

# Lösen Sie (fast) alle ihre Probleme mit Oracle Advanced Queuing

Matthias Schulz  
Schulz IT Services GmbH  
Nürnberg

## Schlüsselworte

Oracle Datenbank; Oracle Advanced Queuing; AQ; Messaging; IT-Probleme; Lösungen; Praxisbeispiele; Wartezeiten; Systemabhängigkeiten; Hochverfügbarkeit; Lastverteilung; Skalierbarkeit; Performance; Arbeitsteilung; Workflow; Automatisierung; Parallelisierung; Priorisierte Verarbeitung;

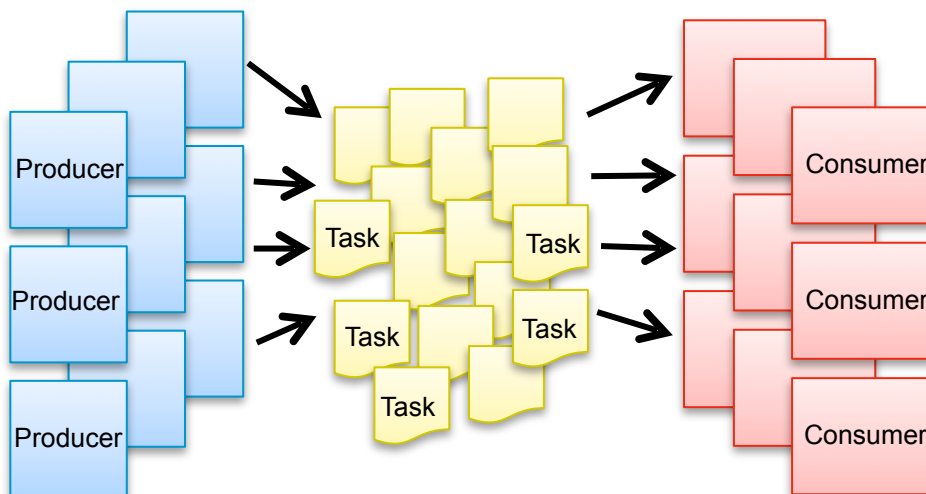
## Einleitung

Oracle Advanced Queuing ist ein in der Oracle Datenbank integriertes Messaging-System, mit dem sich eine Vielzahl häufiger Problemstellungen der IT einfach lösen lassen:

Wartezeiten  
**Hochverfügbarkeit** Wiedervorlage  
Systemabhängigkeiten **Workflow**  
Automatisierung  
Parallelisierung **Performance** Priorisierung  
Skalierbarkeit Lastverteilung

Ich werde Ihnen im Rahmen dieses Vortrags, bewährte Lösungen dieser Themen mit Oracle Advanced Queuing, anhand zahlreicher Praxisbeispiele näher bringen.

Darüber hinaus erhalten Sie einen Einblick in die Grundlagen des Messagings mit Oracle Advanced Queuing.



## IT Probleme und deren Lösung mit Oracle Advanced Queuing

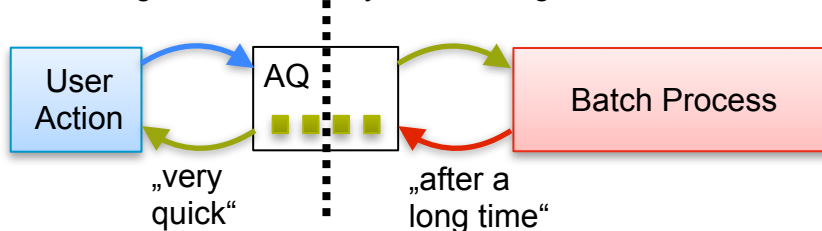
Die folgenden Beispiele entstammen realen Fällen:

### 1. Wartezeiten für Anwender

Die Anwender einer Web-basierten Anwendung mussten nach dem Absenden ihres Auftrags warten, bis der Server den Auftrag erledigt hatte und Rückmeldung gab. Dies führte zu Unzufriedenheit und gelegentlichen Doppelaufträgen durch die Anwender.

#### Die Lösung:

Der Auftrag wird nicht mehr synchron durchgeführt sondern in eine Message Queue eingestellt.



#### Die Vorteile:

- Die Anwender müssen nicht mehr warten, sondern erhalten sofort eine Rückmeldung.
- Durch die Verarbeitung von Auftrag und AQ-Nachricht in einer Transaktion, wird eine sichere und zeitnahe Auftragsbearbeitung gewährleistet.

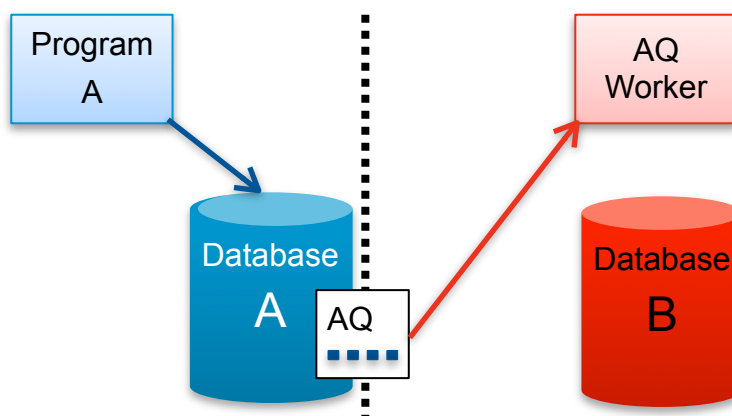
### 2. Systemabhängigkeiten und Hochverfügbarkeit

Viele der zentralen Unternehmensanwendungen waren direkt nur von einer Datenbank abhängig. Da jedoch auch Aufträge und Informationen an andere Systeme zuverlässig übermittelt werden mussten entstand ein komplexes Netz aus Abhängigkeiten.

Der Ausfall bereits einer Datenbank führte zu einem Ausfall aller Systeme.

#### Die Lösung:

Jede Anwendung stellt Aufträge und Informationen für andere Systeme in ihrer eigenen Datenbank als AQ-Nachrichten ein. Worker der Empfängersysteme holen diese Nachrichten ab und verarbeiten sie.



**Die Vorteile:**

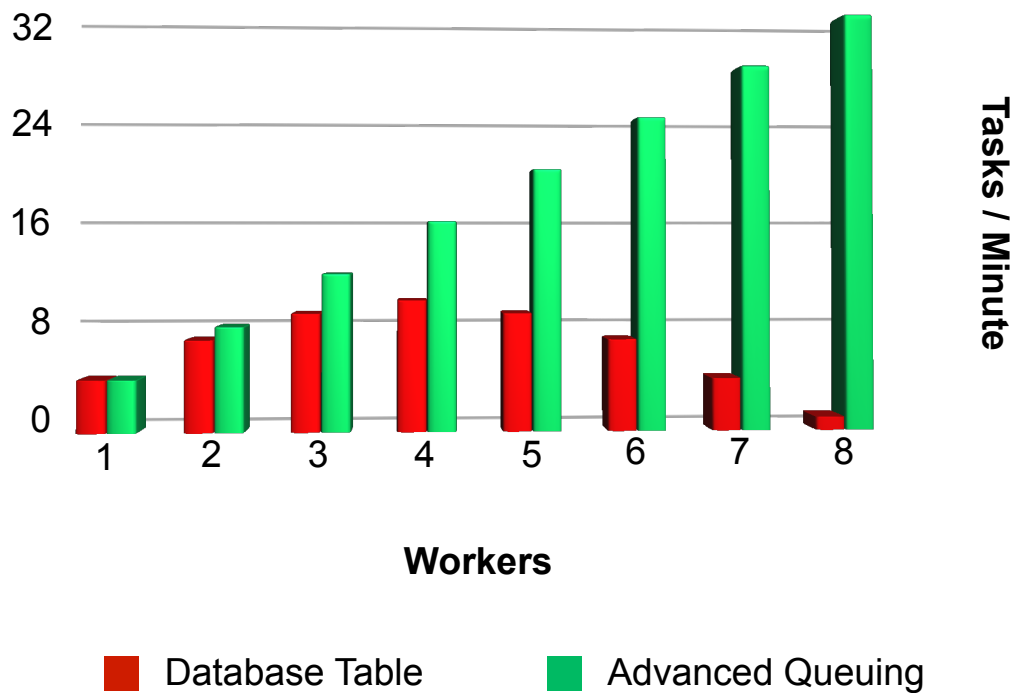
- Die Anwendungen sind nur noch von ihrer eigenen Datenbank abhängig.
- Ein Ausfall oder eine Downtime einer Datenbank betrifft nur noch die direkt abhängigen Anwendungen.
- Das Gesamtsystem ist weniger komplex und leichter zu konfigurieren.

3. *Skalierbarkeit und Performance*

Zur performanten Auftragsverarbeitung wurde ein System mit beliebig skalierbaren Workern eingesetzt. Es zeigte sich jedoch, dass der Durchsatz nicht linear mit der Anzahl der Worker stieg. Bereits bei 4 Workern wurde das Leistungsmaximum erreicht und mit 8 Workern fiel die Leistung nahezu auf Null zurück.

**Die Lösung:**

Die verwendeten Tabellen wurden durch eine Message Queue ersetzt.

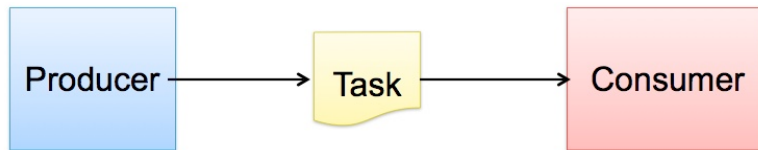


**Die Vorteile:**

- Der Durchsatz skaliert nun nahezu linear mit den eingesetzten Workern.
- Die Datenbanklast konnte deutlich gesenkt werden.

## Message Queue Typen

### 1. *Point to Point*

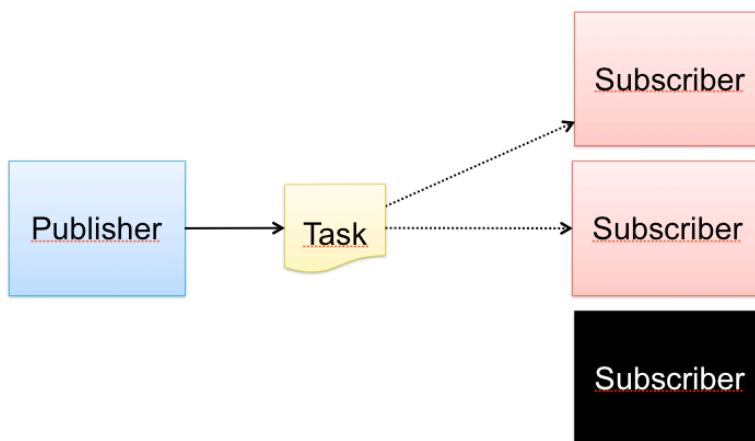


Eine Point-to-Point-Queue entspricht einem Brief, genauer einem Einschreiben mit Rückschein. Auch wenn jeden Tag Millionen Briefe geschrieben werden, so wird jeder einzelne Brief von einem Absender geschrieben und von einem Empfänger angenommen.

Der große Unterschied zum Brief ist, dass man den Erhalt wahlweise erst mit der Erledigung der Aufgabe bestätigen kann. Dies hat den Vorteil, dass beim Ausfall des Empfängers sofort ein anderer die Nachricht erhält.

Der Vorteil von Point-to-Point-Queues gegenüber Publish-Subscribe-Queues ist, dass der Empfänger nicht vor Versand der Nachricht registriert sein muss um diese zu erhalten.

### 2. *Publish / Subscribe - Multicast*



Multicast ist vergleichbar mit dem Abonnement einer Zeitung oder eines Magazins. Es wird an jeden registrierten Empfänger eine identische Kopie der Nachricht zugestellt.

Diese Methode eignet sich zur protokollierten sicheren Zustellung an einen großen Kreis von Empfängern.

Der Nachteil von Multicast ist, dass alle Empfänger vor dem Versand einer Nachricht registriert sein müssen.



### 3. Die Queue

Die Queue ist das Herzstück von Oracle Advanced Queuing und ermöglicht, API gekapselt, das Erstellen und Abrufen von Nachrichten.

Beispiel:

```
EXEC DBMS_AQADM.CREATE_QUEUE('my_message_queue',  
                              'my_queue_message_table');
```

### 4. Der Dispatcher

Der Dispatcher ist der entscheidende Vorteil von Oracle Advanced Queuing. Da er innerhalb der SQL-Engine läuft ist er in der Lage sowohl Locks zu verteilen, als auch zu überspringen. Hierdurch wird das Dilemma von Hotspots und Ausfallsicherheit wirksam aufgelöst. Der Dispatcher wird von der Datenbank verwaltet und automatisch gestartet.

### 5. Der Listener

Der Listener ist ein Programm innerhalb oder außerhalb der Datenbank, welches auf zu verarbeitende Nachrichten wartet.

#### **Kontaktadresse:**

Matthias Schulz  
Schulz IT Services  
Tauberstraße 28  
D-90449 Nürnberg

Telefon: +49 (0) 911-384 9090  
E-Mail: [info@schulz-it-services.de](mailto:info@schulz-it-services.de)  
Internet: [www.schulz-it-services.de](http://www.schulz-it-services.de)