

SQL Server Performance Report

Server Name: QATAR

Report Date: September 10, 2024

Executive Summary

This report provides an analysis of the performance of QATAR SQL Server. The focus is on identifying key performance metrics, potential issues, and recommendations for improving overall database performance.

Current Performance Metrics

CPU and Memory Usage

- **Average CPU Usage:** 62%
- **Average CPU Usage during High-demand Intervals:** 92%
- **Peak CPU Usage:** 100%
- **Total Memory:** 128 GB
- **Average Memory Usage:** 120 GB (94%)
- **Peak Memory Usage:** 121.6 GB (95%)
- **SQL Service Memory Limit:** 98.69 GB
- **Average SQL Service Memory Usage:** 104.87 GB (106%)



Disk I/O (see full data on attached excel report)

- **Average Disk Read Latency:** 2.21 ms
- **Average Disk Write Latency:** 13.13 ms
- **Peak Disk Read Latency:** 5.58 ms
- **Peak Disk Write Latency:** 42.27 ms

Database and Log File Size

- **Total Database Size:** 780.70 GB
- **Total Log files Size:** 76.58 GB

Query Performance last 15 days

- **Total Queries Executed:** 10,367
- **Average Query Response Time:** ~1.97 seconds
- **Slowest Query Response Time:** ~4,349.30 seconds

Potential Issues

1. High Peak CPU Usage:

- The server experiences occasional spikes in CPU usage, reaching up to 100% during peak times. While this is expected under heavy load, it still points to potential CPU bottlenecks during these high-demand periods.

2. Memory Utilization:

- Memory usage is consistently at 94%, peaking at 95%, this leaves about 6 GB (4.7%) of RAM free. While this is not a lot of free memory, it should be sufficient for the operating system and other system processes.
- The SQL service is consistently utilizing approximately 106% of its allocated memory limit, meaning that additional memory usage is occurring beyond the set boundaries. This extra usage could be attributed to allocations outside the buffer pool, including Query Workspace Memory, Extended Stored Procedures, Linked Server Queries, and Thread Stack Memory. After a thorough examination of all SQL components' RAM usage, we found a total of 94,408 MB, which differs from the 104,984 MB reported in task manager. A server restart might help to reconcile this discrepancy.

3. Disk Latency:

- Average read is good and write latency is acceptable. Peak Disk Write Latency is relatively high (42.27 ms) and could be a point of concern. Persistent peaks in write latency above 20 ms might indicate an issue that could affect SQL Server performance, particularly for write-heavy operations

4. Slow Queries:

- Multiple queries are experiencing slow response times, with the slowest taking up to 4,349.30 seconds. This can impact user experience and application performance.

5. **Database Index:**

- We identified around 60 missing indexes that could potentially benefit the server performance.
- We identified around 40 current indexes in place that are not considered as beneficial for performance, since they have a high number of updates but a low number of seeks, scans, or lookups.

6. **Database Log File Size:**

- The recommended size for a database log file is typically 25% to 50% of the database's total size. However, the log files for the "Forecast," "Steve," and "XREF" databases are exceeding these recommended limits. This may negatively impact I/O performance, increase backup and restore times, and contribute to overall performance degradation.

Recommendations

1. **CPU Optimization:**

- Review and optimize queries and stored procedures that are consuming high CPU resources (Attached excel file has the expensive queries).
- Consider upgrading the CPU if high usage persists after optimization efforts. Increasing CPU cores and logical processors from 4 to 16.

2. **RAM Optimization:**

- The SQL Server process is exceeding its configured RAM limit, which can negatively impact both SQL Server performance and the overall system.
- Investigate Non-Buffer Pool Memory Consumption
- Identify queries that are using excessive memory and optimize them by adjusting indexing, query plans, or batch processing methods to reduce memory pressure
- Pay particular attention to memory allocations outside the buffer pool (e.g., query workspace memory, thread stack memory, and linked server queries)
- Monitor and Fine-Tune
- Consider increasing the maximum memory allocated to SQL Server
- Consider Upgrading Hardware.

3. **Disk Performance:**

- Investigate the storage subsystem for potential issues or bottlenecks.
- Ensure that TempDB is optimized, as it can be a major contributor to write latency.
- Regularly back up the transaction log to prevent it from growing excessively. Ensure that the log file is placed on a separate disk from the data files to reduce contention.
- Use RAID configurations optimized for performance, such as RAID 1 (mirroring) or RAID 10 (striping and mirroring).

4. General Server:

- Implement regular performance monitoring and reporting to proactively identify and address potential issues.
- Consider scheduling a monthly or quarterly reboot. On 09/10/24 we noticed that the server has been up for 119 days.
- Avg. internet down/up speed 400/220Mbps, which is OK.

5. Query Optimization:

- We identified around 60 indexes that can be added to your databases and could improve the performance.
- Optimize the slowest queries by analyzing execution plans and making necessary indexing improvements.
- Select necessary columns only.
- Consider using LIMIT or TOP clause

6. Adjust Log File Size:

- Auto-Growth Settings
- Shrink the Log File if possible.

Conclusion

The current performance of the QATAR database highlights opportunities for enhancing overall performance and reliability. Optimizing table indexing, SQL Queries improvement, fine-tuning SQL RAM configuration, and reducing disk write latency will have a significant positive impact. By addressing these issues and implementing the recommended actions, we can improve resource utilization and boost query performance.