

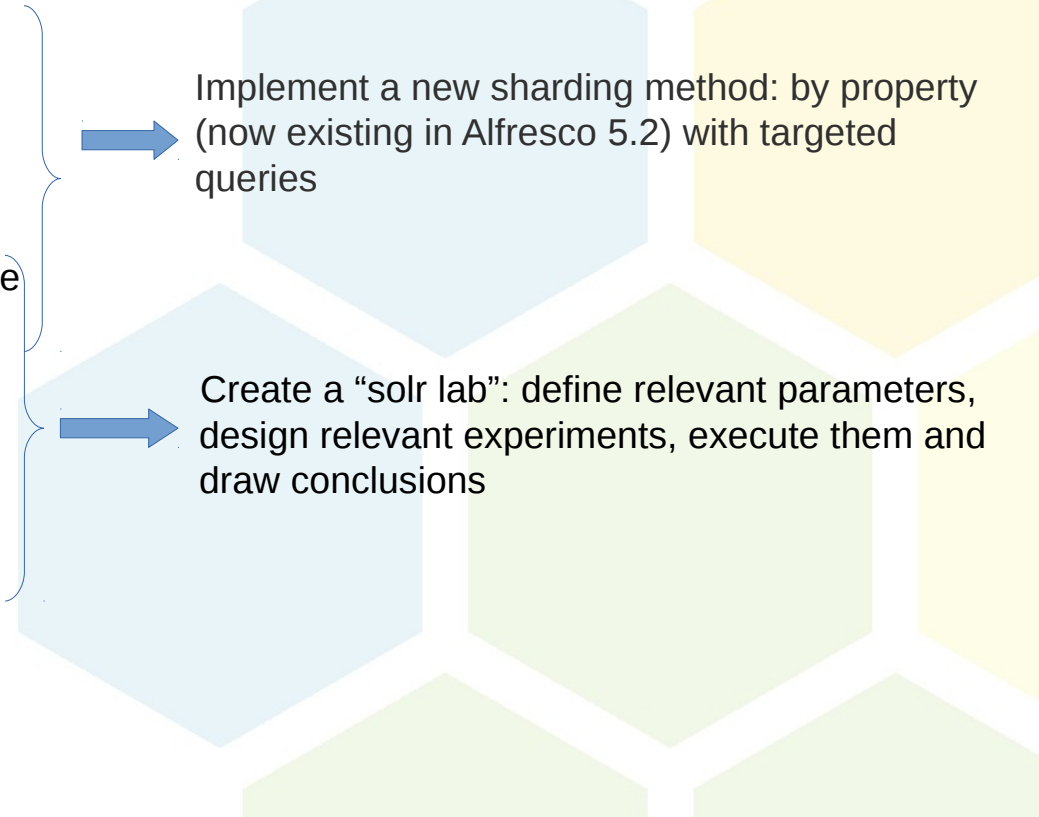
# Solr lab for sharding experiments

Roxana Angheluta (Xenit Solutions)

Axel Faust (Acosix GmbH)

# Motivation and solution

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

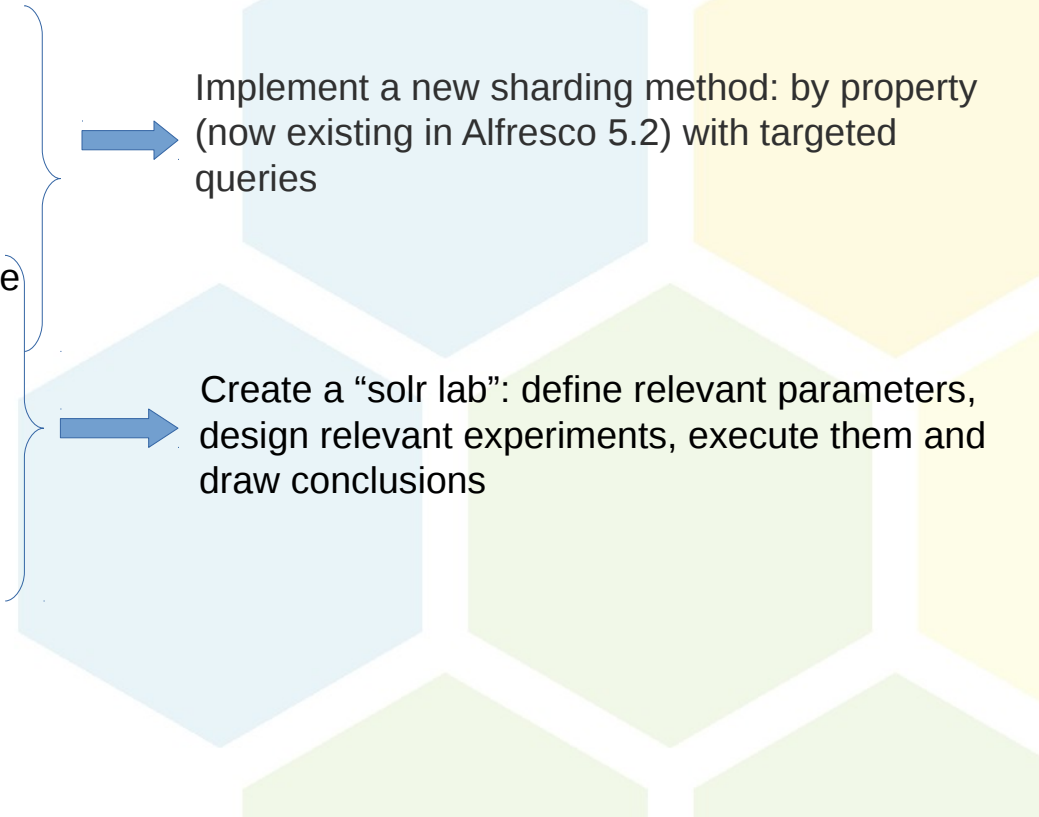


Implement a new sharding method: by property (now existing in Alfresco 5.2) with targeted queries

Create a “solr lab”: define relevant parameters, design relevant experiments, execute them and draw conclusions

# Motivation and solution

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

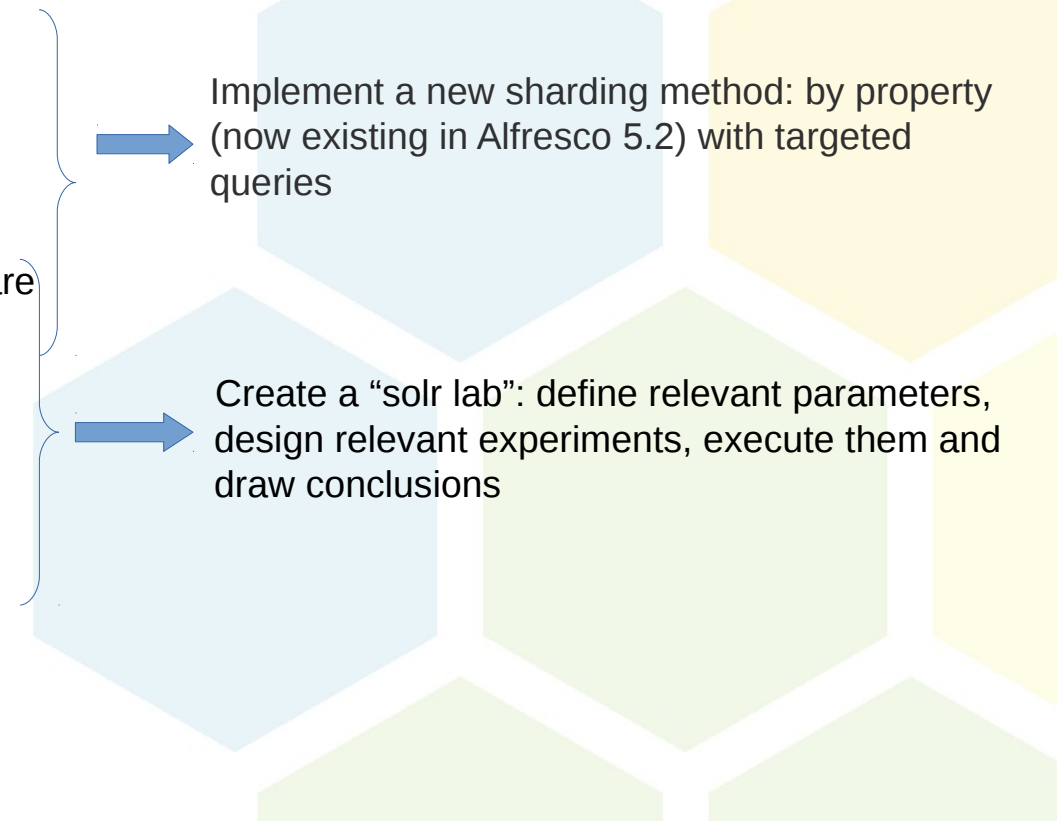


Implement a new sharding method: by property (now existing in Alfresco 5.2) with targeted queries

Create a “solr lab”: define relevant parameters, design relevant experiments, execute them and draw conclusions

# Motivation and solution

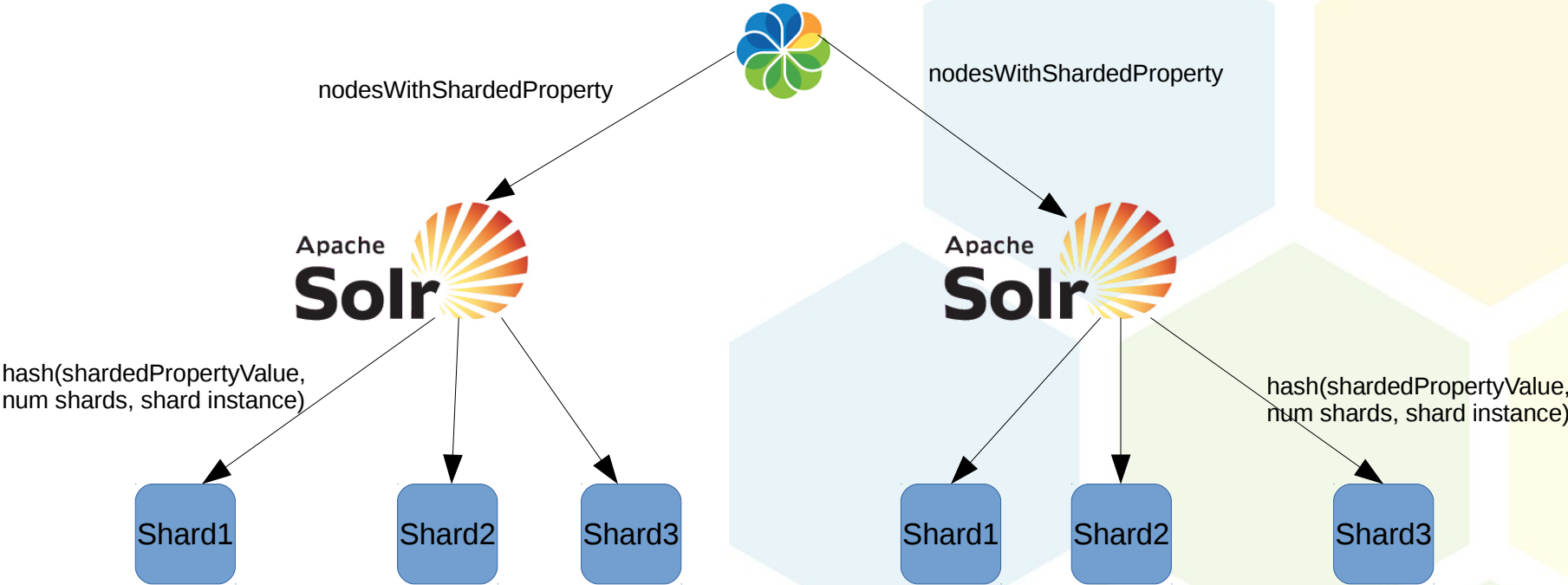
- Large archives with >50mil documents
- No sites
- Particular use case: most of the queries are targeted to a specific client id
- Search speed <3s



Implement a new sharding method: by property (now existing in Alfresco 5.2) with targeted queries

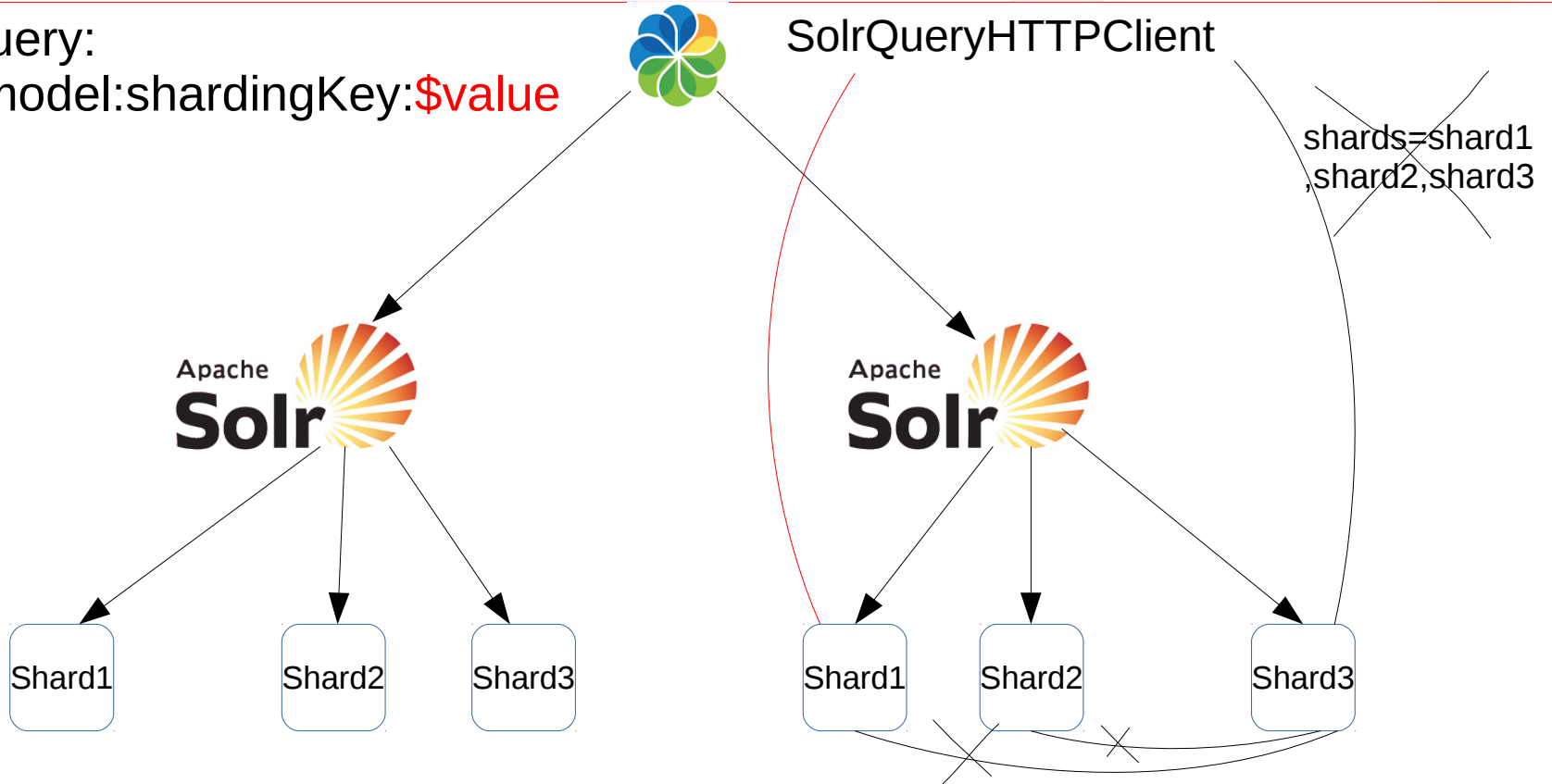
Create a “solr lab”: define relevant parameters, design relevant experiments, execute them and draw conclusions

# Sharding by property - Solr side



# 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
  - Easy to monitor



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



# Solr lab

- Requirements
  - Easy to start
  - Easy to replicate
  - Easy to change parameters
  - Easy to resume
  - Easy to monitor



```
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:

- alfresco10milnoacsl:/opt/alfresco/alf\_data

environment:

- 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
  - 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

.....

## solr1:

image: hub.xenit.eu/solr-sharded-by-property:5.1

volumes:

- solr10milnoacsl1index:/opt/alfresco/alf\_data

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=xxx.xxx.xxx.xxx

.....

# Solr lab

## core1:

image: image: hub.xenit.eu/alfresco-ethias-sharded-by-property:build-3

volumes:

- alfresco10milnoacls:/opt/alfresco/alf\_data

environment:

.....

- 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
  - 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

## solr1:

image: hub.xenit.eu/solr-sharded-by-property:5.1

volumes:

- solr10milnoacls1index:/opt/alfresco/alf\_data

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=xxx.xxx.xxx.xxx

# Experiments – dimensions and metrics

Dimension	Values		
Size of repo	1mil	10mil	more
Sharding method	property	acl	
Number of shards	1	3	11
Acls	With artificial acls	Without artificial acls	
Types of queries	simple	complex	
Query concurrency	yes	no	
Load in the same time	yes	no	

- Response times
- Memory usage
- Load
- Caches utilization
- Database performance

# Experiments – dimensions and metrics

Dimension	Values		
Size of repo	1mil	10mil	more
Sharding method	property	acl	
Number of shards	1	3	11
Acls	With artificial acls	Without artificial acls	
Types of queries	simple	complex	
Query concurrency	yes	no	
Load in the same time	yes	no	

- Response times
- Memory usage
- Load
- Caches utilization
- Database performance

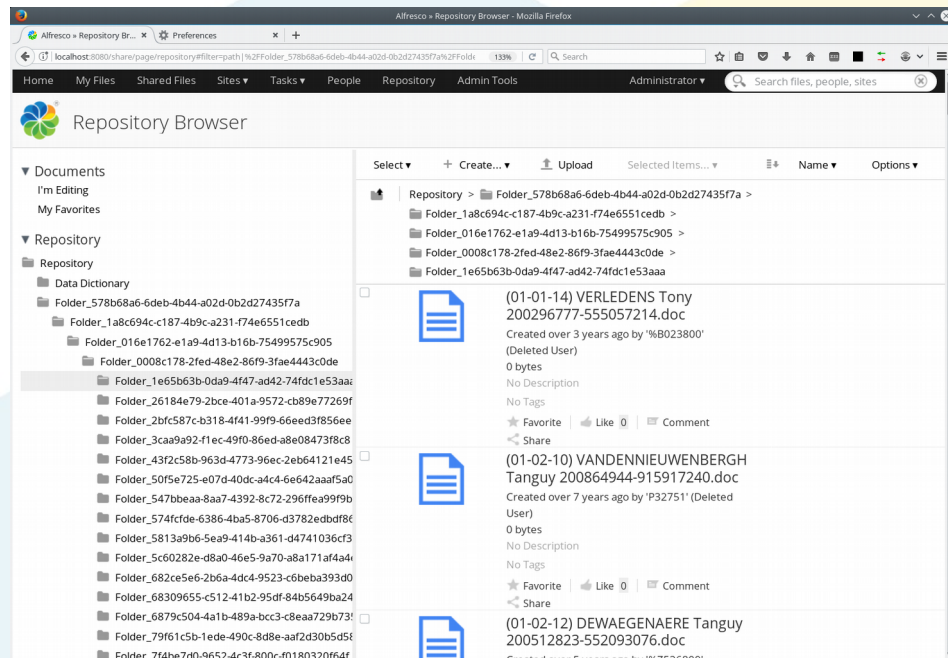
# Experiments - hardware

- 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](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	1 shard	1.3T	500G
	3 shard		164G

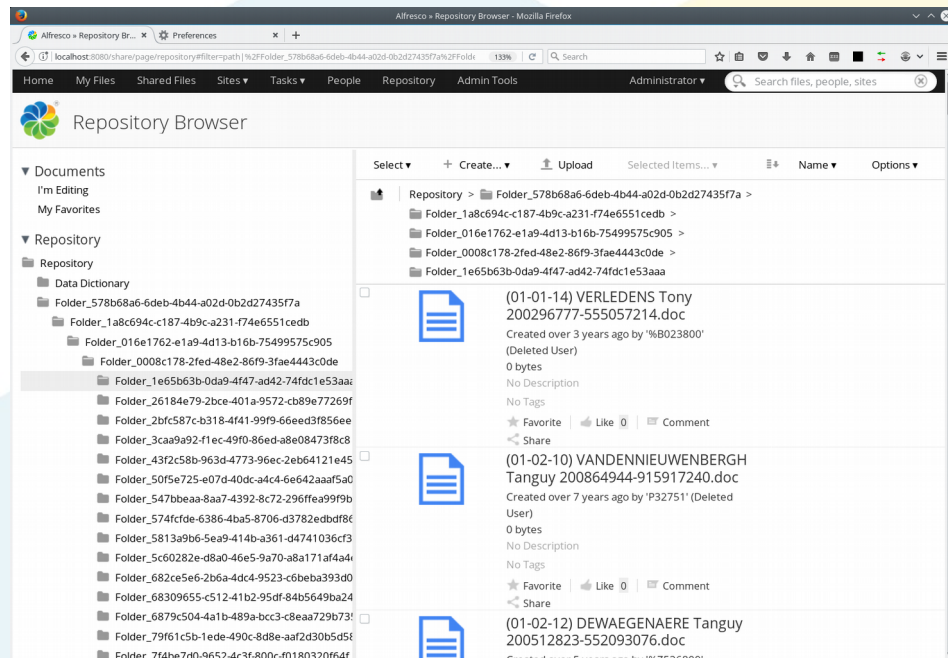
# Experiments - loading documents

- 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



# 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



# Experiments – problems encountered

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

Performance improvements:

- Disabling acl-checks on Alfresco's side, post-searching



# Results

Size	Shards	Context	Method	Artificial 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 (after write tests)	#1	292	237	539	890	3496	51,1
							#2	412	324	859	1486	2088	35,0
		regular				100 (after write tests)	#1	88	85	119	167	243	142
							#2	336	219	693	1152	4367	43,2
			acl	no	16	150	#1	465	342	938	3174	4605	30,8
							#2	111	95	162	375	3102	103,3
		with facets	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
							#2	937	778	1645	3190	9628	16,3
							#1	385	364	540	764	1165	38,3
							#2	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](mailto:roxana.angheluta@xenit.eu)

[axel.faust@acosix.de](mailto:axel.faust@acosix.de)

