Many Members Performance Testing

- Introduction
- Optimization Strategies
- Conclusions
- Setup:
- Instance Definition
- n-children.sh
- n-members.sh
- n-uris.sh
- n-binaries.sh (time for loading binary files)
- Conclusions

Introduction

The Fedora Community has raised significant concerns about retrieval times for resources with many members. By "members" we are talking about at least three scenarios

1. Many children
2. Many hasMember outlinks
3. Many memberOf inlinks

Using a set of tests originally devised by Esmé Cowles I modified them slightly and added a couple of new tests. The latest versions can be found here : te st scripts. The test results, in most cases, reflect that fastest retrieval speeds I was able to get. Initial retrieval times tended to be slower in most cases due to the fact that modeshape needs to warm up the cache. The scripts now make an initial GET call to warm the cache before timing a second call to the same resource.

Optimization Strategies

I pursued three main strategies for improving the performance of the "many members" use cases. I first tried configuring custom modeshape indexes in conjunction with attempts to pull the member ids using targeted jcr-sql2 queries that queried specific property rather than iterating through nodes. The hope here was that by using jcr-sql2 we could by pass the potentially costly operation of loading the node before accessing a property on that node. No matter how I went about it, it seemed that this approach surprisingly did not move the needle at all. In some cases it look like it made the problem slightly worse. Eventually I abandoned this line of thinking. The next thing I tried was to "turn on" parallel processing of streams. This required a minor tweak to the code and promised significant improvement on multiprocessor machines. Finally I tried setting the cacheSize param in repository.json to a large number in the hopes that this might improve the performance.

Conclusions

Looking at the hasMember and memberOf use cases it would seem that the most significant improvements could be seen by combining the parallelization changes with the increased cacheSize using a local postgresql database. hasMember improved by 5x while memberOf performance improved by a 8x.

Setup:

In order to set up the machines for the tests, I performed the following steps.

1. Install postgresql
2. Install mysql-server
3. Build the fcrepo4 project from the specified commit.
4. Run fcrepo-webapp using mvn clean jetty:run and the appropriate MAVEN_OPTS such as fcrepo.modeshape.configuration, fcrepo.mysql. username, fcrepo.mysql.password, fcrepo.postgres.username, fcrepo.postgres.password.
5. git clone https://github.com/fcrepo4-labs/fcrepo-performance-test-scripts.git
6. For each test, I ran the scripts with 1000 and 10000 items.

Instance Definition

1. AWS / Ubuntu 16 / Oracle Java 8 / m3.medium (3.7 GiB memory / Intel Xeon E5-2670 (Sandy Bridge) Processor @ 2.6 GHz x 1)
2. Lenovo / Ubuntu 16.10 / Java HotSpot 1.8.0_111 / 11.6 GiB memory / Intel i7-4600U CPU @ 2.10GHz x 4
3. AWS / Ubuntu 16 / Oracle Java 8 / m3.xlarge (14 GiB memory / Intel Xeon E5-2670 (Sandy Bridge) Processor @ 2.6 GHz x 4)
4. AWS / Ubuntu 16 / Oracle Java 8 / c4.xlarge (7.5 GiB memory / Intel Xeon E5-2666 v3 (Haswell) @ 2.9 GHz x 4)
5. AWS / Ubuntu 16 / Oracle Java 8 / c4.2xlarge (15 GiB memory /Intel Xeon E5-2666 v3 (Haswell) @ 2.9 GHz x 8)
6. AWS / Ubuntu 16 / Oracle Java 8 / c4.4xlarge (30 GiB memory /Intel Xeon E5-2666 v3 (Haswell) @ 2.9 GHz x 16)

n-children.sh

FCREPO Version	Repo	Branch	Commit	Environment	Modeshape Config	# of relations	Test Duration (seconds)	Tester	
4.8.0-SNAPSHOT	dbernstein		b60d4e	5	file-simple	10000	5.222	Danny Bernstein	
4.8.0-SNAPSHOT	dbernstein		daa11f3	5	file-simple	10000	5.567	Danny Bernstein	
4.8.0-SNAPSHOT	dbernstein		b60d4e	6	file-simple	10000	4.859	Danny Bernstein	
4.8.0-SNAPSHOT	dbernstein		daa11f3	6	file-simple	10000	4.841	Danny Bernstein	

n-members.sh

FCREPO Version	Repo	Branch	Commit	modeshape	Environment	# of relations	Test Duration (seconds)	Tester	
4.8.0-SNAPSHOT	fcrepo4	master	2df32	file-simple	1	1000	1.408	Danny Bernstein	
4.7.1	fcrepo4	4.7.1	546f5a5	file-simple	2	1000	1.45	Andrew Woods	
4.8.0-SNAPSHOT	fcrepo4	master	2df32	file-simple	1	10,000	28.583	Danny Bernstein	
4.7.1	fcrepo4	4.7.1	546f5a5	file-simple	2	10,000	24.79	Andrew Woods	
4.8.0-SNAPSHOT	fcrepo4	master	b60d4e	file-simple	3	10,000	9.58	Danny Bernstein	
4.8.0-SNAPSHOT	dbernstein	fcrepo-2105-v4-parallelization	daa11f3	file-simple	3	10,000	12.16	Danny Bernstein	parallel streams enabled.
4.8.0-SNAPSHOT	dbernstein	fcrepo-2105-v4-parallelization	daa11f3	file-simple	4	10,000	9.71	Danny Bernstein	parallel streams enabled.
4.8.0-SNAPSHOT	dbernstein	fcrepo-2105-v4-parallelization	daa11f3	file-simple	5	10,000	8.51	Danny Bernstein	parallel streams enabled.
4.8.0-SNAPSHOT			b60d4e	file-simple	5	10,000	8.815	Danny Bernstein	
4.8.0-SNAPSHOT	dbernstein	fcrepo-2105-v4-parallelization	daa11f3	file-simple	6	10,000	8.29	Danny Bernstein	parallel streams enabled.
4.8.0-SNAPSHOT			b60d4e	file-simple	6	10,000	8.661	Danny Bernstein	
4.8.0-SNAPSHOT	bbranan	fcrepo-2402	f0a51e	jdbc-postgresql	1	1000	1.543	Danny Bernstein	
4.8.0-SNAPSHOT	bbranan	fcrepo-2402	f0a51e	jdbc-postgresql	1	10,000	61.381	Danny Bernstein	perhaps postgres needs caching configured?
4.8.0-SNAPSHOT	dbernstein	fcrepo-2105-v4-parallelization	daa11f3	jdbc-postgresql	3	1000	0.592	Danny Bernstein	
4.8.0-SNAPSHOT	dbernstein	fcrepo-2105-v4-parallelization	daa11f3	jdbc-postgresql	3	10,000	39.671	Danny Bernstein	
4.8.0-SNAPSHOT	bbranan	fcrepo-2402	f0a51e	jdbc-postgresql	3	1000	4.435	Danny Bernstein	
4.8.0-SNAPSHOT	bbranan	fcrepo-2402	f0a51e	jdbc-postgresql	3	10,000	39.486	Danny Bernstein	
4.8.0-SNAPSHOT	dbernstein	fcrepo-2105-large-cache	f453a	jdbc-postgresql	3	1000	0.561	Danny Bernstein	1 million item cache
4.8.0-SNAPSHOT	dbernstein	fcrepo-2105-large-cache	f453a	jdbc-postgresql	3	10,000	5.334	Danny Bernstein	1 million item cache
4.8.0-SNAPSHOT	dbernstein	fcrepo-2105-large-cache	5138b4	jdbc-postgresql	3	1000	0.633	Danny Bernstein	1 million item cache + parallelized
4.8.0-SNAPSHOT	dbernstein	fcrepo-2105-large-cache	5138b4	jdbc-postgresql	3	10,000	5.398	Danny Bernstein	1 million item cache + parallelized
4.8.0-SNAPSHOT	dbernstein	fcrepo-2105-large-cache	5138b4	mysql-postgresql	3	1000	0.820	Danny Bernstein	1 million item cache + parallelized
4.8.0-SNAPSHOT	dbernstein	fcrepo-2105-large-cache	5138b4	mysql-postgresql	3	10,000	7.726	Danny Bernstein	1 million item cache + parallelized
4.8.0-SNAPSHOT	dbernstein	fcrepo-2402	75dd1	jdbc-postgresql-s3	3	1000	0.701	Danny Bernstein	1 million item cache + parallelized
4.8.0-SNAPSHOT	dbernstein	fcrepo-2402	75dd1	jdbc-postgresql-s3	3	10,000	5.485	Danny Bernstein	1 million item cache + parallelized
4.8.0-SNAPSHOT	dbernstein	fcrepo-2402	75dd1	jdbc-mysql-s3	3	1000	0.864	Danny Bernstein	1 million item cache + parallelized
4.8.0-SNAPSHOT	dbernstein	fcrepo-2402	75dd1	jdbc-mysql-s3	3	10,000	7.395	Danny Bernstein	1 million item cache + parallelized

4.7.1	fcrepo4	4.7.1	546f5a5	file-simple	2	10,000	13.07	Andrew Woods	cacheSize = 50,000
4.7.1	fcrepo4	4.7.1	546f5a5	file-simple	2	10,000	10.30	Andrew Woods	cacheSize = 1,000,000

n-memberof.sh

Number of relations: 1000

FCREPO Version	Repo	branch	Commit	modeshape config	Environment	# of relations	Test Duration (seconds)	Tester	Notes
4.8.0-SNAPSHOT	fcrepo4	master	2df32	file-simple	1	1000	6.570	Danny Bernstein	
4.7.1	fcrepo4	4.7.1	546f5a5	file-simple	2	1000	2.80	Andrew Woods	
4.8.0-SNAPSHOT	fcrepo4	master	2df32	file-simple	1	10,000	86.000	Danny Bernstein	
4.7.1	fcrepo4	4.7.1	546f5a5	file-simple	2	10,000	57.35	Andrew Woods	
4.8.0-SNAPSHOT	fcrepo4	master	b60d4e	file-simple	3	10,000	30.02	Danny Bernstein	Unlike the n-member example, results begin streaming right away - so the response begins streaming within 2 seconds.
4.8.0-SNAPSHOT	dbernstein	fcrepo-2105-v4-parallelization	daa11f3	file-simple	3	10,000	29.975	Danny Bernstein	parallel streams enabled.
4.8.0-SNAPSHOT	dbernstein	fcrepo-2105-v4-parallelization	daa11f3	file-simple	4	10,000	24.790	Danny Bernstein	parallel streams enabled.
4.8.0-SNAPSHOT	dbernstein	fcrepo-2105-v4-parallelization	daa11f3	file-simple	5	10,000	20.992	Danny Bernstein	parallel streams enabled.
4.8.0-SNAPSHOT	fcrepo4	master	b60d4e	file-simple	5	10,000	26.357	Danny Bernstein	
4.8.0-SNAPSHOT	dbernstein	fcrepo-2105-v4-parallelization	daa11f3	file-simple	6	10,000	20.337	Danny Bernstein	parallel streams enabled.
4.8.0-SNAPSHOT	fcrepo4	master	b60d4e	file-simple	6	10,000	25.782	Danny Bernstein	
4.8.0-SNAPSHOT	bbranan	fcrepo-2402	f0a51e	jdbc-postgresql	1	1000	11.414	Danny Bernstein	
4.8.0-SNAPSHOT	bbranan	fcrepo-2402	f0a51e	jdbc-postgresql	1	10,000	194	Danny Bernstein	
4.8.0-SNAPSHOT	dbernstein	crepo-2105-v4-parallelization	daa11f3	jdbc-postgresql	3	1000	3.961	Danny Bernstein	parallel streams enabled.
4.8.0-SNAPSHOT	dbernstein	crepo-2105-v4-parallelization	daa11f3	jdbc-postgresql	3	10,000	53.452	Danny Bernstein	parallel streams enabled.
4.8.0-SNAPSHOT	bbranan	fcrepo-2402	f0a51e	jdbc-postgresql	3	1000	10.833	Danny Bernstein	
4.8.0-SNAPSHOT	bbranan	fcrepo-2402	f0a51e	jdbc-postgresql	3	10,000	109.530	Danny Bernstein	
4.8.0-SNAPSHOT	dbernstein	fcrepo-2105-large-cache	f453a	jdbc-postgresql	3	10000	22.963	Danny Bernstein	1 million item cache.
4.8.0-SNAPSHOT	dbernstein	fcrepo-2105-v4-parallelization	5138b4	jdbc-postgresql	3	10,000	11.559	Danny Bernstein	1 million item cache + parallelized
4.8.0-SNAPSHOT	dbernstein	fcrepo-2105-v4-parallelization	5138b4	mysql-postgresql	3	1000	2.337	Danny Bernstein	1 million item cache + parallelized
4.8.0-SNAPSHOT	dbernstein	fcrepo-2105-v4-parallelization	5138b4	mysql-postgresql	3	10,000	14.998	Danny Bernstein	1 million item cache + parallelized
4.8.0-SNAPSHOT	dbernstein	fcrepo-2402	75dd1	jdbc-postgresql-s3	3	1000	1.883	Danny Bernstein	1 million item cache + parallelized
4.8.0-SNAPSHOT	dbernstein	fcrepo-2402	75dd1	jdbc-postgresql-s3	3	10,000		Danny Bernstein	1 million item cache + parallelized
4.8.0-SNAPSHOT	dbernstein	fcrepo-2402	75dd1	jdbc-mysql-s3	3	1000	2.374	Danny Bernstein	1 million item cache + parallelized
4.8.0-SNAPSHOT	dbernstein	fcrepo-2402	75dd1	jdbc-mysql-s3	3	10,000	14.956	Danny Bernstein	1 million item cache + parallelized
4.7.1	fcrepo4	4.7.1	546f5a5	file-simple	2	10,000	31.91	Andrew Woods	cacheSize = 50,000
4.7.1	fcrepo4	4.7.1	546f5a5	file-simple	2	10,000	23.07	Andrew Woods	cacheSize = 1,000,000

n-uris.sh

FCREPO Version	Commit	Environment	# of relations	Test Duration (seconds)	
4.8.0-SNAPSHOT	2df32	1	1000	0.039	Danny Bernstein
4.7.1	546f5a5	2	1000	0.14	Andrew Woods
4.8.0-SNAPSHOT	2df32	1	10,000	0.063	Danny Bernstein
4.7.1	546f5a5	2	10,000	0.17	Andrew Woods

n-properties.sh

FCREPO Version	Commit	Environment	# of relations	Test Duration (seconds)	Tester
4.8.0-SNAPSHOT	2df32	1	1000	0.060	Danny Bernstein
4.7.1	546f5a5	2	1000	0.089	Andrew Woods
4.8.0-SNAPSHOT	2df32	1	10,000	0.119	Danny Bernstein
4.7.1	546f5a5	2	10,000	0.281	Andrew Woods

n-binaries.sh (time for loading binary files)

repo	branch	Commit	Environment	Modeshape Config	# of binaries	size in KB	Test Duration (seconds)	Tester
https://github.com/bbranan/fcrepo4.git	fcrepo-2402	f0a51e	1	file-s3	1000	1000	00:06:02	Danny Bernstein
https://github.com/bbranan/fcrepo4.git	fcrepo-2402	f0a51e	1	file-simple	1000	1000	00:02:16	Danny Bernstein
https://github.com/bbranan/fcrepo4.git	fcrepo-2402	f0a51e	1	jdbc-postgresql	1000	1000	00:02:13	Danny Bernstein
https://github.com/bbranan/fcrepo4.git	fcrepo-2402	f0a51e	3	jdbc-postgresql-s3	1000	1000	00:06:36	Danny Bernstein
https://github.com/bbranan/fcrepo4.git	fcrepo-2402	f0a51e	3	jdbc-mysql-s3	1000	1000	00:06:32	Danny Bernstein

Conclusions