Produkční elastic cluster

“you know, for search”

Jan Hrnčíř
@Master Internet

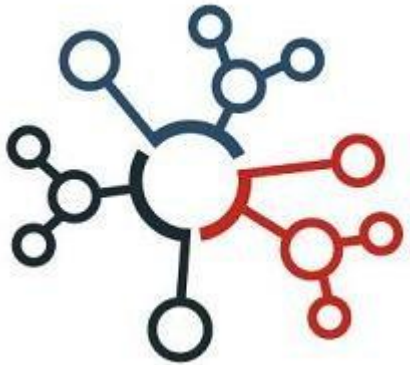
master
data in motion

Očekávání

Dostupnost - HA

Rychlost (peak hour) - navržení clusteru / rozvržení shardů...

Rozšiřitelnost - přidávání dalších nodů do clusteru za běhu



Design perfektního clusteru

Ovšem ne dnes:

Vyhledávání / zápis

Komplexita dotazů

Růst indexů

Celková zátěž

...

Návrh HW

CPU - CPU vs vCPU, verze CPU

Paměť - Java heap vs. zbytek

HDD - rychlost, prostor
- lokální storage / latence
- SSD only
- file system (ext4 / xfs)

Síť - šířka pásma
- latence(!)
- interní komunikace
- použití jumbo rámců (MTU 9000)

ES nody

Master - udržuje stav celého clusteru

Data - ukládání dat. CRUD / vyhledávání

Ingest - zpracování dokumentů (Logstash?)

Coordinating (5.0+) / Client

- REST API, komunikace s data nody (~LB)
- stav umístění dat v clusteru, vrací výsledky

Tribe - deprecated (5.4) - komunikace napříč clustery

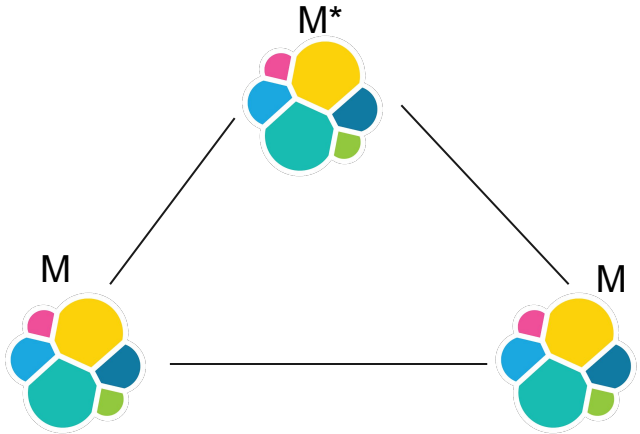
Master node

3+ master nody

Split brain situace

```
minimum_master_nodes = (master_eligible_nodes/2)+1
```

Volba master nodu - automaticky, defaultně ping ze všech nodů



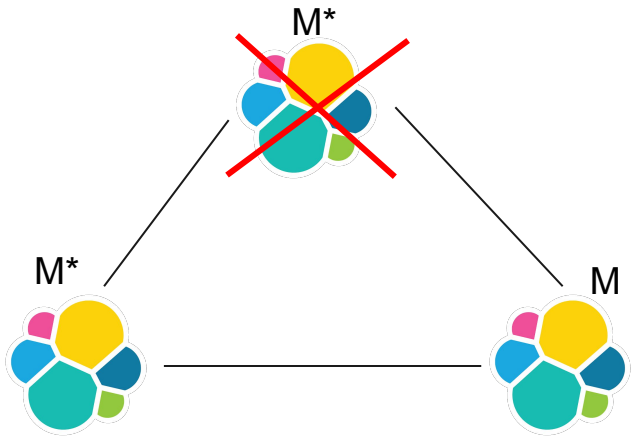
Master node

3+ master nody

Split brain situace

```
minimum_master_nodes = 2
```

Volba master nodu - automaticky, defaultně ping ze všech nodů



Další problémy /řešení

Výpadek hosta

Neodpovídá JVM (přetížení?) - volba nového Master nodu

Stejné platí i pro ostatní nody - rebalancing nodu a následné zpomalení

Zpomalní - debugging využitých prostředků, debug dat (velikost a počet shardů, lucene merge…)

Virtualizace / instance ES

Rozdělení rolí

Rozdělení shardů napříč clusterem - zamezení alokace instancí shardu na jednoho hosta (hostname/adresa)
`cluster.routing.allocation.same_shard.host`

Rozdělení nodů napříč clusterem - affinity pravidla
`node.attr.rack_id: "rack_mai_X"`
`cluster.routing.allocation.awareness.attributes: "rack_mai_X"`
- preferují se lokální shardy v rámci jedné lokality

Nepřekračovat <32GB pro heap size, nealokovat veškerou paměť JVM, nastavit memlock

Monitoring - node

Open file descriptors

```
ulimit -n 65536
```

JVM heap, memory pool information, garbage collection

Systémové prostředky (RAM, CPU - lucene merge, HDD…)

```
stats_limit=process,jvm,os,fs...
curl -XGET http://localhost:9200/_nodes/stats/$stats_limit?pretty
```

Monitoring - cluster

Cluster status

```
curl -XGET http://localhost:9200/_cluster/health?pretty
```

Index status

```
curl -XGET http://localhost:9200/_cat/indices?v
```

Informace o nodech

```
curl -XGET http://localhost:9200/_cat/nodes?v
```

Kontrola alokace shardů

```
curl -XGET http://localhost:9200/_cluster/allocation/explain?pretty
```

Zálohování - Snapshot & Restore

Sdílené uložiště - mount fs na všech m-d nodech do stejného umístění

```
path.repo: ["/mnt/backup/elastic"]
```

Frekvence záloh - inkrementální zálohování

Obnova přes API, možnost změny názvu indexu

Demo

Demo

- 3x Master / 3x Data node
- OS - CentOS 7
- Java - openjdk 1.8
- Elastic - 6.5.4
- Nastaveni firewallu (default http port 9300)
- Konfigurace:

```
node.name: $-node
cluster.name: elastic-brno-meetup
node.master: false or true
node.data: false or true
node.ingest: false or true
bootstrap.memory_lock: true ##systemd service = LimitMEMLOCK=infinity
discovery.zen.ping.unicast.hosts: ["master1", "master2", "master3”]
```

Poděkování

elastic

https://discuss.elastic.co
Komunita
Meetupy jako je tento

Jan Hrnčíř
hrncir@master.cz
@Master Internet

master
data in motion