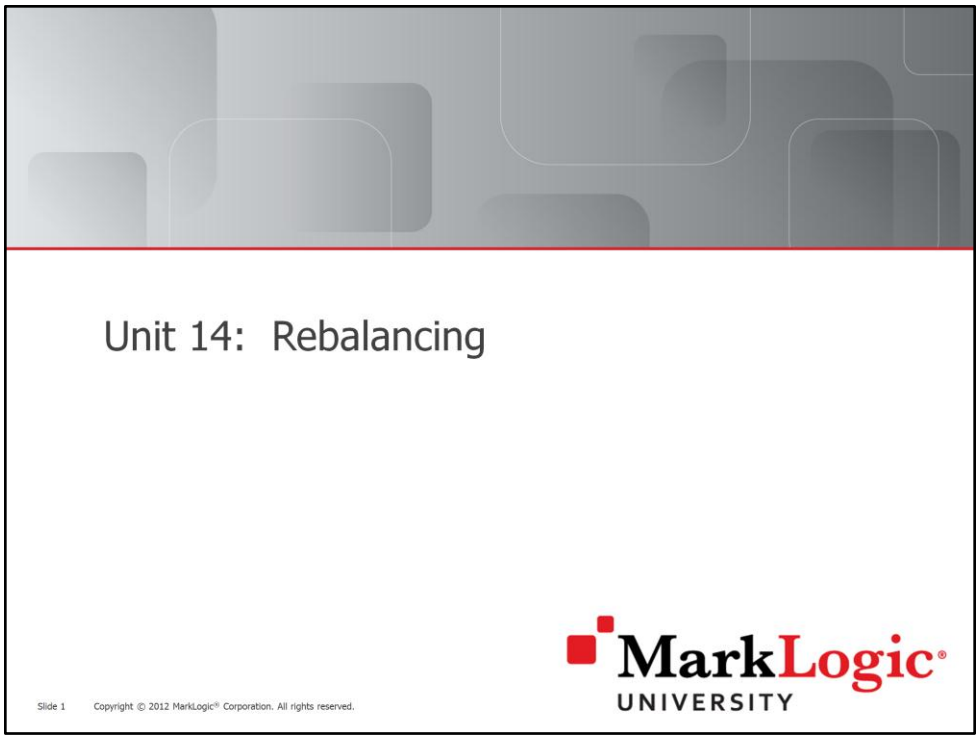


Learning Objectives

- Describe how the rebalancer operates.
- Describe bucket, statistical, range and legacy assignment policies.
- Configure the rebalancer.
- Resize a partition.
- Retire a forest.
- Take a forest or partition online / offline

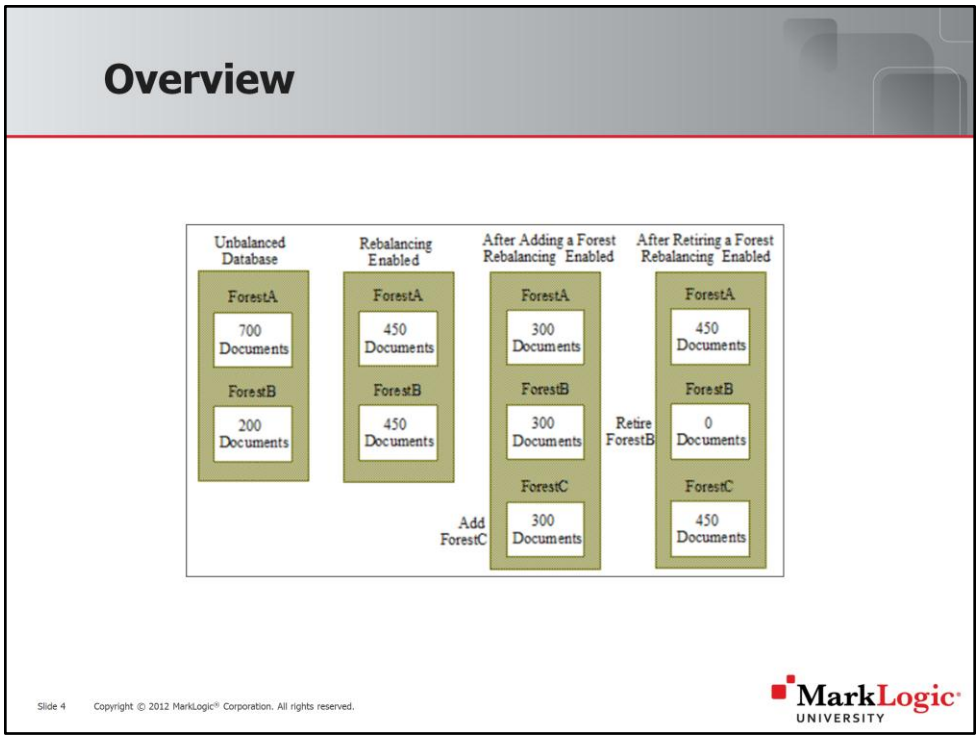
Slide 2

MarkLogic UNIVERSITY

Overview

- Rebalancing Objectives:
  - Easily expand / contract as data needs change
  - Enable an automated, more even distribution of data
  - Optimize use of storage resources
- Rebalancer
  - Moves data around
- Assignment Policy
  - Defines balanced for a given database
  - Determines where docs will land on ingestion
  - Determines where docs will go when you expand / contract
  - Several configurable options
- Rebalancing is on by default for new databases created in ML7
  - For upgraded databases it is off (can turn it on)

MarkLogic UNIVERSITY



Prior to MarkLogic 7, data was ingested in a round robin approach. Over time this yields a generally balanced distribution of data across all hosts in a cluster containing the database and all forests which make up that database. However, as new forests were added, balancing data involved manual redistribution through scripts, etc.

In MarkLogic 7 this rebalancing is automated. So if you need to expand, simply add a forest and the data will be distributed according to the assignment policy. If you need to contract, simply retire a forest, and likewise the data will be rebalanced among the remaining forests automatically.

What Causes Rebalancing?

- Trigger Events
  - Database changes like adding a new or retiring an existing forest
  - Completion of a backup or restore operation
- Document loads (inserts)
  - Don't trigger rebalancing
  - Do follow the assignment policy
  - If rebalancing is off, ingest follows the assignment policy to determine where to place documents.
    - They just won't move around after that as long as rebalancing is off.

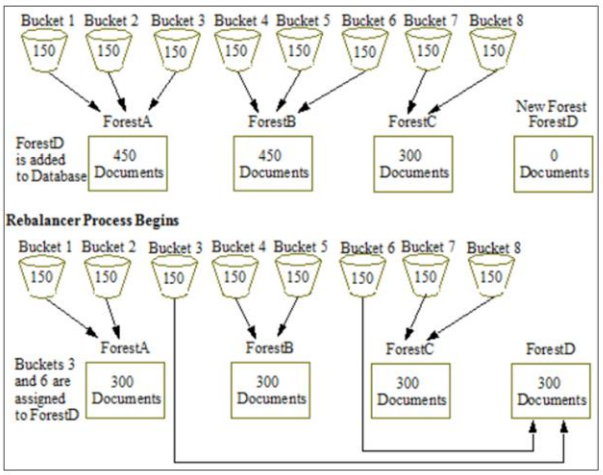
MarkLogic UNIVERSITY

Assignment Policies

- Bucket
- Statistical
- Range
- Legacy

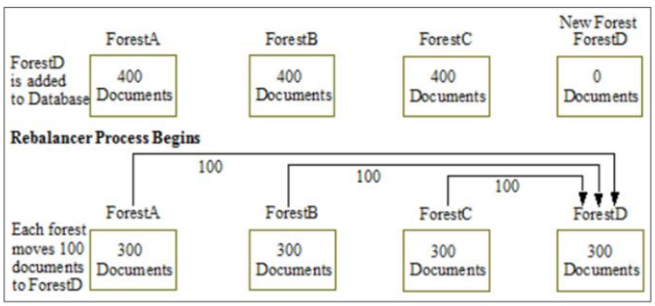
Assignment Policies: Bucket

- Generally most efficient, least data movement
- URI→Bucket→Forest
  - Bucket:Forest mapping is kept in memory for fast access
- Deterministic algorithm
  - An input will always produce same output, sequence of states
- 16k buckets regardless of number of forests



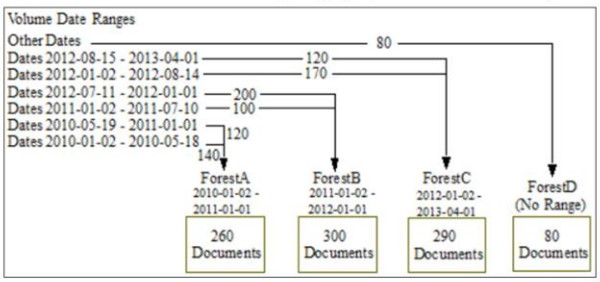
Assignment Policies: Statistical

- Each forest keeps track of the number of documents it contains
- Documents move from forests with most to forests with least
  - Moves the least number of documents to achieve balance
  - Balance does NOT require equal number of documents across all forests



Assignment Policies: Range

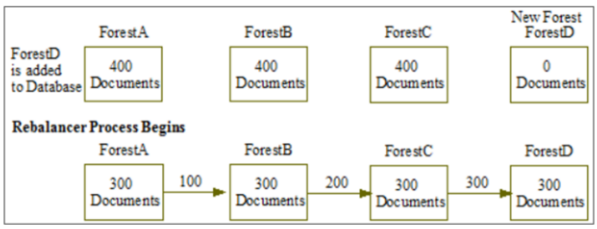
- For use with Tiered Storage
  - Partitions, Partition Key concepts
- Multiple forests can cover same range
  - Statistical assignment policy among multiple forests in a partition
- Partial range overlap is not valid
- Default partition to allow for range gaps, docs without partition key



MarkLogic UNIVERSITY

Assignment Policies: Legacy

- Databases that have been upgraded to ML7
  - Rebalancer is off
  - Assignment policy = legacy
  - No automatic rebalancing will occur
- If you turn the rebalancer on and leave the assignment policy as legacy, behavior will be as follows:
  - Least efficient, most data movement



Assignment Policies: Summary

Policy	Data Movement	Deterministic?	Backward Compatible?
Bucket	Less	Yes (URI based)	No
Statistical	Least	No	No
Range	Less	Yes (Partition key based)	No
Legacy	Most	Yes (URI based)	Yes

Slide 11

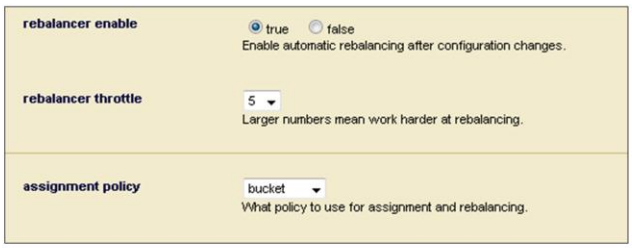
Rebalancing: How it Works

- Similar to reindexing
- Configured at database level, runs at forest level
- Rebalancer + Assignment Policy
  - Figures out which URIs don't belong in a forest and moves them where they do belong
- Each transaction moves a batch of documents
  - Delete from origin and insert in destination is in same transaction
  - All fragments with same URI are in same transaction
    - Only if fragmentation is on for your database
    - Default database configuration is 1 document = 1 fragment

MarkLogic UNIVERSITY

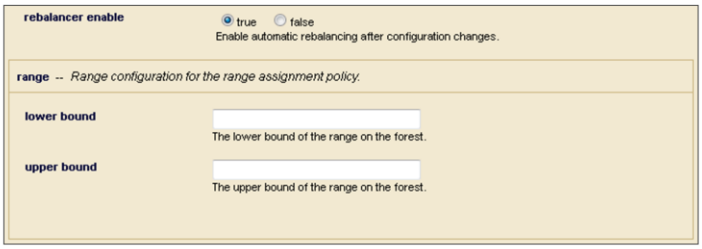
Database Level Configuration

- Throttle
  - Higher number = rebalancing is high priority
    - 5 says after a rebalancing transaction completes, start the next one
    - Uses more system resources
  - Lower number = lower priority
    - Time between transactions is longer as you move to lower numbers



Forest Level Configuration

- Database must have rebalancing on for a forest to participate
- Database rebalancing on, but certain forests turned off is valid

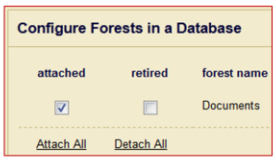


Retiring a Forest

- Retire
  - Rebalancer will move all docs in the retired forest to other forests
  - Retired forest not an eligible destination for new inserts
- Detach
  - After documents have been moved from retired forest
- Delete
  - Remove the forest from your configuration
- Can be accomplished via Admin interface, XQuery or REST APIs

```
curl --anyauth --user admin:admin -X POST -i \
-d "state=retire" -d "database=myDatabase" \
-H "Content-type:application/x-www-form-urlencoded" \
http://localhost:8002/manage/v2/forests/myForest
```

```
admin:database-retire-forest()
```



Unit 14: Applying the Learning Objectives

- Describe how the rebalancer operates.
- Describe bucket, statistical, range and legacy assignment policies.
- Configure the rebalancer.
- Resize a partition.
  - Exercise 1
- Retire a forest.
  - Exercise 2
- Take a forest or partition online / offline
  - Exercise 3

Slide 16