



Last- und Performance-Monitoring von Business-Prozessen in der Datenbank

Kai Pillatzki, Stefan Triep und Andriy Terletsyy, Berenberg Bank

Um das Laufzeit-Verhalten von nachrichtengetriebenen Datenbank-Anwendungen sichtbar zu machen, werden detaillierte Informationen über das aktuelle Daten-Aufkommen sowie Verarbeitungs- und eventuell Wartezeiten verfügbar gemacht. Ein weiteres Ziel ist die Visualisierung dieser Daten und die Alarmierung im Falle einer Überschreitung bestimmter Schwellwerte. Dieser Artikel beschreibt das Last- und Performance-Monitoring von Business-Prozessen in der Datenbank mittels PL/SQL und Icinga.

Bei Berenberg wurde entschieden, dass die Business-Logik zur Datenverarbeitung in der Oracle-Datenbank liegen soll. Diese Logik wird mit der PL/SQL-Engine abgebildet. Für Einzelsatz-Verarbeitung werden Object Types (UDT), für Massendaten-Verarbeitung Packages sowie Oracle Advanced Queuing für die Jobsteuerung verwendet. Bei externen, nachrichtengetriebenen Systemen wie Börsen-Anbindungen werden die Messages erst in einer Queue gespeichert, von dort mit einem Scheduler-Job abgearbeitet und in einer Message-Tabelle gespeichert. Bei internen Systemen, die eine asynchrone oder parallele Verarbeitung erfordern, werden die Daten zuerst in einer Inbox-Message-Tabelle gespeichert und von dort über eine Eventqueue abgearbeitet (siehe [Abbildung 1](#)).

Diese Prozesse müssen aktiv überwacht werden, um zeitnah auf negative Verarbeitungs-Performance, Mes-

sage-Staus oder Langzeit-Statistiken reagieren zu können. Das Monitoring soll keine negativen Einflüsse auf die Systemprozesse haben und einfach konfigurierbar sein. Die Daten-Visualisierung soll von der Messdaten-Ermittlung getrennt sein.

Architektur

[Abbildung 2](#) zeigt eine Übersicht über die Monitoring-Architektur. Sie lässt sich in fünf grobe Bereiche unterteilen:

- Messpunkte (rot)
- Konfiguration (grün)
- Messdaten-Ermittlung (orange)
- Messdaten (blau)
- Visualisierung und Alarmierung (grau)

Messpunkte

Die Messpunkte beziehungsweise die zu überwachenden Objekte sind Queue- und Message-Tabellen. Eine wichti-

ge Voraussetzung für eine einheitliche Messwert-Ermittlung sind Standardspalten an den Message-Tabellen. Die wichtigsten Spalten sind „APPLY_START“ (Verarbeitungsstart), „APPLY_END“ (Verarbeitungsende) und „ENQUEUE_ZST“ (Ankunftszeit) vom Typ „TIMESTAMP“ (siehe [Abbildung 3](#)). Anhand dieser Spalten lassen sich millisekundengenau die Verarbeitungs- oder Wartezeiten ermitteln. Die Spalten in den Message-Tabellen gehören zum Entwicklungsstandard.

Die Definition der Queue-Tabellen ist durch Oracle vorgegeben und bietet daher bereits alle notwendigen Spalten. Es werden aktuell mehr als 700 Messpunkte auf sieben Datenbank-Systemen überwacht. Jeder Messpunkt wird gleich behandelt und durchläuft folgendes Dreistufen-Prinzip:

1. Einzelwert

Der Messpunkt enthält die Einzel

werte (siehe Abbildung 2, Message-Tabelle).

2. Aggregation

Durch das Überwachungsintervall werden alle notwendigen Messdaten ermittelt (siehe Abbildung 2, „PA_MONITORING“).

3. Aufbereitung

Durch eine Schnittstelle (siehe Abbildung 2, „V_MONITORING“) werden alle Messwerte für Icinga aufbereitet.

Aus Übersichtlichkeitsgründen sind folgende Messpunkttypen definiert:

- *Processed Messages*
Die Anzahl der im Intervall verarbeiteten Messages. Es werden alle Messages ermittelt, die im Mess-Intervall liegen und einen „APPLY_START“-Zeitstempel besitzen.
- *Waiting Messages*
Die Anzahl der im Intervall unverarbeiteten Messages, also „APPLY_START“ und „APPLY_END“, sind „null“.
- *Max. Processing Time*
Die maximale Verarbeitungszeit aller Messages während des Intervalls in Millisekunden („APPLY_END“ – „APPLY_START“).
- *Max. Waiting Time*
Die maximale (Warte-)Zeit einer unverarbeiteten Message während des Intervalls in Millisekunden („APPLY_START“ – „ENQUEUE_ZST“).
- *Arrived Time*
Die maximale Dauer für den Message-Empfang aller Messages innerhalb des Intervalls („ENQUEUE_ZST“ – „SENDING_TIME“).

Konfiguration

Ein weiterer elementarer Bestandteil ist die Konfiguration der Messpunkte sowie deren Namen, Schwellwerte etc. Nachfolgend eine Auflistung der wesentlichen Konfigurationspunkte:

- *Messpunkt*
Der einzelne Messpunkt ist über einen Eintrag in der Tabelle „MONITORING_CONFIG“ definiert.
- *Messpunkt-Typ*
Die einzelnen Typen wurden bereits vorhin (siehe Messpunkte) erläutert.

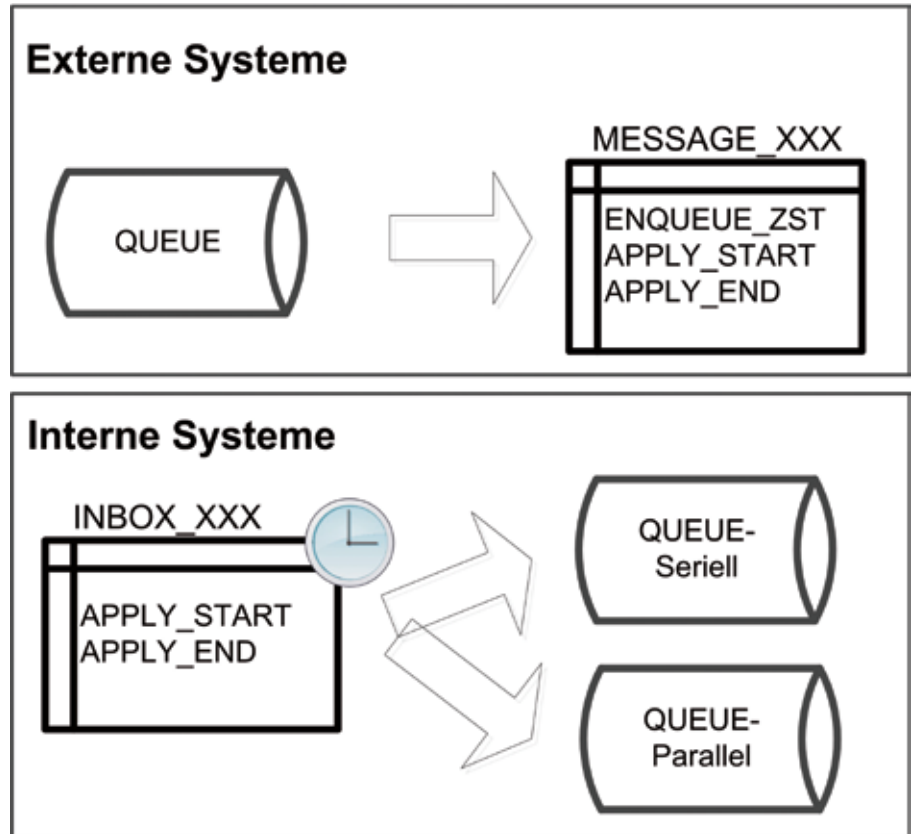


Abbildung 1: Anbindungsarten und deren Verarbeitung

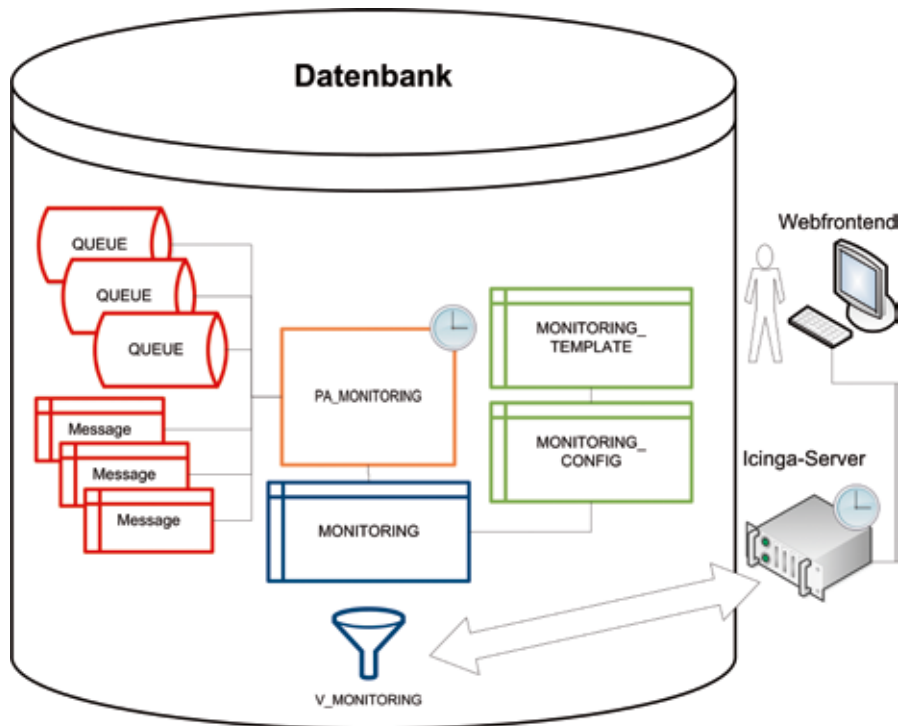


Abbildung 2: Übersicht über die Monitoring-Architektur

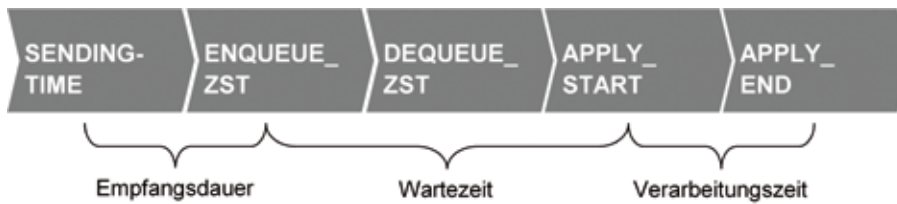


Abbildung 3: Wichtige Zeitpunkte bei der Message-Verarbeitung

tert. Jeder Messpunkt muss einem Typ zugeordnet sein.

- **Messpunkt-Bezeichnung**
Jede Messpunkt-Bezeichnung ist eindeutig pro überwachte Datenbank. Die Bezeichnung wird im Alarmierungsfall ausgegeben und kann im Web-Frontend gesucht werden.
- **Schwellwerte**
Schwellwerte dienen bei Überschreitung eines Messwertes zur Alarmierung. Jeder Messpunkt erhält individuelle Schwellwerte. Bei den Schwellwerten unterscheidet man wiederum:
 - **Warnung**
Jede Überschreitung der Warnschwelle wird gemeldet.
 - **Kritisch**
Kritische Schwellwerte sind per Definition größer als die Warnschwelle. Jede Überschreitung dieses Schwellwert-Typs sollte unbedingt zeitnah verfolgt und geklärt werden.
- **Template**
Ein Template dient zur einheitlichen Darstellung von Messdaten. In diesem Fall haben sich die Autoren auf unterschiedliche Hintergrundfarben der Messpunktcharts je Datenbanksystem beschränkt. Icinga bietet jedoch eine große Auswahl an Darstellungsmöglichkeiten durch sogenannte „Special-Templates“. Damit ist es möglich, mehrere Messpunkte in einem Chart zu kombinieren. Dazu später mehr im Abschnitt „Visualisierung“.

Messdaten-Ermittlung

Zu jedem Messpunkt-Typ gibt es eine dynamische Abfrage (siehe „PA_MONITORING“), der der jeweilige Messpunkt mitgeteilt wird. So wird sicher-

gestellt, dass jedem Messpunkt-Typ die gleiche Ermittlungslogik zugrunde liegt und die Messwerte untereinander vergleichbar sind.

Das Intervall zur Messdaten-Ermittlung muss nach den persönlichen Anforderungen bestimmt werden. Dabei ist zwischen einer hohen Taktung mit einer besseren Monitoring-Genauigkeit und der dadurch erzeugten Systemlast (sogenanntes „Grundrauschen“) abzuwägen. In diesem Fall wurde entschieden, ein minutliches Intervall zu verwenden.

Das Intervall ist entscheidend für die Konfiguration im Zusammenspiel mit Icinga. Denn hier gilt das Alles-oder-nichts-Prinzip. Eine Mischung aus minutlichen und beispielsweise sekundlichen Intervallen je nach Messpunkt ist in der beschriebenen Konfiguration nicht möglich, da Icinga alle Messpunkte gleichzeitig abfragt. Bei Intervall-Mischung würde dies zu einer falschen oder fehlerhaften Darstellung der Messwerte führen.

Die Messdaten sind in der Tabelle „MONITORING“ gespeichert und nach kurzer Zeit reorganisiert. Die Schnittstelle zu Icinga bildet die View „V_MONITORING“. Icinga fragt minutlich die aktuellen Messdaten ab.

Visualisierung und Alarmierung

Zur Darstellung der Messwerte wurde Icinga ausgewählt. Icinga ist eine sogenannte „Fork“ von Nagios. Diese freie Software wurde im Hause bereits von der Systemtechnik genutzt, um beispielsweise die Netzwerk-Auslastung oder andere Rechenzentrums-Informationen zu überwachen:

- **(Special-)Templates**
Zur Darstellung von Messpunkten, auch Checks genannt, werden in der Regel Templates verwendet. Jeder Messpunkt ohne definier-

tes Template verwendet ein Default-Template. Eine besondere Ausprägung sind die sogenannten „Special-Templates“. Hier können viele Parameter eines Charts vorgegeben werden wie Hintergrundfarbe, Diagrammtyp (Balken oder Linie), Farben der Balken oder Linien, Achsenbeschriftung und Titelbezeichnung. Außerdem bietet ein solches Template die Möglichkeit, mehrere Messpunkte in einem Chart darzustellen, beispielsweise Eingangs- und Ausgangs-Messages für alle Systeme.

- **Dashboard**
Das Web-Frontend von Icinga bietet alle Informationen zu den Messpunkten. Jedoch fehlt eine komprimierte Gesamtansicht bestimmter Messpunkte über alle Datenbanksysteme. Die zuvor beschriebenen Special-Templates können dazu genutzt werden, eine eigene Übersicht beziehungsweise ein Dashboard individuell zusammenzustellen. Die Dashboard-Anzeige wird als eine Fernseher- und Arbeitsplatzrechner-Ansicht mit Drill-Down-Funktionalität angeboten.
- **Alarm**
Nachfolgend eine kurze Auflistung der möglichen Alarmierungswege:
 - **E-Mail:**
Icinga übernimmt die Überwachung der Messpunkte und alarmiert standardmäßig per E-Mail.
 - **Nagios Status Monitor (NagStaMon)**
Das Nagios-Desktop-Tool „NagStaMon“ bietet eine E-Mail-Alarm-Alternative. Das Tool verbindet sich neben Nagios- auch mit Icinga-Servern und visualisiert eventuelle Schwellwert-Überschreitungen mithilfe eines Statusbalkens auf dem Desktop (siehe <http://nagstamon.ifw-dresden.de>).

Umgebungen

Es geht in erster Linie darum, die Produktions-Umgebung zu überwachen. Um negative Performance-Änderungen vor der Produktions-Übernahme erkennen zu können, sind auch eine frühzeitigere Überwachung der Pre-

Produktion und User-Acceptance-Test-Systeme (UAT) sinnvoll. Es wurde entschieden, auch in Integrations- und QS-Umgebung(en) das Monitoring zu ermöglichen.

Dabei ist jedoch zu beachten, dass sich beispielsweise die Messwerte aus der Produktion nicht direkt mit der QS-Umgebung vergleichen lassen, da die jeweiligen Umgebungen unterschiedliche Hardware-Konfigurationen verwenden. Es kann hier also nur die Performance der Integration mit der Integration-Umgebung und die der QS mit der QS-Umgebung verglichen werden.

Erfahrungen und Herausforderungen

Folgende Besonderheiten sind während der vergangenen Jahre im Laufe der Entwicklung der Monitoring-Landschaft aufgefallen:

- **Einheitliche Messwert-Ermittlung**
Eine einheitliche Messwert-Ermittlung ist extrem wichtig. Es sollte dringend vermieden werden, pro Messpunkt oder Datenbank immer wieder neue Abfragen zur Messwert-Ermittlung zu schreiben.
- **Verwendung von Subviews bei Queue-Abfragen**
Leider enthalten die „AQ\$“-Views zu den Queues nicht alle die erwarteten Daten. Daher musste teilweise auf Data-Dictionary-Subviews (wie „*_H“, „*_L“, „*_S“) zurückgegriffen werden.
- **Sommer-/Winterzeit bei Queues**
Die Messwerte sind abhängig vom Queue-Erstellungszeitpunkt (Sommer- oder Winterzeit). Wurde eine Queue während der Sommerzeit angelegt, wird die Enqueue-Time um eine Stunde reduziert. Dies kann zu verwirrenden Messwert-Ergebnissen führen.
- **Enqueue nur sekundengenau**
Der Enqueue-Zeitstempel bietet lediglich „DATE“-Genauigkeit. Wer mehr Genauigkeit will, also Millisekunden („TIMESTAMP“), muss selbst tätig werden. Als Workaround kommt der „JMS Payload“-Type zum Einsatz. Dieser besitzt ein zusätzliches Attribut mit Timestamp-Genauigkeit.

- **Messpunkt-Bezeichnung**

Es ist darauf zu achten, dass die Messpunkt-Bezeichnung nicht zu lang wird. Icinga kennt zwar keine Längen-Begrenzung, aber es wird gegebenenfalls nicht die komplette Bezeichnung darstellt. Daher ist hier eine gewisse Namenskonvention (etwa „<Schema>.<Tabelle>.<Typ>“) auch aus Übersichtlichkeitsgründen sehr hilfreich.

- **Automatische Messpunkt-Erstellung**

Nutzt man eventuell mehrere Message-Typen in einer Queue und möchte pro Typ Messwerte erhalten, ist bei neuen Typen immer die Konfiguration zu erweitern. Einfacher ist es, man lässt sich beim ersten Auftreten eines neuen Typs automatisch einen neuen Konfigurationseintrag mit Standard-Schwellwerten erstellen.

- **Automatische De-/Reaktivierung von Messpunkten**

Speziell bei der Einführung neuer Messpunkte kann es schwierig sein, realistische Schwellwerte zu definieren. Gelingt einem dies nicht, kann man nach der Einführung neuer Messpunkte möglicherweise durch viele überflüssige Alarm-E-Mails überrascht werden. Eine weitere Herausforderung besteht darin, dass die Messwert-Ermittlung trotz des Test-Monitorings auf den Test-Systemen langsamer in Produktion läuft. Hierfür wurde eine automatische, temporäre De- und Reaktivierung einzelner Messpunkte eingeführt. Läuft beispielsweise eine Messwert-Ermittlung länger als fünf Sekunden, wird dies protokolliert. Geschieht dies dreimal hintereinander, wird der Messpunkt für eine bestimmte Zeit deaktiviert. Der fehlerhafte Messpunkt wurde nun bereits dreimal gemeldet. Nach Ablauf des Zeitfensters reaktiviert sich der Messpunkt und versucht sich wieder an der Messwert-Ermittlung.

Fazit

Die beschriebene Umsetzung bietet eine kostengünstige Monitoring-Umgebung: Man benötigt einfache PL/SQL- und SQL-, jedoch tiefere Icinga-Kenntnisse. Wer keine Out-of-the-Box-

Lösung erwartet und einen gewissen Startaufwand zum Aufbau und zur Konfiguration nicht scheut, erhält ein Monitoring-Paket, das sich an viele eigene Bedürfnisse anpassen lässt. Durch die Verwendung von Icinga anstelle von Nagios bekommt man gegebenenfalls schnelle Unterstützung von den Entwicklern, falls Erweiterungen dort gewünscht sind.

Diese Konfiguration ist nun schon seit einigen Jahren im Einsatz. Dank der eigenen Monitoring-Logik in PL/SQL können schnell und problemlos neue Messpunkte ergänzt und bei Bedarf optimiert werden. Störungen werden jetzt schneller erkannt und können deutlich zügiger behoben beziehungsweise kommuniziert werden.

Kai Pillatzki
(Datenbank-Entwickler)
kai.pillatzki@berenberg.de



Stefan Triep
(Icinga)
stefan.triep@berenberg.de



Andriy Terletskyy
(Datenbank-Architektur)
andriy.terletskyy@berenberg.de

