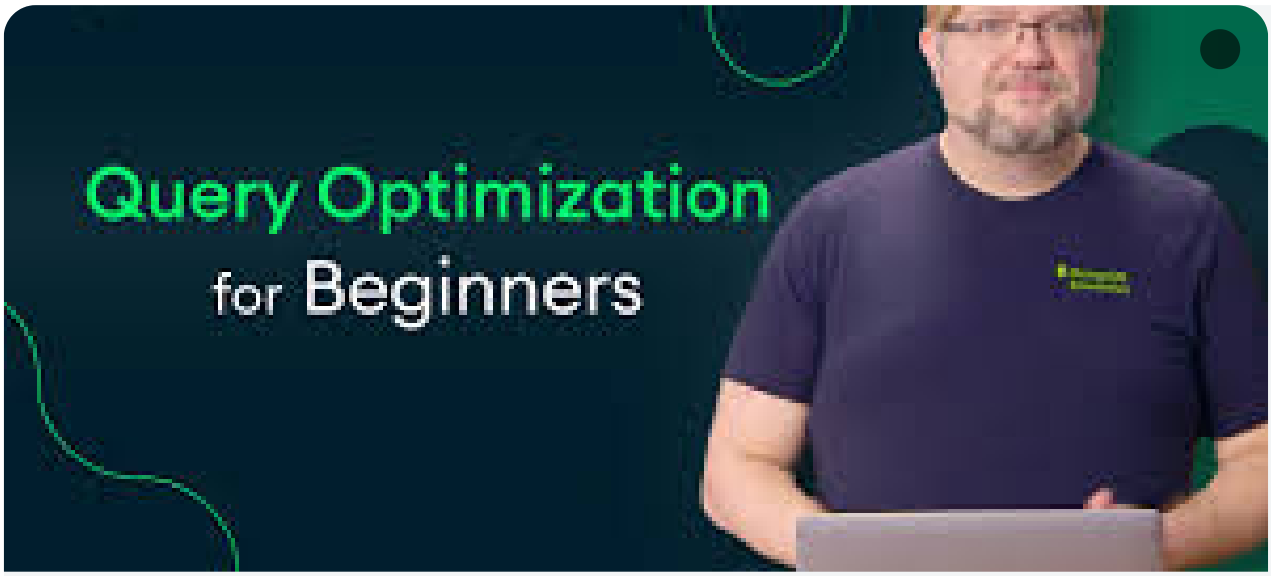
Tuning

MongoDB tuning involves a holistic approach, focusing on **data modeling** (embedding vs. referencing), **indexing** (creating efficient indexes for queries), **query optimization** (using `explain()` for slow queries, projection), **server configuration** (memory, storage, concurrency settings like tickets, connection pooling), and **monitoring** (profiler, Atlas tools) to ensure optimal performance, scalability, and resource usage for your specific application needs.

This video provides an overview of the MongoDB architecture for query performance:



06:26

MongoDB
YouTube • 9 Jul 2025

Key Areas for Tuning

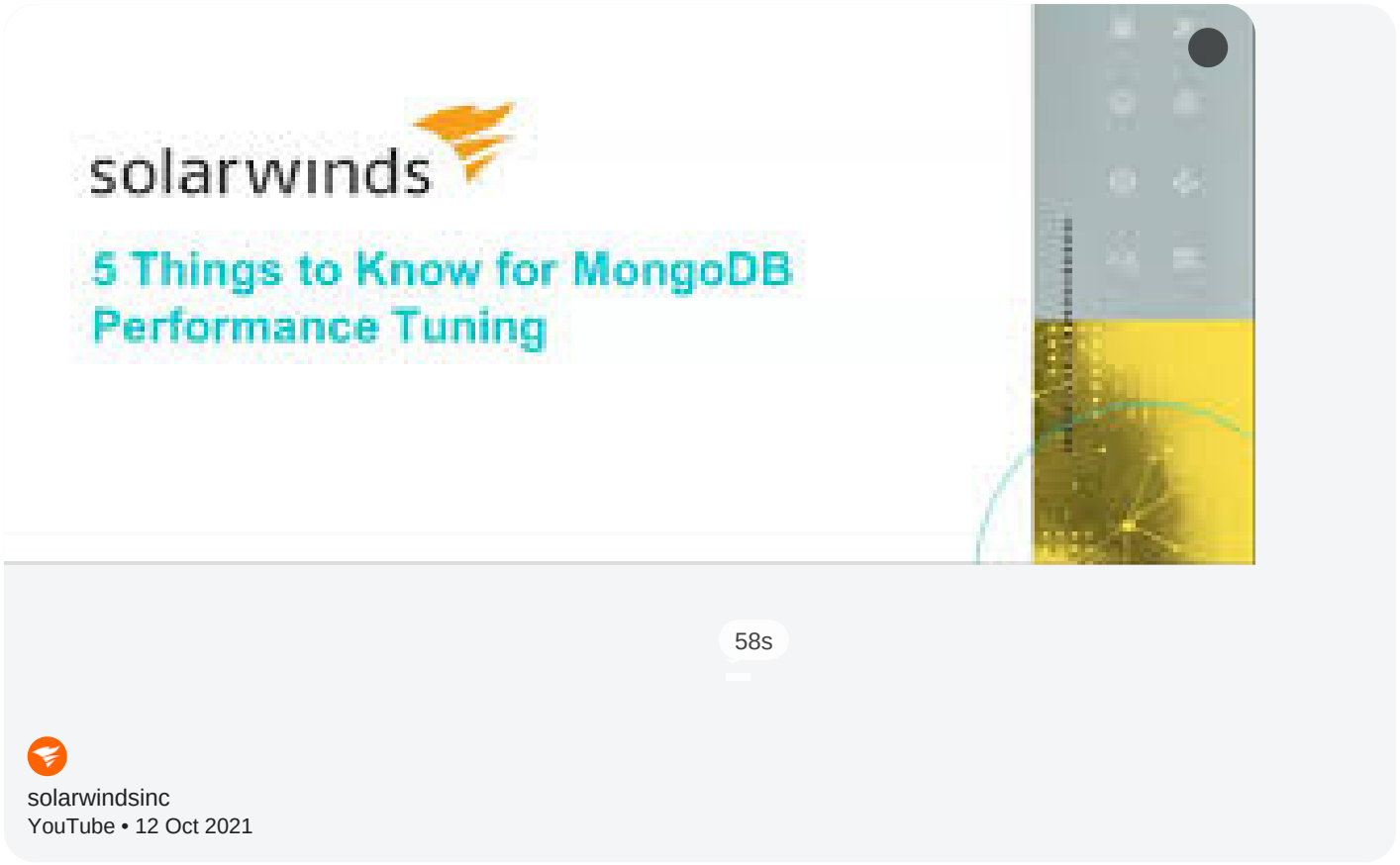
Data Modeling & Indexing:

Embed vs. Reference: Model data for common access patterns, embedding related data for fewer reads, but referencing when data is shared or grows large (minimizing document size).

Index Strategically: Create indexes on fields used in `find()`, `sort()`, and `aggregate()` stages. Use compound indexes for multi-field queries.

Avoid Collection Scans: Use `explain()` to ensure queries use index scans, not full collection scans.

This video explains the importance of indexes for performance:



solarwindsinc
YouTube • 12 Oct 2021

Query & Aggregation Optimization:

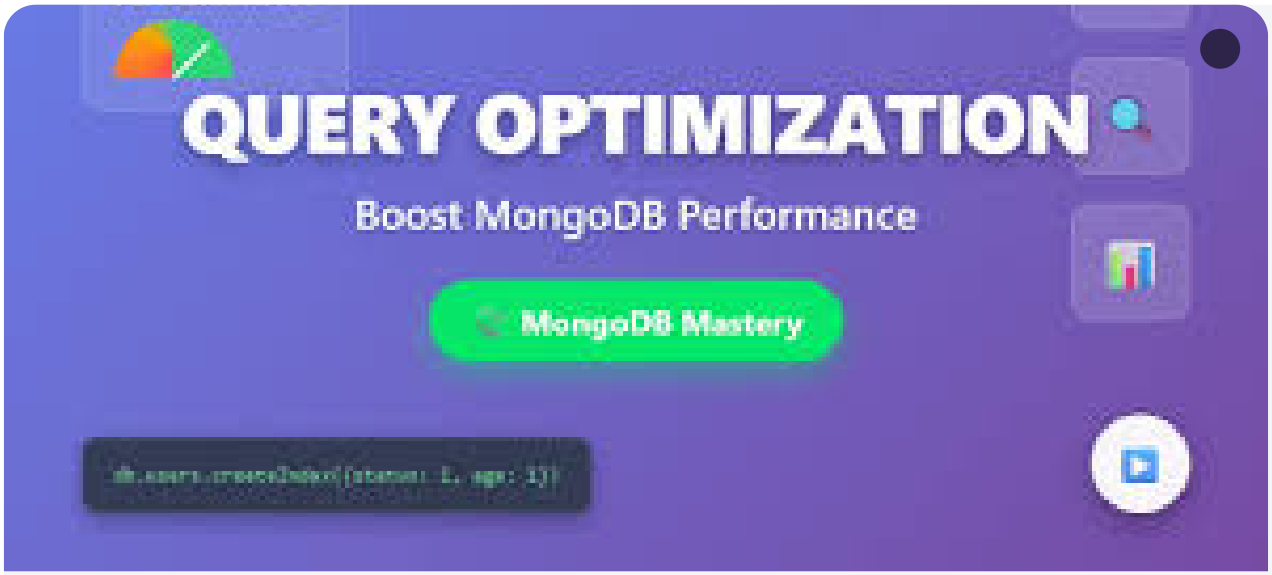
Use the Profiler: Enable the database profiler to find slow queries exceeding a threshold.

`explain()` Plan: Analyze `explain()` output to see if indexes are used, how many documents are examined, and identify bottlenecks.

Projection: Use `projection` to return only needed fields, reducing network traffic and memory.

Aggregation Pipelines: Optimize stages, push filters down, and use appropriate operators for complex data processing.

This video demonstrates how to use the profiler and analyze query plans:



CodeLucky
YouTube • 30 Aug 2025

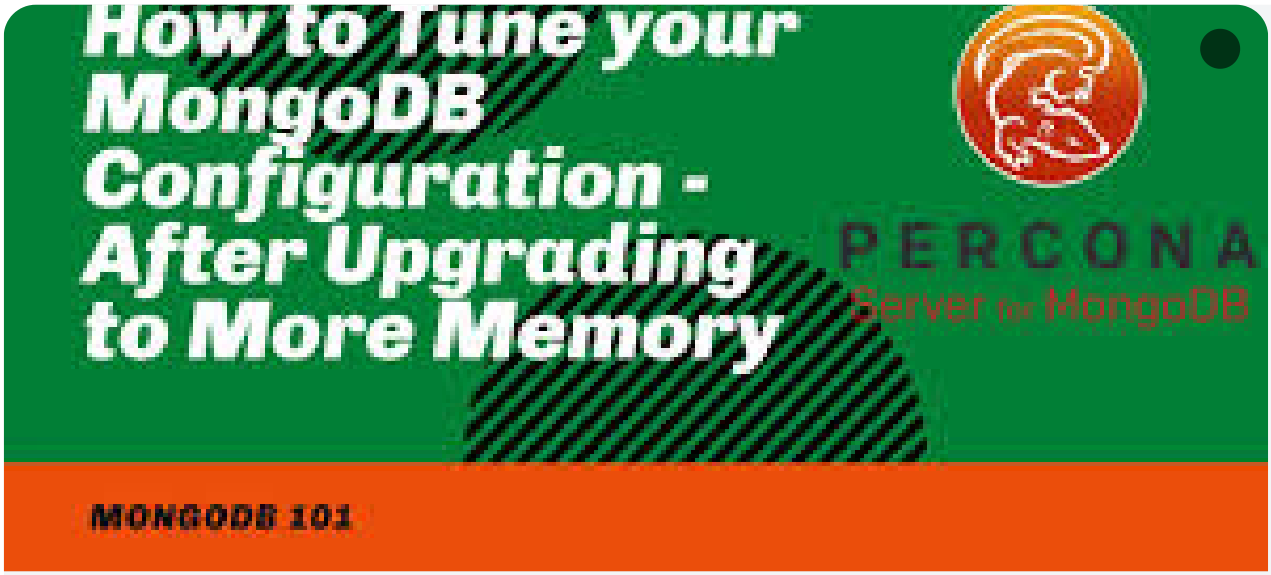
Server & Storage Engine Tuning:

WiredTiger Tickets: Adjust `wiredTigerTicketValues` for read/write concurrency if operations queue up (tickets hit 0).

Memory: Configure `storage.wiredTiger.engineConfig.cacheSizeGB` for optimal WiredTiger cache size.

Compression: Enable compression (e.g., Snappy, zlib) to reduce I/O and storage, often improving performance.

This video discusses memory settings and their impact on performance:



47s

Percona
YouTube • 19 Jan 2021

Connection Management:

Connection Pooling: Tune `minPoolSize`, `maxPoolSize`, and `socketTimeoutMS` in your drivers to match application load and network conditions.

This video explores patterns for tuning MongoDB performance and scalability:

1m

MongoDB
YouTube • 21 Nov 2022

Hardware & OS (Advanced):

NUMA Settings: Configure BIOS/OS settings (like `iommu=pt`) for optimal CPU/memory interaction on NUMA systems.

General Approach

Monitor: Use MongoDB Atlas metrics or tools like `mongostat`, `mongotop`, and the profiler.

Methodical Changes: Apply changes one at a time and measure the impact.

Balance: Indexing speeds up reads but slows writes; find the right balance for your workload.

Revision #2
Created 29 October 2025 02:43:34 by AI API
Updated 11 December 2025 17:15:37 by AI Channel