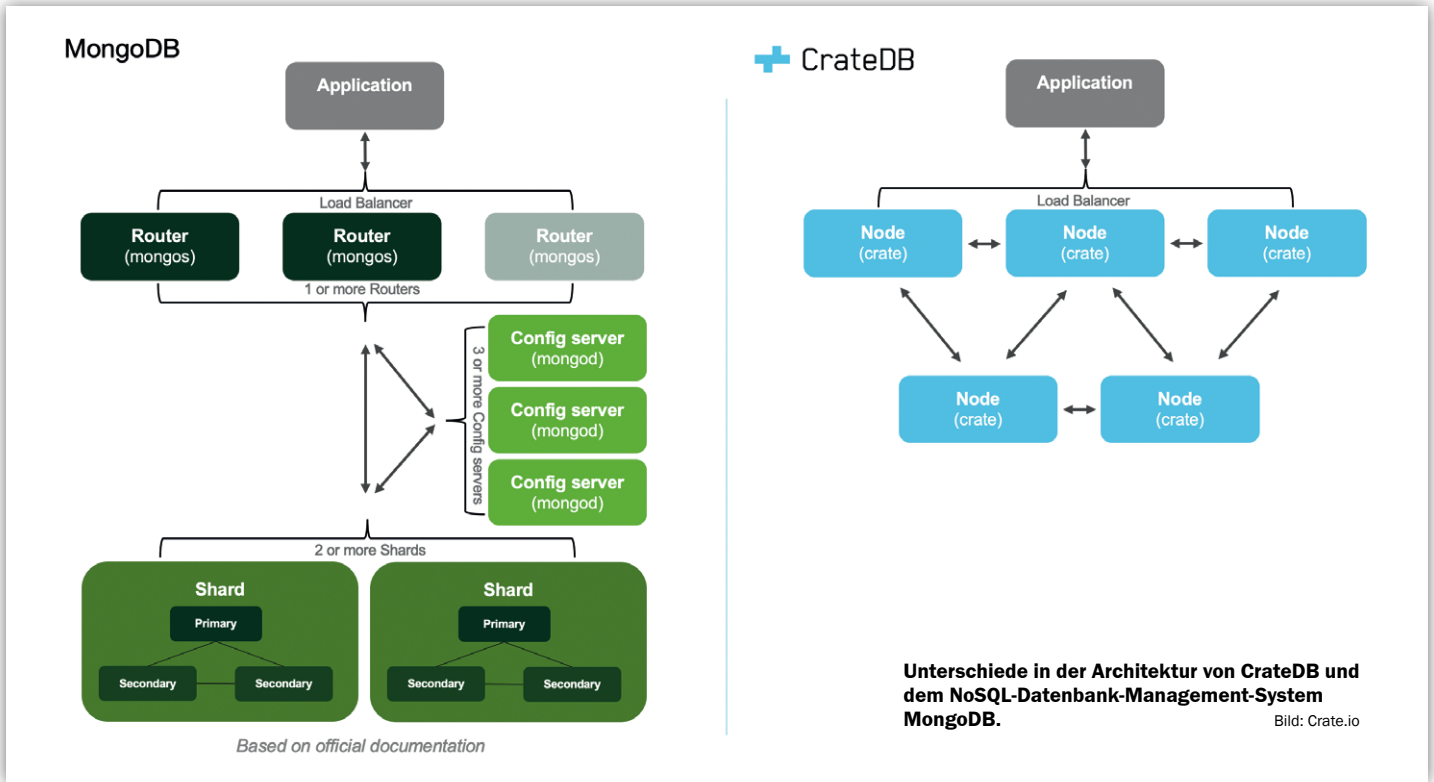
Weniger Datenbank-Wildwuchs

Datenbank für das IIoT

Sowohl das Industrial Internet of Things (IIoT) als auch Big Data sind sehr datenintensiv. Die sich daraus ergebende heterogene Infrastruktur der Datenbanken bedeutet eine stetig komplexere Administration. Um dies langfristig zu vermeiden, ist ein prüfender Blick auf die verwendeten Datenbanken unerlässlich und gegebenenfalls eine Konsolidierung nötig, denn für das IIoT stehen bei Bedarf spezielle Datenbanken bereit.

So nachvollziehbar es auch sein mag, die für den jeweiligen Zweck auf den ersten Blick am besten geeignete Datenbank zu nutzen, so ineffizient und teuer kann dies auf lange Sicht sein. NoSQL für skalierende Data Analytics, SQL für relationale Datenbanken und auch Time-Series-Datenbanken (TSDB) kommen im IIoT zum Einsatz, von proprietären Datenbankformaten ganz zu schweigen. Doch diese Systeme verteilen sich in aller Regel nicht nur auf unterschiedliche Orte, von lokal im Betrieb oder im eigenen Rechenzentrum bis hin zu verschiedenen Clouds, sie bringen auch einen enormen Pflege- und Administrationsaufwand mit sich. Die Komplexität dieser Aufgabe zeigt unter anderem die Tatsache, dass jedes System mit einer spezifischen Sprache programmiert ist. Außerdem unterscheiden sich einzelne Datenbanken zum Teil stark in ihrem Funktionsumfang und im Betrieb. Jede neue Version ist daher eine Herausforderung für Administratoren. Wer hier zumindest auf den SQL-Standard hofft, hofft vergebens, da die meisten skalierenden Datenbanksysteme diesen nicht unterstützen. Dabei ist eine Kombination von Daten aus verschiedenen Systemen in den meisten Anwendungsfällen nötig. Dies umfasst strukturierte Daten aus relationalen Systemen wie ERP, riesige Mengen an unstrukturierten Daten wie JSON (JavaScript Object Notation) und möglicherweise binäre Daten (BLOBs).

Das laufende Zusammenführen dieser, die Synchronisation und das gegenseitige Verlinken sind zwingend nötig. Durch den Betrieb mehrerer Datenbanken, durch mehrfach gehaltene Daten und die dadurch höheren Betriebskosten in der Cloud entstehen jedoch auch erhebliche Kosten. Diese wildwachsende Struktur, die es fast unmöglich macht, das System in seiner Gesamtheit von möglichst wenigen Personen bedienen zu lassen und die Schulungen, die das Fachpersonal besuchen muss, um auf dem neuesten Stand zu bleiben, treiben die Betriebs- und Personalkosten dort in die Höhe, wo der Einsatz von Datenbanken eigentlich



Unterschiede in der Architektur von CrateDB und dem NoSQL-Datenbank-Management-System MongoDB. Bild: Crate.io

helfen soll, diese zu reduzieren. IT-Verantwortliche sollten sich daher genau überlegen, ob sie diesen Wildwuchs eindämmen, indem sie ihre Datenbanken und ihre Daten konsolidieren. Während ersteres durch Vereinheitlichung und Integration die heterogenen Datenbankformate in ein neues Datenbanksystem ablöst, erfordert die Datenkonsolidierung direkte Eingriffe in die Daten, um Abfragen quer durch alle Daten zu ermöglichen, Inkonsistenzen auszugleichen und Fehler zu bereinigen. Um zu visualisieren, welche Idee dahinter steckt, ist Folgendes vorwegzunehmen: Im Industrial IoT ist es Usus, Sensordaten zunächst in einer TSDB zu speichern, von wo aus die Datenüberführung in Form von aggregierten Daten in ein NoSQL-System erfolgt. Dort lassen sie sich nicht nur wesentlich einfacher, sondern auch schneller analysieren. Somit sind die jeweiligen Vor- und Nachteile der Datenbanksysteme ausgeglichen – scheinbar ein logischer Schritt, um das Optimum aus der Masse der Daten herauszuholen.

Was dabei jedoch aufgrund der jahrelang organisch gewachsenen Datenbankgrößen mitunter keine Beachtung in der Planung findet, ist die Tatsache, dass sich auch das Angebot der Datenbankanbieter auf diese Situation eingestellt hat. Längst gibt es eine neue Generation an Datenbanken, die speziell für Maschinendaten entwickelt ist. Beispielsweise verbindet CrateDB die Vorteile von NoSQL und TSDB, während sie gleichzeitig den SQL-Standard mit Postgres unterstützt. Zudem hält sie die Daten auf einem leistungsfähigen und rasch skalierbaren Cluster in der Cloud vorrätig und unterstützt gleichzeitig den lokalen Betrieb (Edge). Dadurch lassen sich die Daten effizient speichern und der Arbeitsaufwand für Entwickler und die Administration sinkt deutlich. Die dadurch eingesparten Server und Leistungen haben auch betriebswirtschaftliche Vorteile. Kostenersparnisse können Unternehmen durch eine optimierte Architektur erzielen, indem zum Beispiel eine Konsolidierung von zwei oder drei Systemen auf eine relationale und unstrukturierte Daten verarbeitende Datenbank erfolgt. Dies macht die Arbeit der eingelernten Teams leichter, da sie auf den Einsatz bestehender Werkzeuge innerhalb des SQL-Standards ebenso bauen können wie auf eine leichte Integration neuer Applikationen beziehungsweise einen einfachen Austausch vorhandener Applikationen und die Einhaltung des vertrauten SQL-Standards.

Erfolgreiche Konsolidierung

Unternehmen, die das Ziel der Konsolidierung gefasst haben, sollten sich darüber im Klaren sein, dass diese zunächst einen einmaligen Mehraufwand bedeutet, der sich jedoch schnell amortisiert. Die optimale Bereitstellung der Dateninfrastruktur auf Basis von Standards ist eine wichtige Voraussetzung, um möglichst umfassend von aktuellen Trends wie Real-Time Data Analytics, Industrial IoT oder Machine Learning zu profitieren.

Als erster Schritt vor der Umsetzung sollte daher eine genaue Planung des Bedarfs und der Ziele erfolgen. Welche Daten und welche Datenbanken will man sinnvollerweise konsolidieren? Welche Anwendungsfälle profitieren davon und welche neuen Use Cases sind dadurch möglich? Erst wenn diese Fragen beantwortet sind, geht es an die technische Seite der Konsolidierung. Damit die Datenformate einheitlich sind, benötigt das Projekt Konnektoren zwischen Quell- und Zieldatenbank. Wenn diese fehlen, sollten zumindest auf beiden Seiten der Konsolidierungs-Pipeline Import/Export-Filter für ein gemeinsames Zwischenformat wie XML vorhanden sein, um auch ältere oder exotische Datenformate in die Zieldatenbank importieren zu können. Auch hier erleichtert der vorhandene SQL-Standard den Vorgang erheblich. Dies lässt sich einerseits darin begründen, dass fast jedes Tool über einen Postgres-Treiber verfügt. Auf der anderen Seite lassen sich die Daten mit SQL herstellerunabhängig lesen, verschieben und speichern. Christian Lutz/am

Christian Lutz ist Co-Gründer von Crate.io.

Marktübersicht: IoT-Gateways

Hersteller/Anbieter	Web
Adva Optical Networking	www.advaoptical.com
Axiomtek	www.axiomtek.de
Axotec	www.axotec.de
Bosch Rexroth	www.boschrexroth.com
Cisco	www.cisco.com
Comp-Mall	www.comp-mall.de
Dell EMC	www.dell.com
Dragino	www.dragino.com
Duagon	www.duagon.com
E.E.P.D.	www.eepd.de
Endian	www.endian.com
FP InnovoLabs	www.inovolabs.com
Hager	www.hager.de
Hilscher	www.hilscher.com
HPE	www.hpe.com
InHand Networks	www.inhandnetworks.com
in.hub	www.inhub.de
Intel	www.intel.com
IoTmaxx	www.iotmaxx.com
IPC2U	www.ipc2u.de
Ixon	www.ixon.cloud
Janztec	www.janztec.com
Lantronix	www.lantronix.com
MB Connect Line	www.mbconnectline.com
Moxa	www.moxa.com
NCP Engineering	www.ncp-e.com
Neousys	www.neousys-tech.com
Onlogic	www.onlogic.com
Owasys	www.owasys.com
Red Lion	www.redlion.net
Siemens	www.siemens.com
Welotec	www.welotec.com