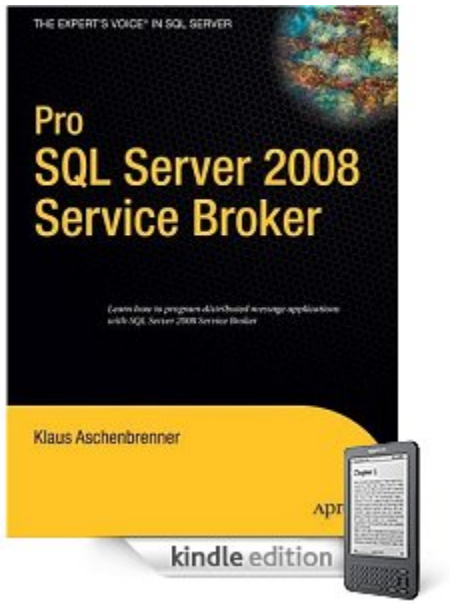
UNIQUEIDENTIFIERs as PRIMARY KEY Values

Klaus Aschenbrenner
Microsoft Certified Master SQL Server 2008
www.SQLpassion.at
Twitter: @Aschenbrenner

About me

- CEO & Founder SQLpassion
- International Speaker, Blogger, Author
- SQL Server 2008 MCM
- "Pro SQL Server 2008 Service Broker"
- Twitter: @Aschenbrenner
- SQLpassion Academy
  - http://www.SQLpassion.at/academy
  - Free Newsletter, Training Videos



Agenda

- Clustered Indexes
- Non-Clustered Indexes
- GUIDs as Primary Keys
- Scalability

Agenda

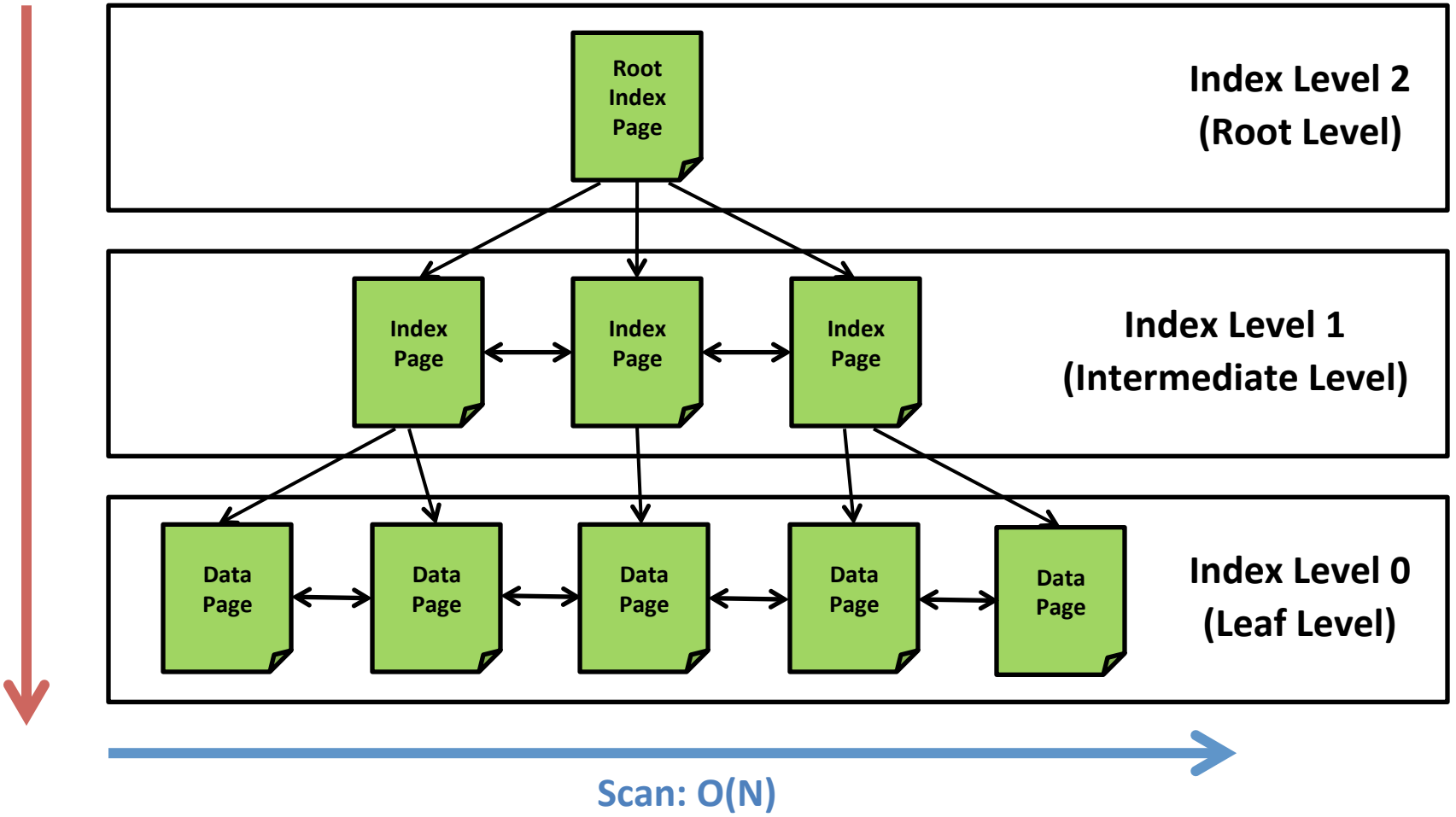
- Clustered Indexes
- Non-Clustered Indexes
- GUIDs as Primary Keys
- Scalability

Clustered Index

- Structured as a (B)alanced-Tree
- The index is the data
- The data is ordered through the Clustered Key
- Analogy: Word-Dictionary

Clustered Indexes

Seek: O(log N)



Demo

Clustered Indexes

Agenda

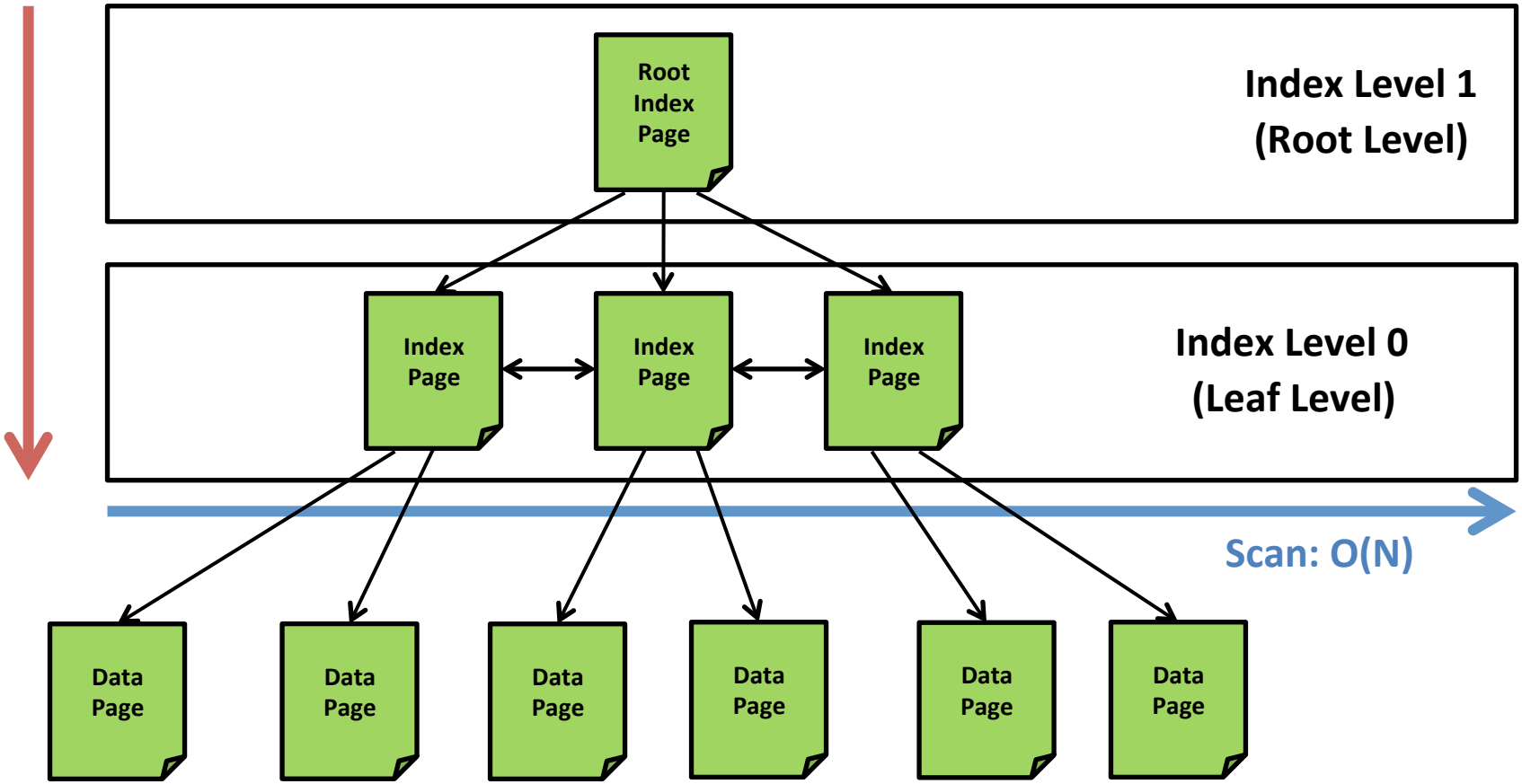
- Clustered Indexes
- Non-Clustered Indexes
- GUIDs as Primary Keys
- Scalability

Non-Clustered Index

- Also structured as a (B)alanced-Tree
- The index rows on the lowest level points to the record
- Data Pages are **NOT** part of the Non-Clustered Index
- Analogy: Index in a book

Non-Clustered Indexes

Seek: O(log N)



Clustered Key

- Clustered Key is replicated in each Non-Clustered Index
  - Must be chosen very carefully...
  - ... because it impacts **EACH** Non-Clustered Index!
- All Clustered Indexes must be unique
  - Non-Clustered Index **MUST** point to one record
  - Otherwise a UNIQUIFIER is added by SQL Server (4 byte Integer overhead)
  - UNIQUIFIER becomes part of the Clustered Key

Clustered Key Dependency

- Clustered Key is most redundant data in database
- If Clustered Key changes...
  - ... every Non-Clustered Index **MUST** be changed
- Optimal Clustered Key
  - Unique
  - Narrow
  - Static
  - Ever increasing
- Later change is sometime very aweful
  - Primary Key constraints!

Demo

Non-Clustered Indexes

Agenda

- Clustered Indexes
- Non-Clustered Indexes
- GUIDs as Primary Keys
- Scalability

Primary Keys vs. Clustered Keys

- Primary Key <> Clustered Key
- Enforces uniqueness in a column
- Uniqueness is enforced by
  - Unique Clustered Index (default)
  - Unique Non-Clustered Index

GUIDs as Primary Keys

- Disadvantages
  - Not sequential -> leads to high fragmentation & Page Splits
  - Storage Overhead (16 bytes long)
  - Lookup value for Non-Clustered Indexes
- Are you really doing range scans by GUIDs?

Size does matter!

- Scenario: Table with 1.000.000 records, 6 NCs
- INT
  - 3,8 MB for CI (1.000.000 x 4 / 1024 / 1024)
  - 22,89 MB for NC (6 x 1.000.000 x 4 / 1024 / 1024)
  - Sum: ~ 27 MB
- GUID
  - 15,26 MB for CI (1.000.000 x 16 / 1024 / 1024)
  - 91,55 MB for NC (6 x 1.000.000 x 16 / 1024 / 1024)
  - Sum: ~ 107 MB
- 400% STORAGE OVERHEAD!!!

Performance does matter!

- Scenario: 100.000 records INSERT
- INT as Primary Key
  - SSD: ~ 30sec
  - 7200rpm: ~ 1:00min
- UNIQUEIDENTIFIER as Primary Key
  - SSD: ~ 40sec
  - 7200rpm: ~ 1:50min

Demo

GUIDs as Primary Keys

Resolution

- FILLFACTOR
  - Leaves empty space on the data pages
  - Decreases page splits and random I/O
- Enforce the Primary Key through a Non-Clustered Index
  - 1. Take database offline
  - 2. Disable all FKs
  - 3. Disable all Non-Clustered Indexes
  - 4. Drop the Clustered Index (the Primary Key)
  - 5. Add new Clustered Key column (INT IDENTITY?)
  - 6. Create the new Clustered Index
  - 7. Create the Primary Key as NONCLUSTERED
  - 8. Enable FKs and Non-Clustered Indexes
  - 9. Bring database online

Demo

Resolution

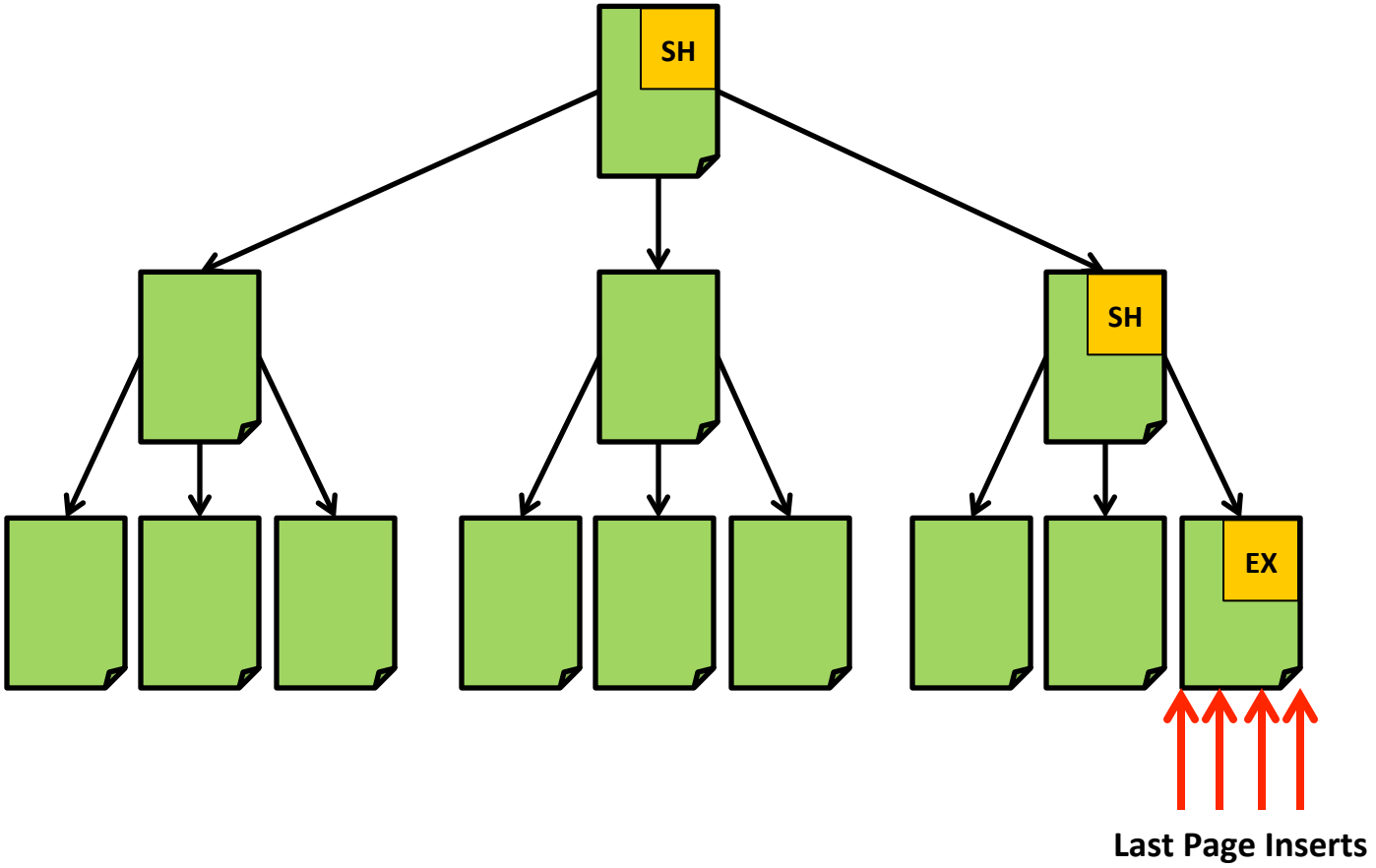
Agenda

- Clustered Indexes
- Non-Clustered Indexes
- GUIDs as Primary Keys
- Scalability

Scalability

- An ever increasing value doesn't scale!
  - INT IDENTITY
  - Last Page Insert Latch Contention

Last Page Insert Latch Contention



Current Solutions

- Random Clustered Keys
  - UNIQUEIDENTIFIER
  - Distributes the INSERTs across the Leaf Level
  - Larger Lookup Values in Non-Clustered Indexes…
- Hash Partitioning
  - Distribute INSERTs across different partitions
  - Every CPU core has its own partition
  - You can't additionally partition your table…
  - Partition Elimination is almost not possible…
- In-Memory OLTP
  - SQL Server 2014+

Demo

Last Page Insert Latch Contention

Summary

- Clustered Indexes
- Non-Clustered Indexes
- GUIDs as Primary Keys
- Scalability

SQL Server Performance Tuning Workshop

- June 1 – 5 in London
- Agenda
  - Database Internals
  - Execution Plans
  - Indexing
  - Statistics
  - Locking, Blocking, Deadlocking
  - Performance Troubleshooting
- More Information
  - http://www.SQLpassion.at/academy/perftuning
  - 10% Discount!