

Whitepaper

Produkt: combit Relationship Manager / address manager

Empfehlungen zur Konfiguration einer Microsoft SQL Server-Instanz

Inhalt

Einleitung	3
Instanzspezifische Konfigurationsmöglichkeiten	3
Begrenzung des maximal genutzten Arbeitsspeichers	3
Begrenzung der Nutzung von logischen Prozessoren für einzelne SQL-Abfragen	4
Datenbankspezifische Konfigurationsmöglichkeiten	7
Auslagerung der Datenbankdatei(en) für das Transaktionsprotokoll	7
Konfiguration der Anfangsgröße und Vergrößerung von Datenbankdateien	8
Konfiguration der Anfangsgröße für die zentrale Datenbankdatei (*.mdf)	8
Konfiguration der Anfangsgröße für die Datenbankdatei des Transaktionsprotokolls (*.ldf)	8
Konfiguration der automatischen Vergrößerung für Datenbankdateien	9

Einleitung

Dieses Whitepaper erläutert die empfohlene Konfiguration einer Microsoft SQL Server-Instanz, um Ressourcen zu schonen und die Performance zu erhöhen.

Instanzspezifische Konfigurationsmöglichkeiten

Begrenzung des maximal genutzten Arbeitsspeichers

Die SQL Server-Instanz sollte nicht den kompletten zur Verfügung stehenden Arbeitsspeicher für sich beanspruchen. Wir empfehlen daher dem Betriebssystem mindestens 2 GB Arbeitsspeicher bzw. 10% des verfügbaren Hauptspeichers zu überlassen, je nachdem welcher Wert zuerst erreicht wird.

Eine Beispielrechnung sieht wie folgt aus:

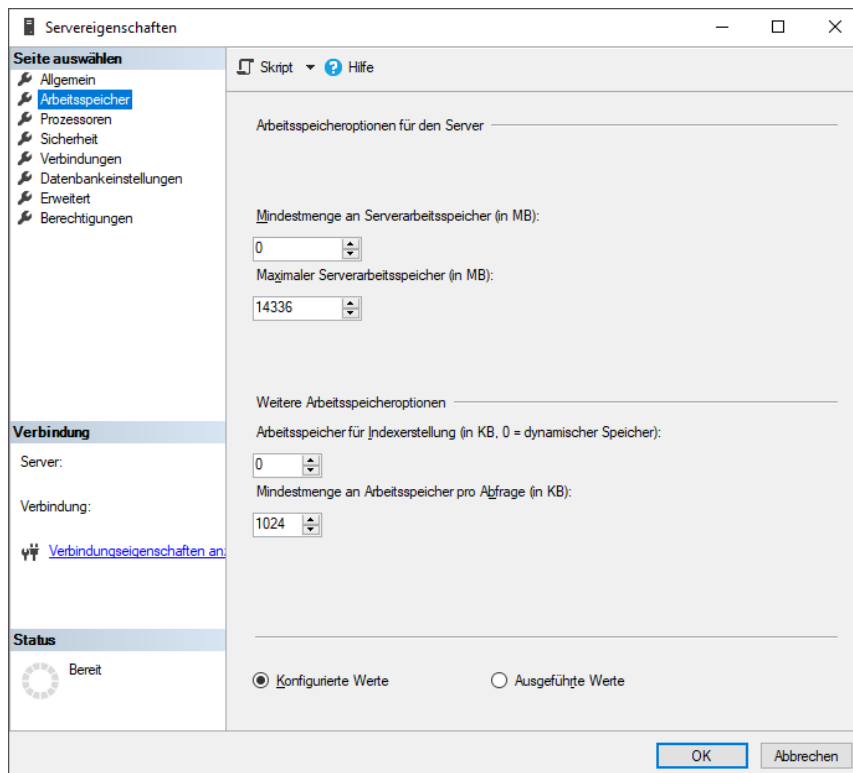
Verfügbare Arbeitsspeicher: 16 GB

10% des Arbeitsspeichers entsprechen: 1,6 GB

Ergebnis: Es sollten maximal 14 GB des zur Verfügung stehenden Arbeitsspeichers für die SQL Server-Instanz freigegeben werden.

Die Konfiguration dieser Eigenschaft ist wie folgt möglich:

1. Verbinden Sie sich mit Hilfe des SQL Server Management Studios mit Ihrer SQL Server-Instanz.
2. Öffnen Sie die Servereigenschaften, indem Sie einen Rechtsklick auf den Instanznamen innerhalb des Objekt-Explorers durchführen und den Eintrag "Eigenschaften" wählen.
3. Wechseln Sie auf die Seite "Arbeitsspeicher" und tragen Sie den gewünschten Wert in der Eigenschaft "Maximaler Serverarbeitsspeicher (in MB)" ein.



Begrenzung der Nutzung von logischen Prozessoren für einzelne SQL-Abfragen

Um zu verhindern, dass eine komplexe SQL-Abfrage alle verfügbaren logischen Prozessoren für sich beansprucht und so die Ausführung weiterer SQL-Abfragen blockiert, ist es möglich den maximalen Grad an Parallelität bzw. die Anzahl der logischen Prozessoren für die Ausführung einzelner SQL-Abfragen zu begrenzen.

Ein empfohlener Wert ist davon abhängig wie viele logische Prozessoren zur Verfügung stehen und ob die SQL Server-Instanz mehrheitlich durch schnell abzuarbeitende oder komplexe SQL-Abfragen genutzt wird.

Üblicherweise werden vom combit Relationship Manager und address manager schnell abzuarbeitende SQL-Abfragen verwendet. Bei der Arbeit mit großen Datenmengen oder komplexen Filterausdrücken kann die Ausführungszeit pro Abfrage jedoch einen kritischen Wert erreichen, sodass unter Umständen parallel auszuführende SQL-Abfragen blockiert werden.

In der nachfolgenden Tabelle finden Sie entsprechende Empfehlungen zur Konfiguration:

Anzahl logischer Prozessoren	Nutzungsart	Empfohlener Wert
Bis zu 8 logische Prozessoren	Mehrheitlich schnell abzuarbeitende SQL-Abfragen	Halten Sie den Wert auf oder unter der Anzahl der logischen Prozessoren
Bis zu 8 logische Prozessoren	Mischung aus schnell abzuarbeitenden und komplexen SQL-Abfragen	Halten Sie den Wert unter der Anzahl der logischen Prozessoren
Bis zu 8 logische Prozessoren	Mehrheitlich komplexe SQL-Abfragen	Halten Sie den Wert maximal bei der Hälfte der logischen Prozessoren
Mehr als 8 logische Prozessoren	Mehrheitlich schnell abzuarbeitende SQL-Abfragen	Halten Sie den Wert bei genau der Hälfte der Anzahl der logischen Prozessoren
Mehr als 8 logische Prozessoren	Mischung aus schnell abzuarbeitenden und komplexen SQL-Abfragen	Halten Sie den Wert bei genau der Hälfte der Anzahl der logischen Prozessoren
Mehr als 8 logische Prozessoren	Mehrheitlich komplexe SQL-Abfragen	Halten Sie den Wert bei genau einem Viertel der Anzahl der logischen Prozessoren

Eine Beispielrechnung sieht wie folgt aus:

Verfügbare logische Prozessoren: 8

Nutzungsart: Mischung aus schnell abzuarbeitenden und komplexen SQL Abfragen

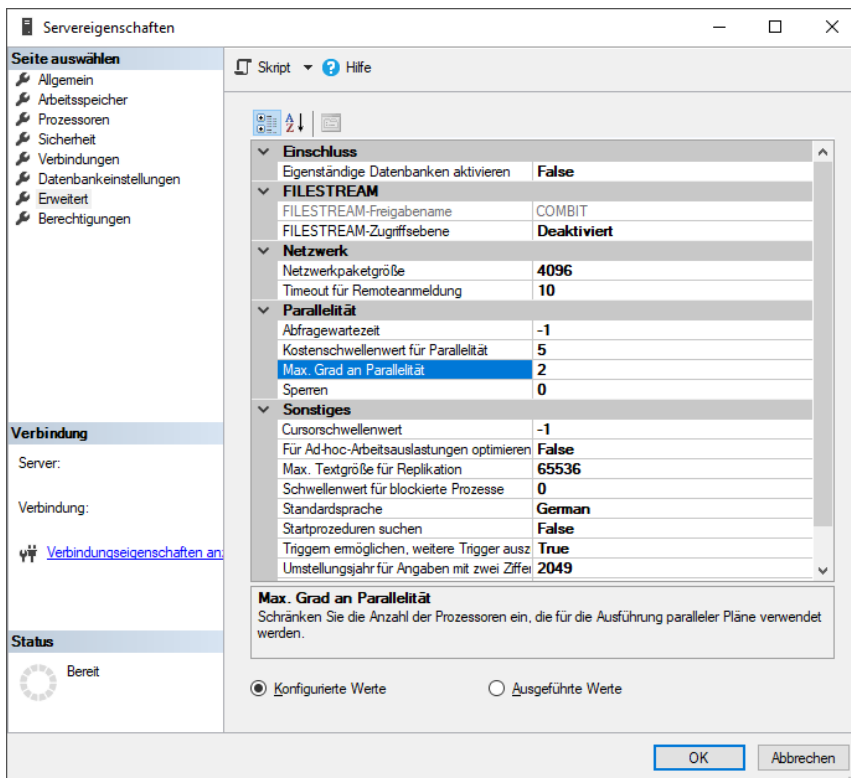
Empfohlener Wert: 2 bis 6

Nach Änderung der Eigenschaft sollte in jedem Fall die Performance überprüft bzw. mit der vorherigen Einstellung verglichen werden.

Die Konfiguration der Eigenschaft ist wie folgt möglich:

1. Verbinden Sie sich mit Hilfe des SQL Server Management Studios mit Ihrer SQL Server-Instanz.
2. Öffnen Sie die Servereigenschaften, indem Sie einen Rechtsklick auf den Instanznamen innerhalb des Objekt-Explorers durchführen und den Eintrag "Eigenschaften" wählen.

3. Wechseln Sie auf die Seite "Erweitert" und tragen Sie den gewünschten Wert in der Eigenschaft "Max. Grad an Parallelität" ein.



Datenbankspezifische Konfigurationsmöglichkeiten

Auslagerung der Datenbankdatei(en) für das Transaktionsprotokoll

Bei Schreibzugriffen in der Datenbank werden mindestens zwei Dateien beschrieben. Neben dem Schreiben bzw. Ändern von Informationen in die zentrale Datenbankdatei wird auch in eine zweite Datei für das Transaktionsprotokoll geschrieben.

Der Schreibzugriff für die zentrale Datenbankdatei erfordert üblicherweise, dass der Schreibkopf der Festplatte in unterschiedlichen Sektoren schreibt und dafür immer wieder neu positioniert werden muss.

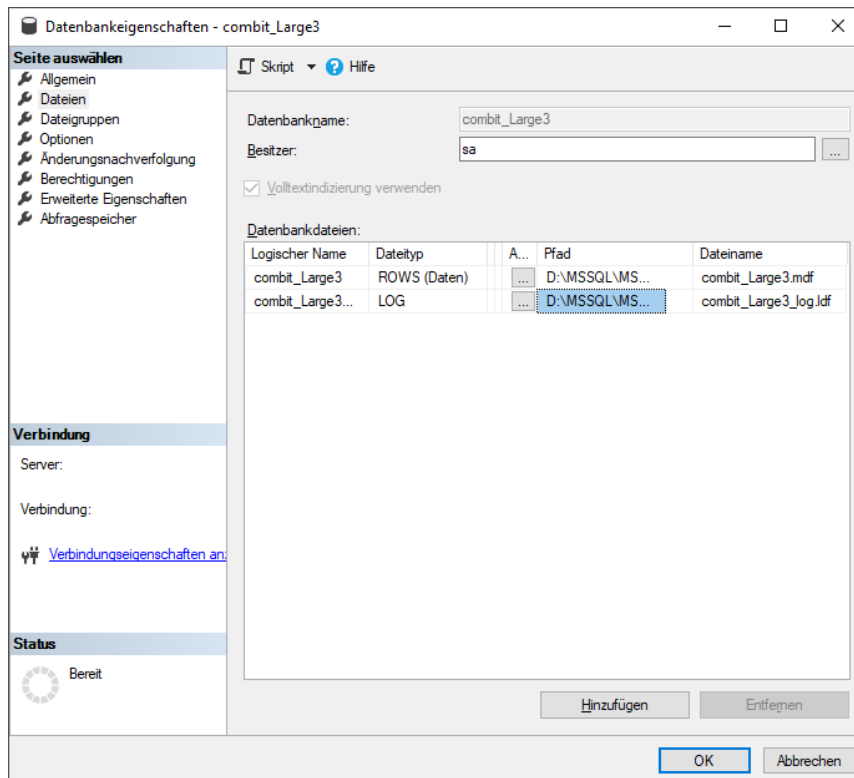
Das Transaktionsprotokoll wird hingegen sequenziell beschrieben, was den Schreibkopf nicht dazu veranlasst sich neu zu positionieren.

Wenn die Datenbankdatei für das Transaktionsprotokoll auf eine andere Festplatte bzw. ein anderes Festplattensystem ausgelagert wird, so können zusätzliche Neupositionierungen des Schreibkopfs minimiert werden, was Zugriffszeiten und Schreibraten positiv beeinflusst.

Bitte gehen Sie zur Konfiguration wie folgt vor:

1. Verbinden Sie sich mit Hilfe des SQL Server Management Studios mit Ihrer SQL Server-Instanz.
2. Erweitern Sie den Menüpunkt "Datenbanken" innerhalb des Objekt-Explorers und öffnen Sie die Eigenschaften der gewünschten Datenbank über einen Rechtsklick und der Auswahl des Menüpunktes "Eigenschaften".
3. Wechseln Sie zur Seite "Dateien" und notieren Sie sich den/die Speicherort(e) und den/die Dateiname(n) für das Transaktionsprotokoll. Sie erkennen die zugehörigen Datenbankdateien anhand der Spalte "Dateityp" und dem Wert "LOG".
4. Schließen Sie den Dialog "Datenbankeigenschaften" und schalten Sie die Datenbank offline. Dies können Sie mit Hilfe eines Rechtsklicks auf die Datenbank und der Auswahl des folgenden Menüpunktes erreichen: "Tasks" > "Offline schalten".
5. Sobald die Datenbank den Status "Offline" erhalten hat, können Sie den Windows Explorer öffnen und den/die unter Punkt 3 notierten Speicherort(e) aufrufen.
6. Verschieben Sie anschließend die Datenbankdatei(en) für das Transaktionsprotokoll an den neuen Speicherort.
7. Wechseln Sie wieder zurück in das SQL Server Management Studio und wählen Sie innerhalb der Eigenschaften und der Seite "Dateien" den neuen Pfad für die Datenbankdatei(en) für das Transaktionsprotokoll aus.
8. Zuletzt wird die Datenbank wieder online geschaltet. Wählen Sie die Datenbank innerhalb des Objekt-Explorers erneut mit einem Rechtsklick aus und nutzen Sie folgenden Menüpunkt: "Tasks" > "Online schalten".
9. Um zu überprüfen, ob die Dateiänderung erfolgreich war, kann folgende SQL-Abfrage ausgeführt werden:

```
SELECT name, physical_name AS CurrentLocation, state_desc
FROM sys.master_files
WHERE database_id = DB_ID(N'<Datenbankname>');
```



Konfiguration der Anfangsgröße und Vergrößerung von Datenbankdateien

Um zu verhindern, dass wiederkehrende Schreibzugriffe notwendig sind, um eine Datenbankdatei zu vergrößern, ist es sinnvoll sowohl eine angemessene Anfangsgröße zu wählen, als auch eine zukunftsfähige Konfiguration der automatischen Vergrößerung vorzunehmen.

Konfiguration der Anfangsgröße für die zentrale Datenbankdatei (*.mdf)

Bei der zentralen Datenbankdatei sollte die Anfangsgröße kurzfristig das Potenzial für das Hinzufügen zusätzlicher Daten zur Datenbank berücksichtigen. Wenn die Anfangsgröße der Daten beispielsweise 500 MB beträgt, aber bekannt ist, dass in den nächsten Monaten zusätzliche 500 MB an Daten hinzugefügt werden, ist es angebracht, die Datenbankdatei von Anfang an mit einer Größe von 1 GB zu erstellen, statt sie mehrmals vergrößern zu müssen, bis sie diese Größe erreicht.

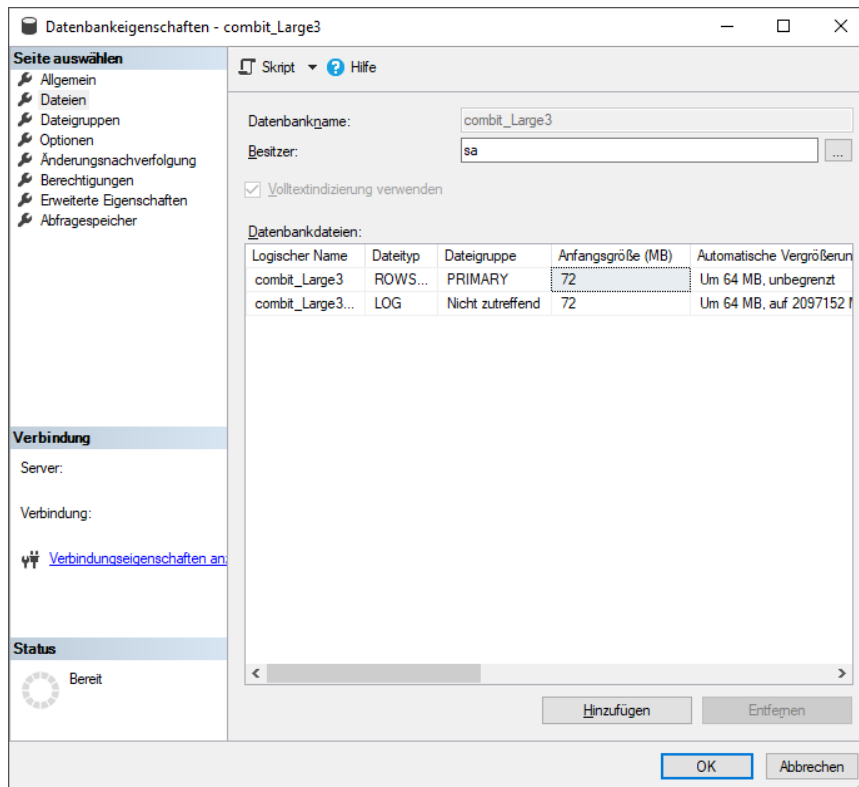
Konfiguration der Anfangsgröße für die Datenbankdatei des Transaktionsprotokolls (*.ldf)

Im Falle der Datenbankdatei für das Transaktionsprotokoll können keine pauschalen Empfehlungen ausgesprochen werden, da unter anderem folgende Faktoren berücksichtigt werden müssen: die Anzahl der Transaktionen, die vor einer Sicherung protokolliert werden und die Häufigkeit der Sicherung des Transaktionsprotokolls, welche die Größe der Datenbankdatei des Transaktionsprotokolls erheblich verkleinern kann, da üblicherweise ein Großteil der Protokollinformationen bei der Durchführung einer Sicherung des Transaktionsprotokolls abgeschnitten und für den laufenden Betrieb des Transaktionsprotokolls verworfen werden. Üblicherweise wird als Anfangsgröße für die Datenbankdatei des Transaktionsprotokolls die Anfangsgröße der zentralen Datenbankdatei verwendet.

Die Konfiguration für die Anfangsgröße kann wie folgt aufgerufen werden:

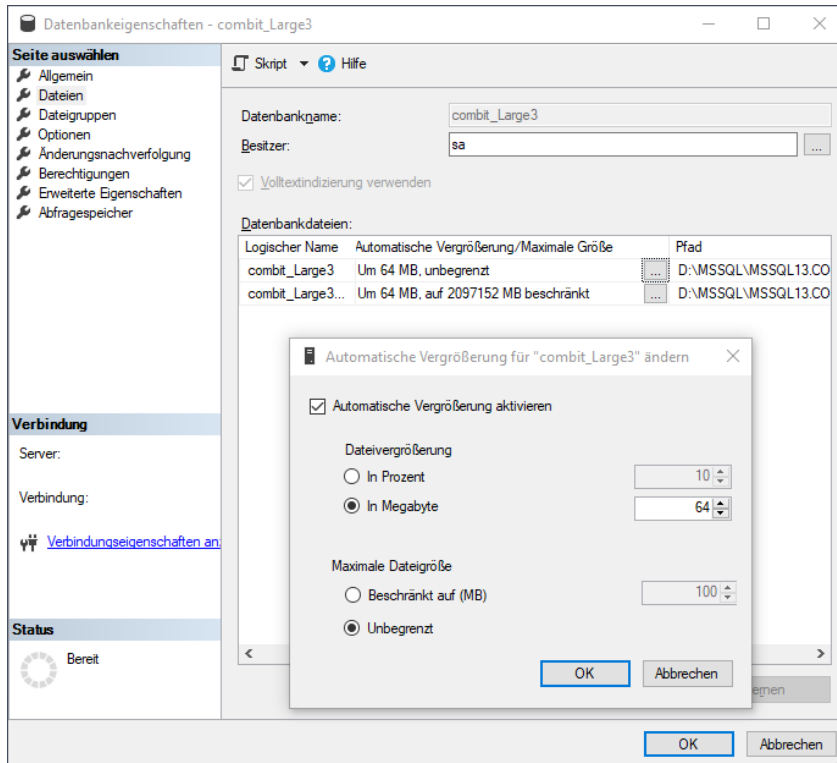
1. Verbinden Sie sich mit Hilfe des SQL Server Management Studios mit Ihrer SQL Server-Instanz.

2. Erweitern Sie den Menüpunkt "Datenbanken" innerhalb des Objekt-Explorers und öffnen Sie die Eigenschaften der gewünschten Datenbank über einen Rechtsklick und der Auswahl des Menüpunktes "Eigenschaften".
3. Wechseln Sie zur Seite "Dateien". Die Spalte "Anfangsgröße" lässt sich nun mit einem einfachen Klick in die Zelle der jeweiligen Datenbankdatei manuell anpassen.



Konfiguration der automatischen Vergrößerung für Datenbankdateien

Hinsichtlich der automatischen Vergrößerung sollte man zunächst die Option zur Dateivergrößerung in Megabyte wählen. Anschließend muss ein passender Wert eingetragen werden. Hier empfiehlt es sich einen Wert einzutragen, welcher nicht zu gering ist, sodass die automatische Vergrößerung regelmäßig zusätzliche Schreibzugriffe auf der Festplatte erzeugt, um die Datei zu erweitern. Außerdem sollte kein zu großer Wert gewählt werden, weil der Prozess der Vergrößerung so unnötig viel Zeit in Anspruch nehmen kann und Ressourcen bindet, die der SQL Server unter Umständen an anderer Stelle sinnvoller einsetzen könnte. Ein konkreter Wert könnte hierbei der Anfangsgröße der Datenbankdatei entsprechen.



Hinweis: combit macht keine Angaben zu einer bestimmten Eignung obiger Informationen. Irrtümer und Fehler bleiben ausdrücklich vorbehalten, die Angaben erfolgen ohne Gewähr und enthalten keine Zusicherung. Die Informationen können z.T. auch ein Versuch sein, Ihnen bei einer Aufgabenstellung zu helfen, selbst wenn das Produkt eigentlich nicht für diesen speziellen Zweck vorgesehen wurde.