Solr lab for sharding experiments

Roxana Angheluta (Xenit Solutions)
Axel Faust (Acosix GmbH)

Motivation (problem)

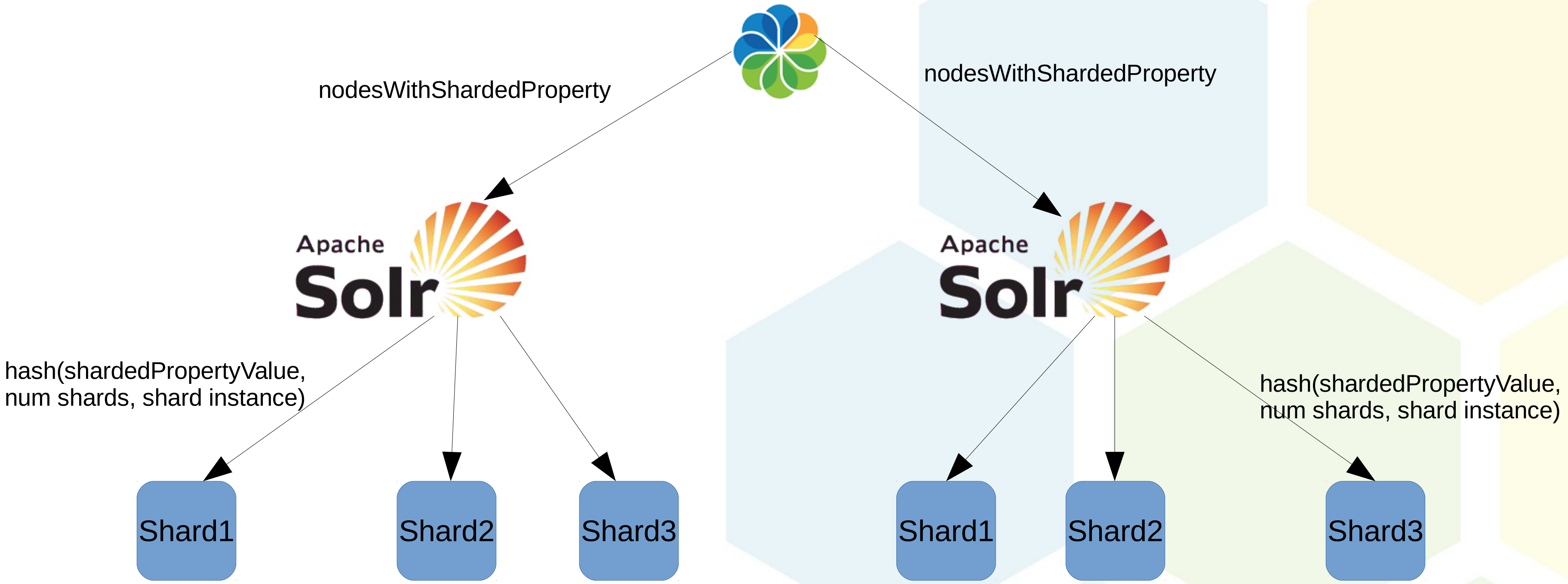
Requirements

- Large archives with >50mil documents
- No sites
- Search speed <3s
- Particular use case: most of the queries are targeted to a specific client id

Solution

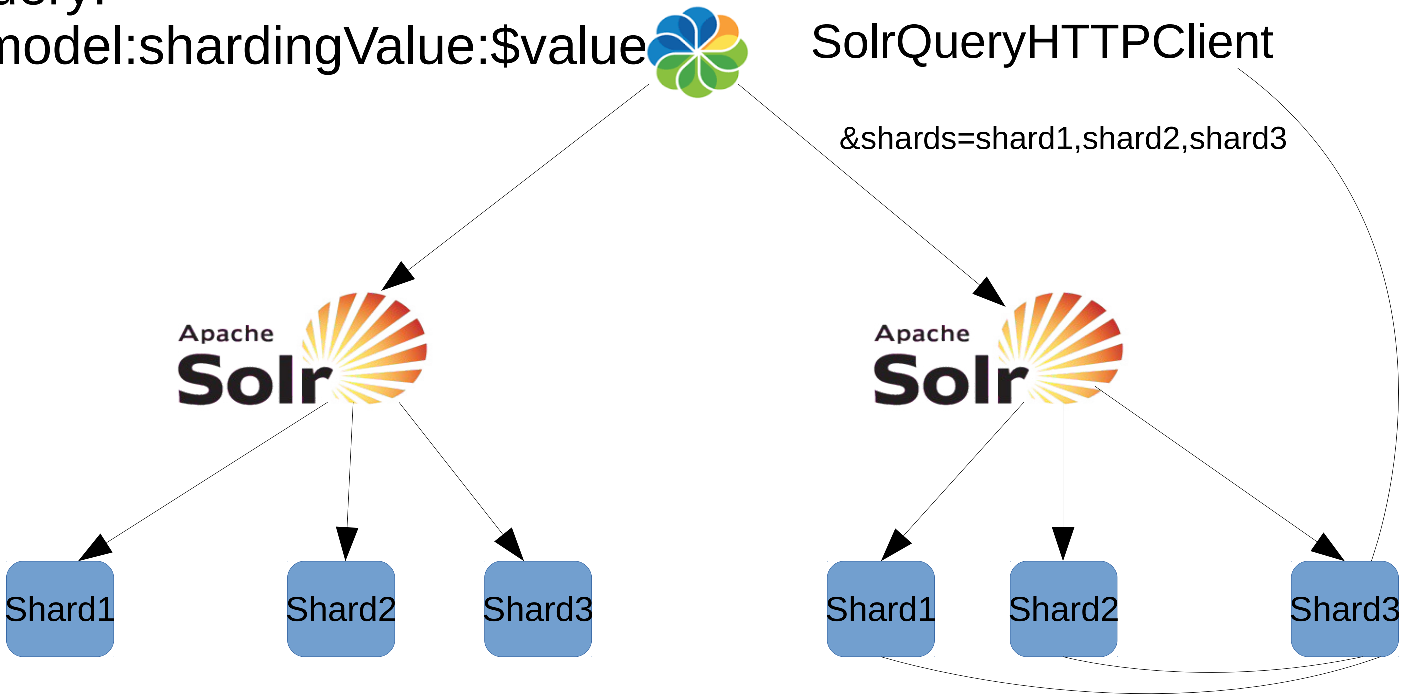
- Implement a new sharding method: by property (now existing in Alfresco 5.2)
- Create a “solr lab”
- Define relevant parameters, metrics
- Design relevant experiments
- Execute experiments
- Monitor
- Get results
- Draw conclusions

Sharding by property - Solr side


nodesWithShardedProperty
nodesWithShardedProperty
Apache Solr
Apache Solr
hash(shardedPropertyValue, num shards, shard instance)
hash(shardedPropertyValue, num shards, shard instance)
Shard1
Shard2
Shard3
Shard1
Shard2
Shard3

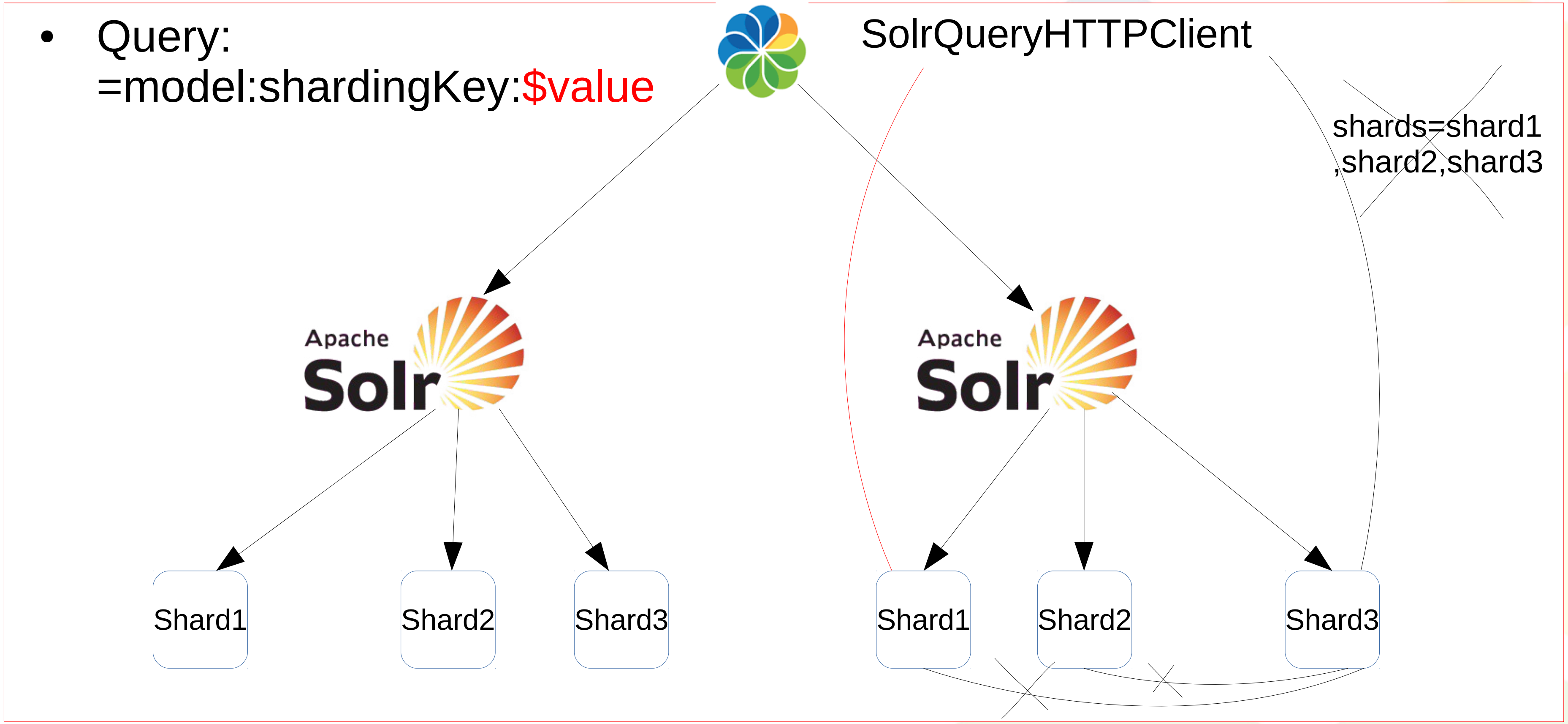
Sharding by property - Alfresco side

- Query: =model:shardingValue:$value



Sharding by property - Alfresco side

- Query: =model:shardingKey:$value

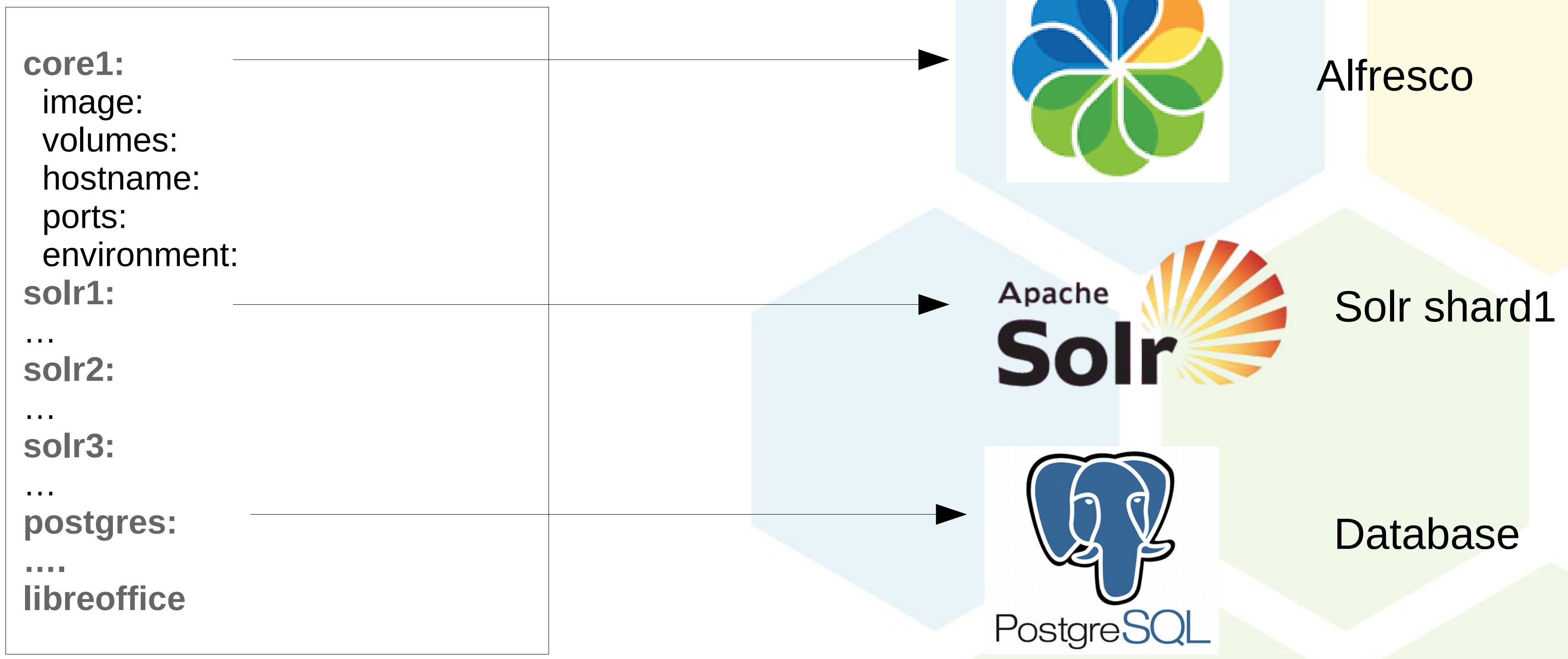


Solr lab

- Requirements
  - Easy to start
  - Easy to replicate
  - Easy to change parameters
  - Easy to resume
  - “Monitorable”

Solr lab

docker-compose.yml



```
core1:
  image:
  volumes:
  hostname:
  ports:
  environment:
solr1:
…
solr2:
…
solr3:
…
postgres:
….
libreoffice
```

Solr lab

core1:

```
image: image: hub.xenit.eu/alfresco-ethias-sharded-by-property:build-3
volumes:
- alfresco10milnoacls:/opt/alfresco/alf_data
- ./license:/opt/alfresco/license
hostname:
ports:
environment:
…….
- DB_NAME=alfresco
- DB_USERNAME=alfresco
- DB_PASSWORD=admin
- DB_URL=jdbc:postgresql://${NET_PRIVATE_IP}:7432/alfresco
- SOLR_HOST=solr1
- SOLR_PORT=8081
- ALFRESCO_HOST=core1
- ALFRESCO_PORT=8080
- JAVA_XMS=6144M
- JAVA_XMX=6144M
- ENABLE_CLUSTERING=true
- DYNAMIC_SHARD_REGISTRATION=true
- SOLR_SSL=none
- SERVICE_8080_NAME=alfresco
- SERVICE_8080_TAGS=proxy-http
- LIBREOFFICE_HOST=libreoffice
- LIBREOFFICE_PORT=8997
```

```
- DEBUG=false
- JMX_ENABLED=true
- JMX_RMI_HOST=xxx.xxx.xxx.xxx
- GLOBAL_shardedProperty.qname={http://www.ethias.be/model/content}shardingKey
- GLOBAL_cache.node.nodesSharedCache.maxItems=1250000
- GLOBAL_cache.node.nodesSharedCache.timeToLiveSeconds=3600
- GLOBAL_cache.node.aspectsSharedCache.maxItems=650000
- GLOBAL_cache.node.aspectsSharedCache.timeToLiveSeconds=3600
- GLOBAL_cache.node.propertiesSharedCache.maxItems=650000
- GLOBAL_cache.node.propertiesSharedCache.timeToLiveSeconds=3600
```

Solr lab

solr1:

```
image: hub.xenit.eu/solr-sharded-by-property:5.1
volumes:
 - solr10milnoacls1index:/opt/alfresco/alf_data
 - solr10milnoacls1conf:/opt/alfresco/solr4
 - ./log4j-solr.properties:/opt/alfresco/solr4/log4j-solr.properties
ports:
hostname: solr1
environment:
 - ALFRESCO_HOST=core1
 - ALFRESCO_PORT=8080
 - SOLR_HOST=solr1
 - SOLR_PORT=8081
 - JAVA_XMS=7168M
 - JAVA_XMX=7168M
 - SHARDING=true
 - NUM_SHARDS=3
 - NUM_NODES=3
 - NODE_INSTANCE=1
 - TEMPLATE=rerank
 - SHARD_IDS=0
 - ALFRESCO_SSL=none
 - ARCHIVE_ENABLE_ALFRESCO_TRACKING=false
 - ARCHIVE_INDEX_CONTENT=false
 - ALFRESCO_INDEX_CONTENT=false
 - MAX_HTTP_HEADER_SIZE=65536
 - JMX_ENABLED=true
 - JMX_RMI_HOST=136.243.138.174
```

Experiments - dimensions

Size of repository: 1mil, 10mil, more

Sharding method: acl, property

Acls: with / without artificial acls for a more fair distribution of documents in the acl sharding method

Number of shards

Types of queries: simple, complex (booleans + facets)

Garbage collector used

Concurrency

With / without loading in the same time

Experiments - metrics

- Important to measure
  - Response times
  - Memory usage
  - Load
  - Caches utilization
  - Database performance

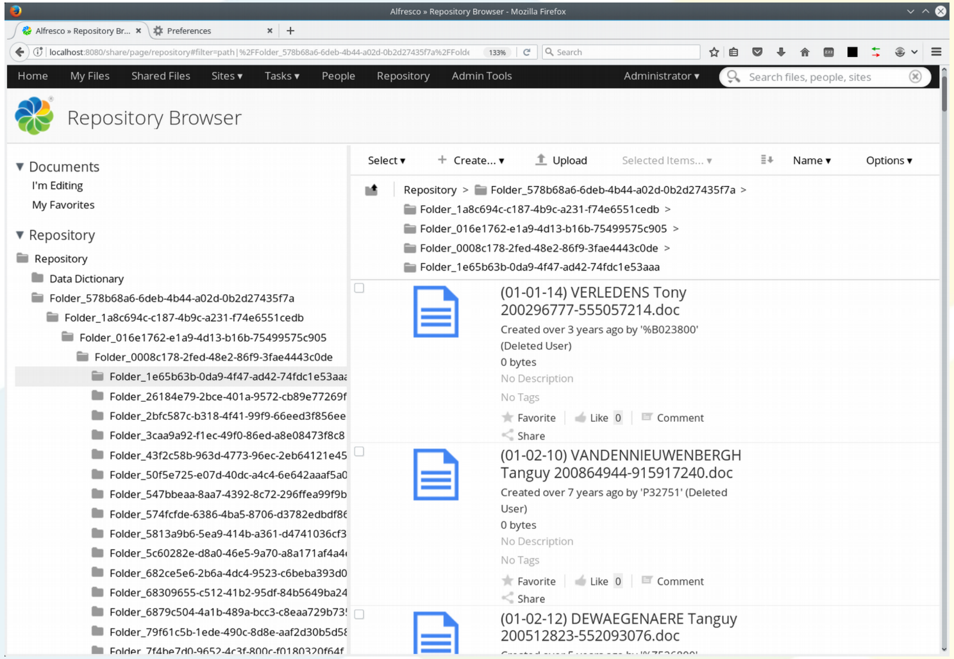
Experiments - hardware

- Started experiments at Amazon, with AuroraDB as database
- Moved to Hetzner dedicated servers, tried multiple configurations, stabilized to: https://www.hetzner.de/nl/hosting/produkte_rootserver/px121ssd
  - Intel(R) Xeon(R) CPU E5-1650 v3 @ 3.50GHz, 12 cores
  - 264GB RAM
  - 2TB SSD disks

		Database	Index 1 shard
1 mil	1 shard	8.7G	5.1G
	3 shard		1.8G
10 mil	1 shard	78G	51G
	3 shard		17G
	11 shard		4.8G
100 mil	3 shard	1.3T	164G

Experiments - loading documents

- No content
- Custom model with >10 metadata fields, values randomly generated based on constraints or data type
- Folder structure with / without artificial acls
- Load speed at 12 threads:
  - Approx 1.3 mil docs / hour
- Solr indexing real time


Alfresco » Repository Browser - Mozilla Firefox
Alfresco » Repository Br...
Preferences
localhost:8080/share/page/repository#filter=path|%2FFolder_578b68a6-6deb-4b44-a02d-0b2d27435f7a%2FFolde
133%
Search
Home
My Files
Shared Files
Sites ▾
Tasks ▾
People
Repository
Admin Tools
Administrator ▾
Search files, people, sites
Select ▾
Create... ▾
Upload
Selected Items... ▾
Name ▾
Options ▾

Repository Browser

Documents
I'm Editing
My Favorites

Repository
Repository
Data Dictionary
Folder_578b68a6-6deb-4b44-a02d-0b2d27435f7a
Folder_1a8c694c-c187-4b9c-a231-f74e6551cedb
Folder_016e1762-e1a9-4d13-b16b-75499575c905
Folder_0008c178-2fed-48e2-86f9-3fae4443c0de
Folder_1e65b63b-0da9-4f47-ad42-74fdc1e53aa
Folder_26184e79-2bce-401a-9572-cb89e77269f
Folder_2bfc587c-b318-4f41-99f9-66eed3f856ee
Folder_3caa9a92-f1ec-49f0-86ed-a8e08473f8c8
Folder_43f2c58b-963d-4773-96ec-2eb64121e45
Folder_50f5e725-e07d-40dc-a4c4-6e642aaaf5a0
Folder_547bbeaa-8aa7-4392-8c72-296ffea99f9b
Folder_574fcfde-6386-4ba5-8706-d3782edbdf8
Folder_5813a9b6-5ea9-414b-a361-d4741036cf3
Folder_5c60282e-d8a0-46e5-9a70-a8a171af4a4
Folder_682ce5e6-2b6a-4dc4-9523-c6beba393d0
Folder_68309655-c512-41b2-95df-84b5649ba24
Folder_6879c504-4a1b-489a-bcc3-c8eaa729b73
Folder_79f61c5b-1ede-490c-8d8e-aaf2d30b5d5
Folder_7f4be7d0-9652-4c3f-800c-f0180320f64f

Repository > Folder_578b68a6-6deb-4b44-a02d-0b2d27435f7a >
Folder_1a8c694c-c187-4b9c-a231-f74e6551cedb >
Folder_016e1762-e1a9-4d13-b16b-75499575c905 >
Folder_0008c178-2fed-48e2-86f9-3fae4443c0de >
Folder_1e65b63b-0da9-4f47-ad42-74fdc1e53aaa

(01-01-14) VERLEDENS Tony 200296777-555057214.doc
Created over 3 years ago by '%B023800' (Deleted User)
0 bytes
No Description
No Tags
Favorite | Like 0 | Comment
Share

(01-02-10) VANDENNIEUWENBERGH Tanguy 200864944-915917240.doc
Created over 7 years ago by 'P32751' (Deleted User)
0 bytes
No Description
No Tags
Favorite | Like 0 | Comment
Share

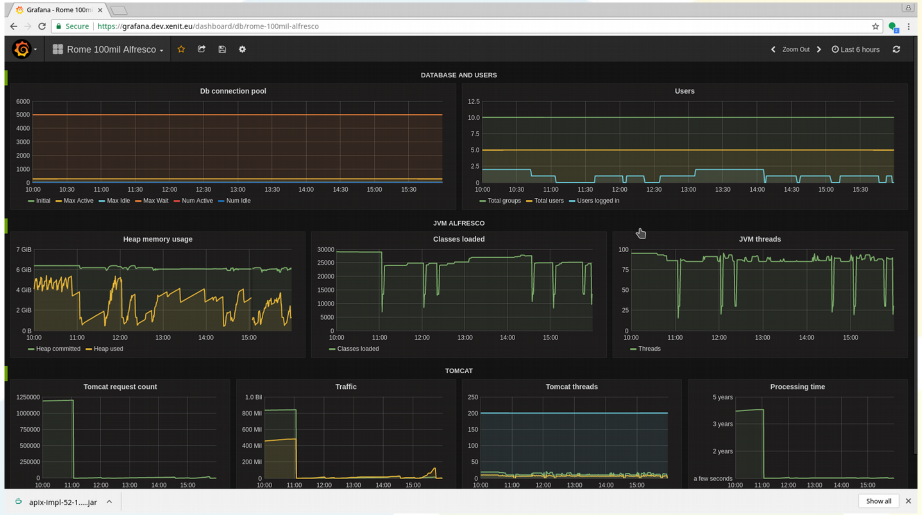
(01-02-12) DEWAEGENAERE Tanguy 200512823-552093076.doc

Experiments – problems encountered

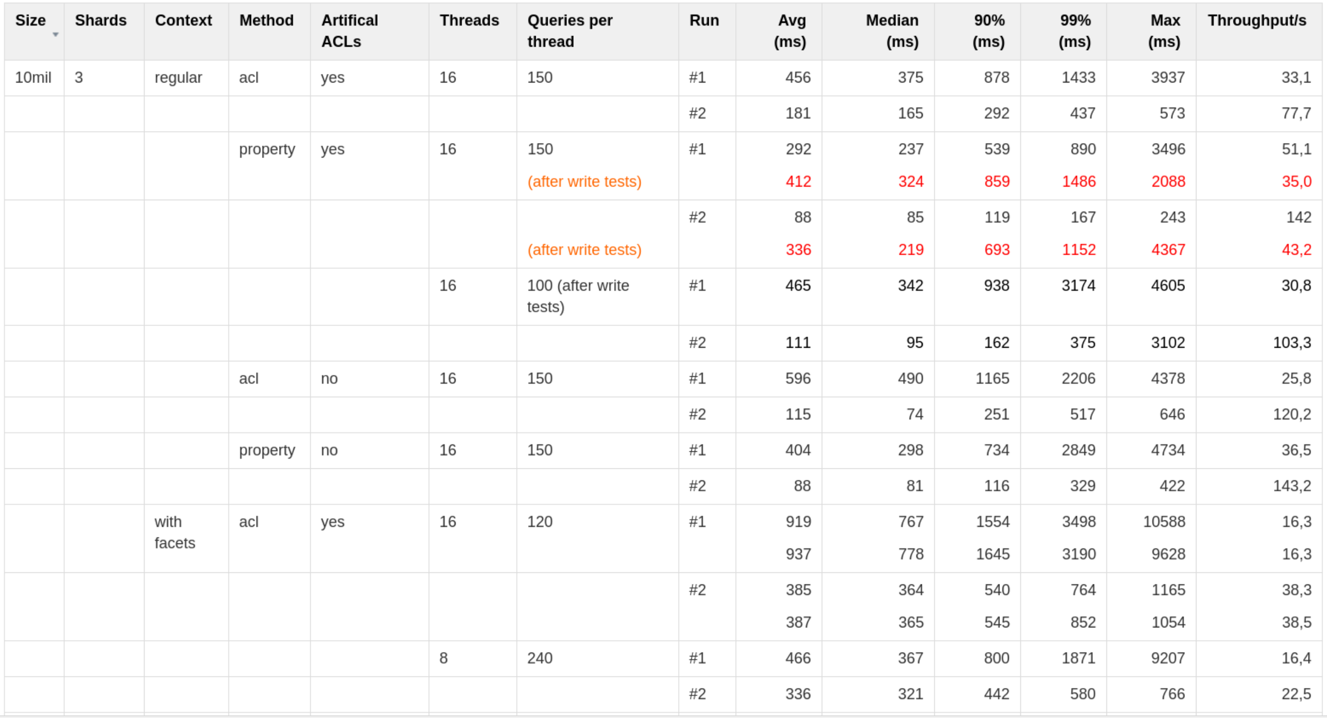
- Ran out of inodes on solr' partition – needed to reformat the partition with small block size
- Db tunning – max connections
- Memory tunning on db, alfresco + solr size
- Increased max header size in tomcat

Monitoring

- Jmeter
- Jvisualvm
- Grafana stack
  - Jmxtrans to collect jmx metrics
  - Grafana container



Results



Size	Shards	Context	Method	Artifical ACLs	Threads	Queries per thread	Run	Avg (ms)	Median (ms)	90% (ms)	99% (ms)	Max (ms)	Throughput/s
10mil	3	regular	acl	yes	16	150	#1	456	375	878	1433	3937	33,1
							#2	181	165	292	437	573	77,7
			property	yes	16	150	#1	292	237	539	890	3496	51,1
						(after write tests)		412	324	859	1486	2088	35,0
							#2	88	85	119	167	243	142
						(after write tests)		336	219	693	1152	4367	43,2
					16	100 (after write tests)	#1	465	342	938	3174	4605	30,8
							#2	111	95	162	375	3102	103,3
			acl	no	16	150	#1	596	490	1165	2206	4378	25,8
							#2	115	74	251	517	646	120,2
			property	no	16	150	#1	404	298	734	2849	4734	36,5
							#2	88	81	116	329	422	143,2
		with facets	acl	yes	16	120	#1	919	767	1554	3498	10588	16,3
								937	778	1645	3190	9628	16,3
							#2	385	364	540	764	1165	38,3
								387	365	545	852	1054	38,5
					8	240	#1	466	367	800	1871	9207	16,4
							#2	336	321	442	580	766	22,5

Conclusions

- Search slow?
- Primary culprit: DB access / cache
  - Expensive bulk load
  - Cache “sabotage”: nodesSharedCache TTL (used OOTBee Support Tools for insights)

Conclusions

- Search slow?
- Primary culprit: DB access / cache
  - Expensive bulk load
  - Cache “sabotage”: nodesSharedCache TTL
    (used OOTBee Support Tools for insights)
- Secondary culprit: Repository-tier ACL checking
  - Misleading: PermissionEvaluationMode.NONE
  - Low default size of caches for readers/readersDenied

Conclusions

- Natural best performer: property sharding + shard-targeted queries
  - ~20% (+-10%) better general performance
  - Improved scaling for parallel requests

Further work

- In progress: 100 million documents
  - Extended data model
  - Realistic value distribution for sharding property

- Planned: customer reference setup
  - Ideally: 100% live data clone, augmented via generation
  - Oracle instead of PostgreSQL

Speaker contacts

roxana.angheluta@xenit.eu
axel.faust@acosix.de