Visualizing solr performance in Kibana

Motivation

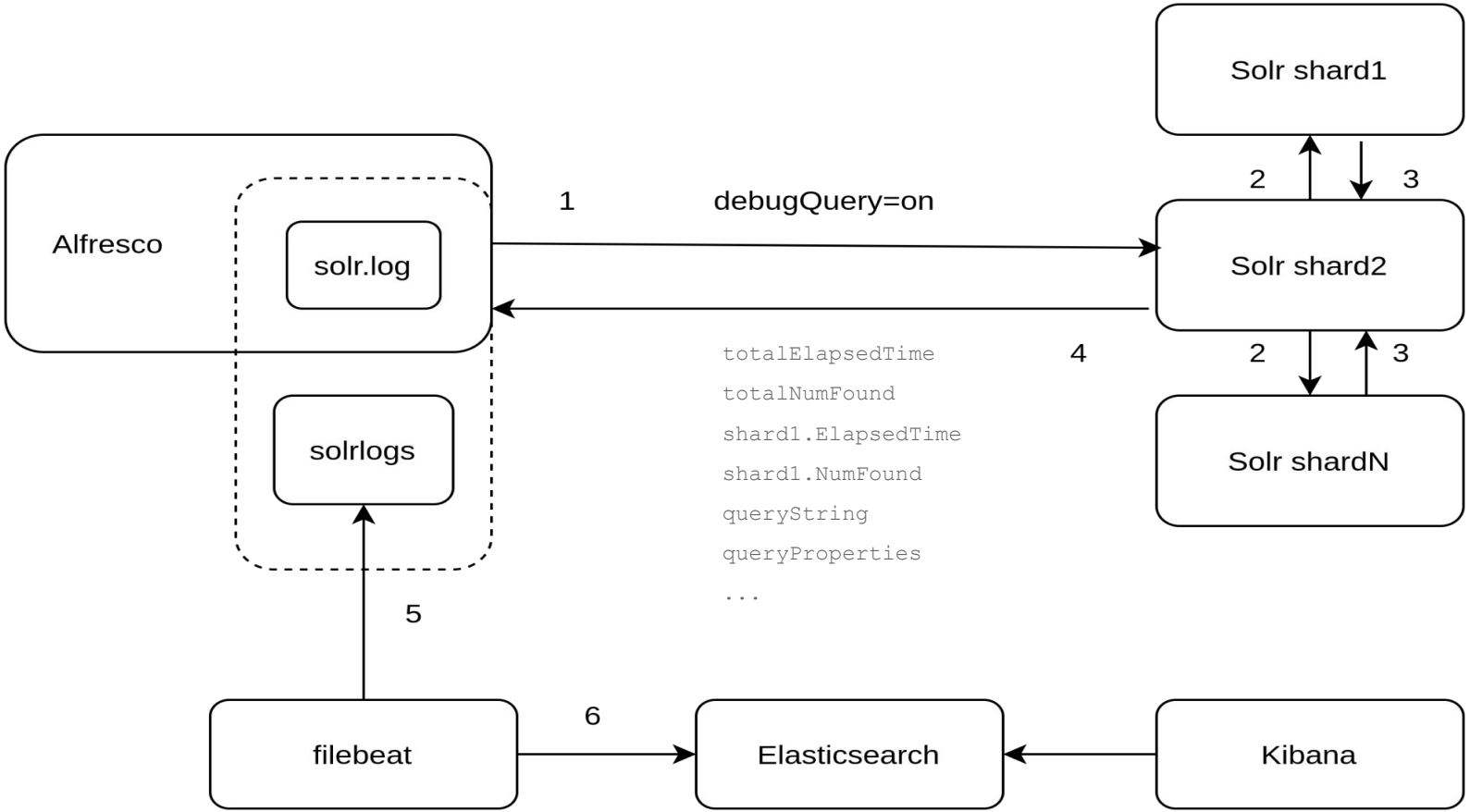
Ethias

- Get an idea about solr queries in terms of
    - Response times
    - Number of hits
    - Properties most searched into
    - Split per
        - Nodes
        - Shards

Proposals

- Loggers on alfresco + solr side (via slow logging)
  - Multiple lines relevant for 1 log request
  - No easy correlation between alfresco and solr logs
- Instrumentation (zipkin)
  - Allows to create virtual spans without instrumenting solr (see also Future work)
- Solr debug parameter (https://github.com/xenit-eu/alfresco-solr-debug)
  - Centralized information (alfresco)
  - Relatively easy to implement
  - Integration with Elasticsearch + Kibana

Architecture



Alfresco amp

SolrQueryHTTPClient

```
if(useSolrDebug) {
                url.append("&debugQuery=on");
            }

...

if(useSolrDebug) {
                JSONObject queryString = solrQueryParser.formatToQueryJson((String) body.get("query"));
                s_logger.debug("{\"parsedQuery\":" + queryString + ", \"track\":" +
((JSONObject)json.get("debug")).get("track") + ",\"totalNumFound\":" + results.getNumberFound() +
",\"totalElapsedTime\":" + results.getQueryTime() + "}");
            }
```

Alfresco amp

```
log4j.appender.solrQueryLog=org.apache.log4j.DailyRollingFileAppender
log4j.appender.solrQueryLog.File=/usr/local/tomcat/logs/solrquery.log
log4j.appender.File.Append=true
log4j.appender.File.DatePattern='.'yyyy-MM-dd
log4j.appender.solrQueryLog.layout=net.logstash.log4j.JSONEventLayoutV1
log4j.appender.solrQueryLog.layout.ConversionPattern=%d{yyyy-MM-dd} %d{ABSOLUTE} %-5p [%c] [%t] %m%n


log4j.logger.eu.xenit.subsystems.patches.solr.SolrQueryHTTPClient=debug, solrQueryLog
log4j.additivity.eu.xenit.subsystems.patches.solr.SolrQueryHTTPClient=false
```

Solrlogs

```
alfresco:
    volumes:
    - logs:/usr/local/tomcat/logs
solrlogs:
    image: busybox
    command: [/bin/sh, -c, 'sleep 30 ; tail -n+1 -F /usr/local/tomcat/logs/solrquery.log']
    container_name: solrlogs
    depends_on:
      - alfresco
    volumes:
      - logs:/usr/local/tomcat/logs
    labels:
    - co.elastic.logs/module=json-alfresco
    - eu.xenit.index=solrlogs
```

Elasticsearch integration

- filebeat.yml

```
output.elasticsearch:
  hosts: ["172.17.0.1"]
  ....
  indices:
   - index: "%{[docker.container.labels.eu_xenit_index]}"
```

- **filebeat/module/alfresco-json/log/** (parsing/ingest module)
- **elasticsearch/post-start.sh** (ilm for new index + index pattern in kibana)
- **elasticsearch/mapping.json** (types for fields, aliases for shorter paths)
- **send_searches.sh** (send some searches via apix to have data in ES)

Future work

1. Test on working environment
   - Demo?
2. Alerts via ES
3. Continue exploratory work on zipkin (https://github.com/xenit-eu/alfresco-zipkin/blob/tmp-solr-debug/alfresco-zipkin-repo/src/main/java/eu/xenit/alfresco/instrumentation/repo/TracingHttpClient.java)