexováno za 4h
    - 8 datových uzlů (indexerů)
    - 16 sdílených parserů (LS na datovém uzlu)
    - 4 uzly - 8h
  - rychlost kolísá hlavně kvůli charakteru zpracovávaných dat (mnoho maličkých událostí)
    - během běžného provozu není “tak velký” cluster nutný

Souhrn LS + ES

- Rychlost nahrávání dat do ES záleží na ...
  - velikosti grokovaných dat a finálních dokumentů,
  - velikosti dávek při dávkovém zpracování,
  - filterů a akcí během zpracování,
  - velikosti interních front v LS v případně více výstupních pluginů (stejné jako odbočovací pruhy v rsyslogu:),
  - nastavení indexů v elasticsearch,
  - ...
  - ...
  - vyladění na vysokou rychlost je vždy manuální prácička (graphite, ...)

Souhrn LS + ES

- rychlost vyhledávání ~

```
# du -sh .
105G .
# time grep -R "realuser" * > search.txt
real 18m39.447s
user 1m7.796s
sys 1m24.565s
# wc search.txt
   81636  1773549 21777400 search.txt
#

# ./el_listnodes.py
10.0.0.31 id HRf5TJebQw6_cYxAsS2mtQ indices.docs.count 388345192
10.0.0.3 id BBcHTUk9SkWxhynzoPafog indices.docs.count 409604004
10.0.0.1 id 1pTq45TKTGavwnZGKXPcfQ indices.docs.count 0
# time sh curltest.sh > search1
real 0m34.944s
user 0m0.536s

# grep '"_id"' search1 |wc
  81636  244908 3265440
```

Podrobnější analýza logů

- ES je fulltext SE, nikoli databáze
  - ale my bychom databázičku potřebovali ...

- Dokumentově orientovaná databáze
  - Auto-Sharding
  - Mapreduce a agregační framework

Podrobnější analýza logů

- MongoDB
  - Lze nakrmit přes Logstash podobně jako ES
    - analýza logů z sshd

```
AAARESULT (?:Accepted|Failed|Authorized|identification|Invalid|disconnect|tried|refused)
METHOD (?:[a-z-]+|correct key)
PRINCIPAL [a-zA-Z0-9_/-]+@%{HOSTNAME}

AUTHN %{AAARESULT:result} %{METHOD:method} for (invalid user )?%{USER:user} from %{IPORHOST:rem
AUTHZ %{AAARESULT:result} to %{USER:user}, krb5 principal %{PRINCIPAL:principal} \(krb5_kuserok
SCAN Did not receive %{AAARESULT:result} string from %{IPORHOST:remote}
INVALID %{AAARESULT:result} user %{USER:user} from %{IPORHOST:remote}
DISCONNECT Received %{AAARESULT:result} from %{IPORHOST:remote}: 11: disconnected by user
WRONGKEY Authentication %{AAARESULT:result} for %{USER:user} with %{METHOD:method} but not from
REFUSED %{AAARESULT:result} connect from %{IPORHOST:remote} \(%{IPORHOST:remote}\)

SSHATTEMPT (?:%{AUTHN}|%{AUTHZ}|%{SCAN}|%{INVALID}|%{DISCONNECT}|%{WRONGKEY}|%{REFUSED})

SSHBASE3 (%{SYSLOGTIMESTAMP} (%{IP:coll} )?%{SYSLOGHOST:logsource} )?%{SYSLOGTIMESTAMP:timestam
SSHLINE %{SSHBASE3} %{SSHATTEMPT:message}
```

MapReduce

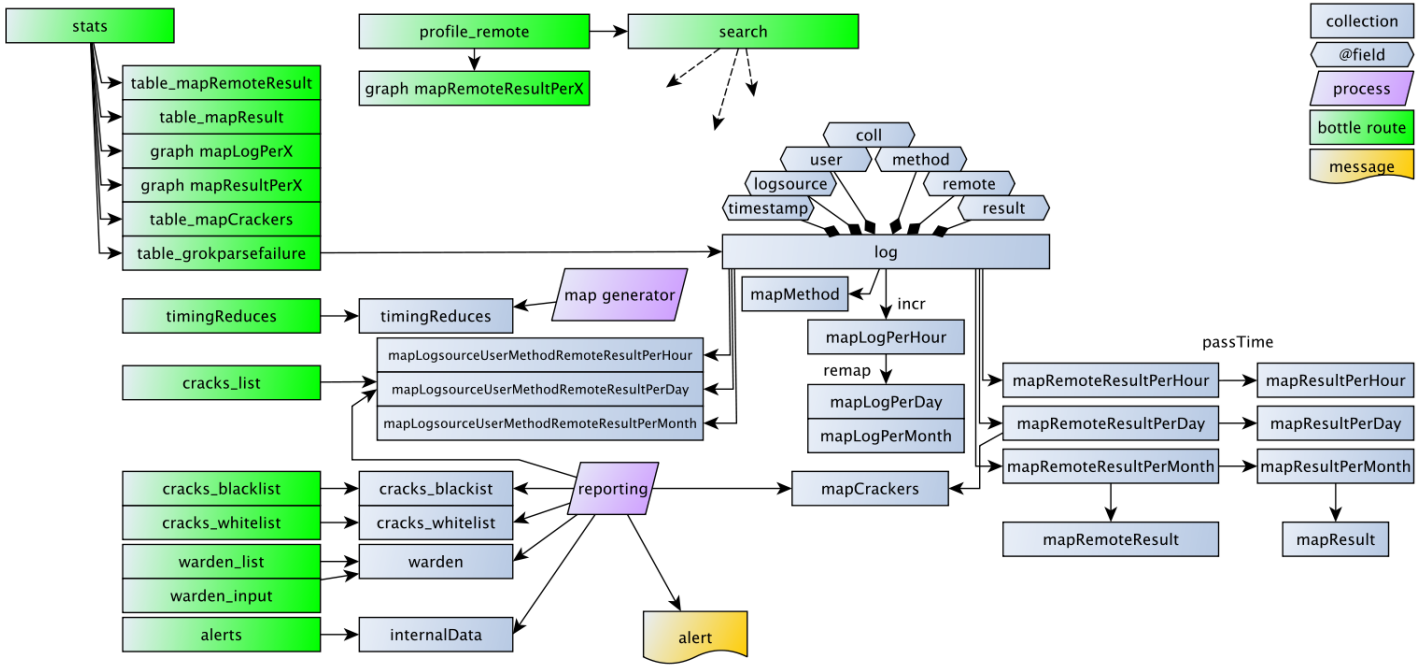
```
'map': Code("
                function() {
                        var a = new Date(
                                this.@timestamp.getFullYear(),
                                this.@timestamp.getMonth(),
                                this.@timestamp.getDate(),
                                this.@timestamp.getHours(),
                                0, 0, 0);
                        emit(
                                { t: a,
                                logsource: (this.@fields.logsource ? this.@fields.logsource.toString() : 'NULL'),
                                user: (this.@fields.user ? this.@fields.user.toString() : 'NULL'),
                                method: (this.@fields.method ? this.@fields.method.toString() : 'NULL'),
                                remote: (this.@fields.remote ? this.@fields.remote.toString() : 'NULL'),
                                result: (this.@fields.result ? this.@fields.result.toString() : 'NULL'),
                                },
                                {count: (this.value ? this.value.count : 1)}
                        );
                }",
        {}),

'reduce': Code('
                function(k,v) {
                        var r = { count:0 } ;
                        v.forEach(function(v) {r.count+=v.count });
                        return r;
                }',
        {}),
```

```
{
   "_id": ObjectId("51379dd1e4b0fad32a766fa7"),
   "@timestamp": ISODate("2013-02-03T13:12:44.0Z"),
   "@tags": [],
   "@fields": {
     "coll": {
       "0": "ddd.aaa.bbb.ccc"
    },
     "logsource": {
       "0": "wknode23.sub.domain.cz"
    },
     "result": {
       "0": "Invalid"
    },
     "user": {
       "0": "prueba"
    },
     "remote": {
       "0": "218.85.135.29"
    }
  },
   "@message": "Feb 3 14:12:42 ddd.aaa.bbb.ccc wknode23.sub.domai
   "@type": "ssh"
}
```

Mongomine

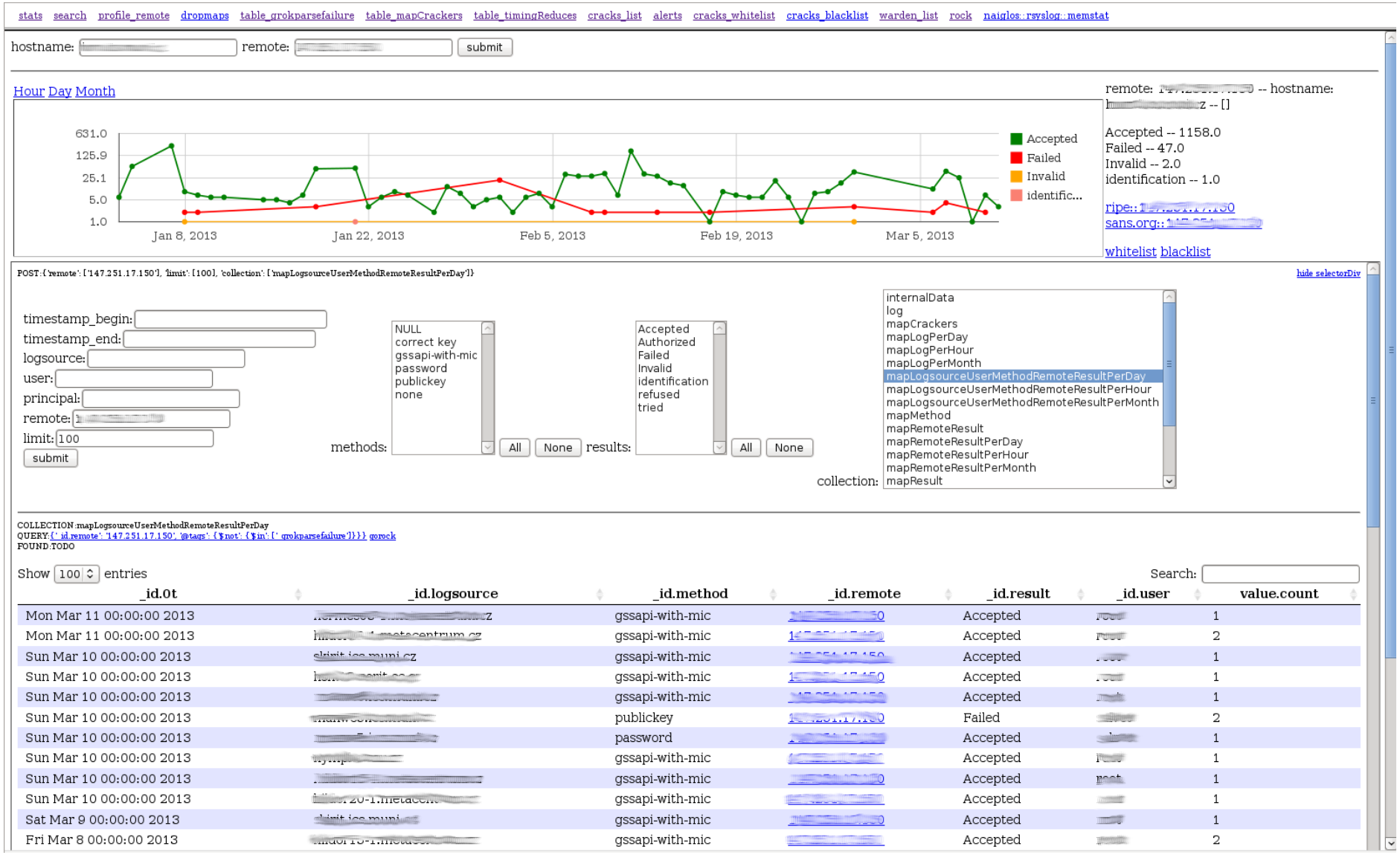
- nad kolekcí/tabulkou ze získaných logů
  - časově orientované hierarchické agregace
    - perHour, perDay, perMonth -- profilování, prohlížení)
  - speciální pohledy (mapCrackers)
    - `mapRemoteResultsPerDay.find( {time= last 14days, result={fail}, count>20} )`
  - externí data (Warden, Tor, ...)



Mongomine

- aplikace pro prohlížení výsledků Logstash + MongoDB
  - analýza bezpečnostních událostí
    - python bottle webapp
    - ~~Google~~ Highcharts
  - automatický reporting
    - úspěšná příhlášení z
      - mapCrackers
      - Warden
      - výstupních uzlů sítě Tor
      - ...

Mongomine


stats search profile_remote dropmaps table_grokparsefailure table_mapCrackers table_timingReduces cracks_list alerts cracks_whitelist cracks_blacklist warden_list rock naiglos::rsyslog::memstat
hostname: remote: submit
Hour Day Month
631.0
125.9
25.1
5.0
1.0
Jan 8, 2013
Jan 22, 2013
Feb 5, 2013
Feb 19, 2013
Mar 5, 2013
Accepted
Failed
Invalid
identific...
remote: -- hostname:
-- []
Accepted -- 1158.0
Failed -- 47.0
Invalid -- 2.0
identification -- 1.0
ripe::
sans.org::
whitelist blacklist
POST:{'remote': ['147.251.17.150'], 'limit': [100], 'collection': ['mapLogsourceUserMethodRemoteResultPerDay']}
hide selectorDiv
timestamp_begin:
timestamp_end:
logsource:
user:
principal:
remote:
limit: 100
submit
methods:
NULL
correct key
gssapi-with-mic
password
publickey
none
All
None
results:
Accepted
Authorized
Failed
Invalid
identification
refused
tried
All
None
collection:
internalData
log
mapCrackers
mapLogPerDay
mapLogPerHour
mapLogPerMonth
mapLogsourceUserMethodRemoteResultPerDay
mapLogsourceUserMethodRemoteResultPerHour
mapLogsourceUserMethodRemoteResultPerMonth
mapMethod
mapRemoteResult
mapRemoteResultPerDay
mapRemoteResultPerHour
mapRemoteResultPerMonth
mapResult
COLLECTION:mapLogsourceUserMethodRemoteResultPerDay
QUERY:{' id.remote': '147.251.17.150', '@tags': {'$not': {'$in': [' grokparsefailure']}}} gorock
FOUND:TODO
Show 100 entries
Search:
_id.0t
_id.logsource
_id.method
_id.remote
_id.result
_id.user
value.count
Mon Mar 11 00:00:00 2013 gssapi-with-mic Accepted 1
Mon Mar 11 00:00:00 2013 gssapi-with-mic Accepted 2
Sun Mar 10 00:00:00 2013 skirit.ics.muni.cz gssapi-with-mic Accepted 1
Sun Mar 10 00:00:00 2013 gssapi-with-mic Accepted 1
Sun Mar 10 00:00:00 2013 gssapi-with-mic Accepted 1
Sun Mar 10 00:00:00 2013 publickey Failed 2
Sun Mar 10 00:00:00 2013 password Accepted 1
Sun Mar 10 00:00:00 2013 gssapi-with-mic Accepted 1
Sun Mar 10 00:00:00 2013 gssapi-with-mic Accepted 1
Sun Mar 10 00:00:00 2013 gssapi-with-mic Accepted 1
Sat Mar 9 00:00:00 2013 gssapi-with-mic Accepted 1
Fri Mar 8 00:00:00 2013 gssapi-with-mic Accepted 2

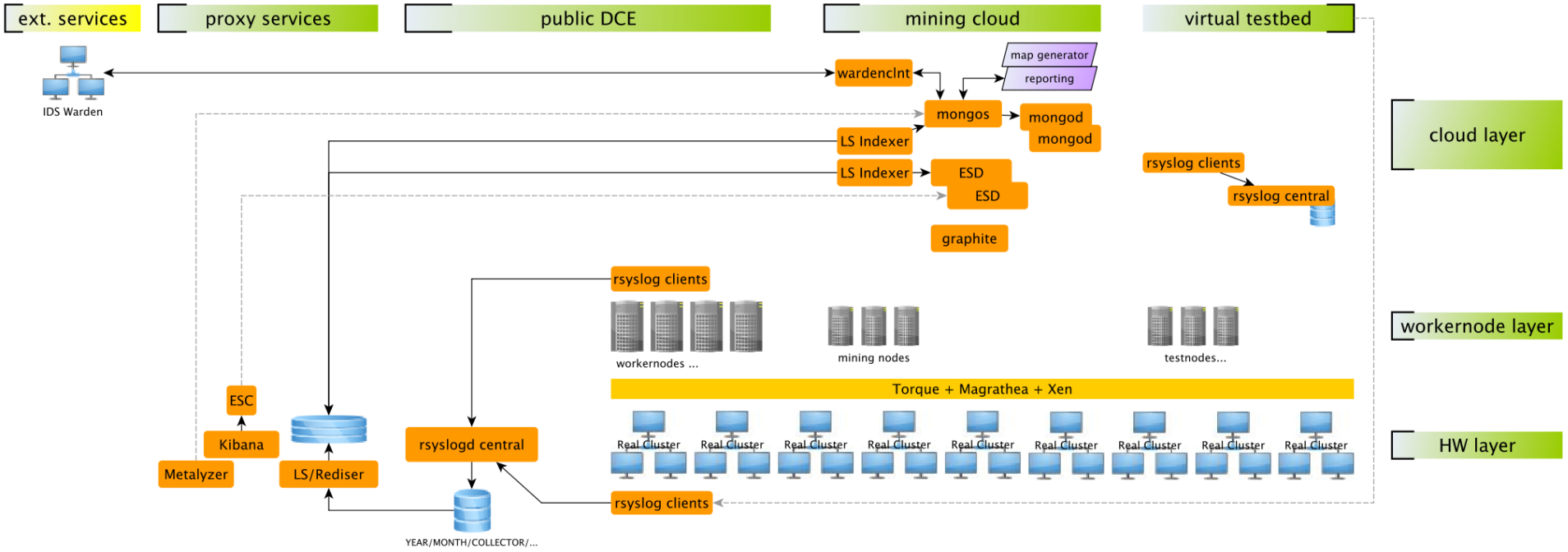
Sourhn Mongomine

- testcase
  - 20GB -- leden 2013
  - 1 uzel MongoDB, 24 CPUs, 20 shardů
  - 1 uzel parseru, 6 LS parserů

- rychlost
  - upload -- přibl. 8h (nemá dávkové zpracování :(
  - 1st MR job -- přibl. 4h
  - počítávání inkrementálních MR během provozu -- přibl. 10s

Pohled z výšky ...

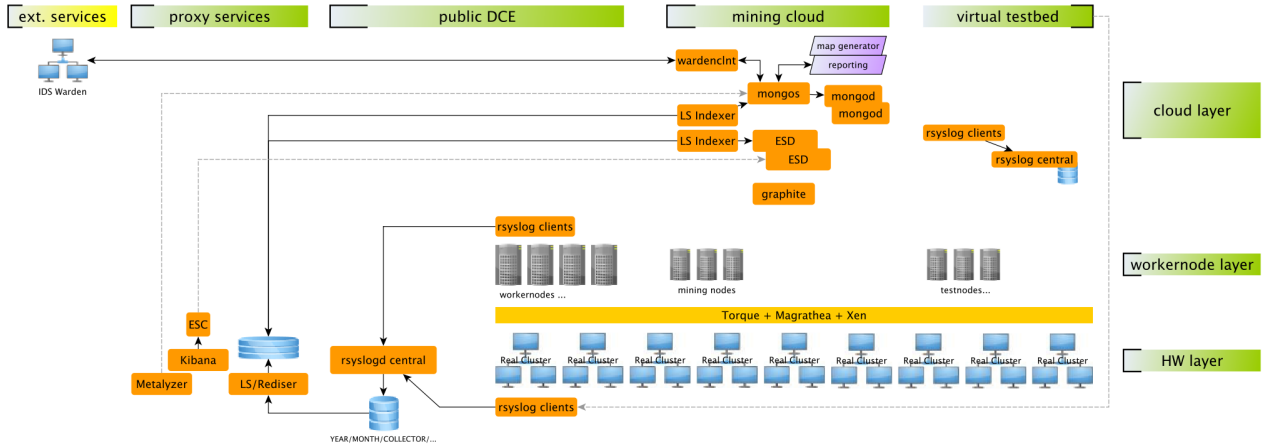
- rsyslogd + testbed
- LS + ES
- LS + Mongomine + Ext



Měření se nám líbilo ...

- proxy je samozřejmě nejdůležitější součást
  - musíme jí udělat redundantní
- virtualizace je boží (když seje ;)
  - pokud je skutečně dynamická umožní extrémně zrychlit vývoj i testování aplikací
- více analýz
  - víc externích dat
  - rychleji

Závěr



- Funguje to
  - škáluje to jak potřebujeme
  - sem tam jsme si něco dopsali
  - řešení je schopné zpracovat rozličná data
    - která mají různou nebo prakticky žádnou strukturu

- Vlastnosti
  - sběr dat -- rsyslog
  - zpracování dat -- logstash
  - rozhraní s vysokou interakcí -- ES, kibana
  - analýza a poplachy -- mongomine

ESB EGI Technical forum 2013
Virtual Machine Walkthrough

http://home.zcu.cz/~bodik/metasw/esbegitf/

Otázky ?

Teď nebo or ...

https://wiki.metacentrum.cz/wiki/User:Bodik

mailto:bodik@civ.zcu.cz

mailto:kouril@ics.muni.cz