

Digitalisierung, Industrie 4.0 und Künstliche Intelligenz (KI) in der Produktion stellen Ihr Unternehmen vor immer neue Herausforderungen. Smarte Maschinen und Anlagen, deren Vernetzung, aber auch eine intelligente Gebäudeautomatisierung erzeugen rund um die Uhr eine Flut von Sensordaten. Um Prozesse effizienter zu gestalten, Abhängigkeiten zu erkennen oder Störungen bereits vor deren Eintreten zu bemerken, muss diese Vielzahl an Informationen zuerst abgespeichert, jedoch gleichzeitig auch für Analysen zugänglich gemacht werden.

In diesem *Nachgelesen* möchten wir Ihnen die Herausforderungen im Umgang mit Sensor- und Zeitreihendaten näherbringen. Erfahren Sie, wie Sensordaten mittels freier Soft-

ware auch in Ihrem Unternehmen performant gespeichert und visualisiert werden können.

Im Detail lesen Sie:

- was man unter Zeitreihendaten versteht und wo Daten entstehen,
- welche Datenbanksysteme für die Ablage zur Verfügung stehen, wie Daten vorbereitet werden und was Sie bei der Planung dieser Systeme beachten sollten,
- wie Sie von den Datensätzen zu beeindruckenden Visualisierungen gelangen und
- welche Tricks und Kniffe möglich sind, um die Performance zu erhöhen.

Die Datenflut

Fast jede Maschine oder produktionstechnische Anlage besitzt gegenwärtig eine Vielzahl von Sensoren um ihre Funktion zu gewährleisten, die Qualität zu überwachen oder auf Änderungen reagieren zu können. Vielleicht haben Sie auch im Rahmen eines Retrofit Bestandsanlagen mit erweiterter Sensorik ausstatten lassen, um zusätzliche Betriebsdaten zu erheben. Zusätzlich stehen oftmals auch Sensordaten aus der Gebäudetechnik oder von Logistikkitteln zur Verfügung. Neben der primären Ver-

wendung der Daten innerhalb der Anlage können sich durch deren Speicherung, Auswertung und vor allem Kombination mit anderen Daten eine Vielzahl von Zusatznutzen ergeben, die Sie in Ihrem Unternehmen nicht ignorieren sollten.

Steht eine Rollenbahn wiederkehrend still, ist dieses Wissen nicht allein an dieser Anlage von Interesse. Im besten Fall würden nicht nur vor- oder nachgelagerte Prozesse automatisiert gedrosselt, sondern sogar

ein Wartungsmitarbeiter informiert. Mehrere Stillstände summieren sich, doch quantitative Aussagen über die Stillstandzeiten lassen sich nur treffen, wenn die Sensordaten auch gespeichert wurden und anschließend analysiert werden können. Was hat zum Stillstand geführt? Das Erkennen von Mustern in den Daten ist ein Paradebeispiel für die Anwendung von KI-Software. Ist die Anzahl der Anlagen oder der Störungen überschaubar und ist, wie in vielen KMU, Personal mit Expertenwissen durch langjährige Erfahrung mit der Anlage vorhanden, lässt sich durch manuelle Sichtung der Sensordaten neues Wissen erzeugen. Wie sah der Verlauf der Stromaufnahme der Antriebsmotoren oder deren Temperaturverlauf kurz vor den Störungen in der letzten Woche aus?

Sensordaten lassen sich hier als Zeitreihendaten klassifizieren. Wir verstehen eine Zeitreihe als zeitlich geordnete Sequenz diskreter Datenpunkte¹. Zusammen mit den Datenpunkten sollte auch der genaue Zeitpunkt ihrer Aufnahme erfasst werden. Die nötige Genauigkeit ist aus der jeweiligen Anwendung abzuleiten. Für einige Anwendungsfälle sollten die Zeitstempel unter Angabe von Millisekunden abgelegt werden. Es sollte ebenfalls darauf geachtet werden, die Uhren der beteiligten IT-Systeme über einen Zeitserver (NTP) miteinander zu synchronisieren. Sollten die Systeme aus Sicherheitsgründen nicht mit dem Internet verbunden sein, kann ein NTP-Server auch im unternehmenseigenen Netz betrieben werden.



Ein Anwendungsbeispiel

Dieses Nachgelesen veranschaulicht die Thematik im weiteren Verlauf am praktischen Beispiel der Erfassung, Speicherung und Visualisierung von Energiedaten, die in der Experimentier- und Digitalfabrik des Mittelstand 4.0-Kompetenzzentrum Chemnitz an Anlagen und Gebäuden anfallen.

Multifunktionsgeräte PAC4200 zum Einsatz (Abb.1). Diese verfügen über eine Ethernet-Schnittstelle, über welche 12 elektroenergiebezogene Messwerte gleichzeitig mittels des Modbus-Protokolls abgefragt werden. Diese Abfrage erfolgt zyklisch, im Intervall von 1 Sekunde. Bei Modbus handelt es sich um ein offenes Protokoll zur Kommunikation bspw. mit speicherprogrammierbaren Steuerungen und es stehen Implementierungen in den gängigen Programmiersprachen zur Verfügung. Um

Daten sammeln

Im betrachteten Szenario kommen als Sensorik zehn Siemens SENTRON

später viele Datenströme aus unterschiedlichen Quellen in unterschiedlichen Formaten und Protokollen verarbeiten und kombinieren zu können, bietet sich die Verwendung einer an Datenstrom orientierten Entwicklungsplattform an.

Unsere Umsetzung wurde mit der Open-Source-Software Node-RED implementiert. Diese Umgebung ist durch Plugin-Module erweiterbar. So lassen sich bspw. das Modbus-Protokoll sowie Module für die Kommunikation mit SPS- oder PC-basierten Steuerungen wie S7² oder Twincat ADS³ integrieren.



Abb. 1: Multifunktionsmessgerät PAC4200

Wohin mit den Daten?

Der folgende Abschnitt gibt einen Überblick über verschiedene Datenbanksysteme, die zur Ablage von Sensordaten genutzt werden können. Innerhalb des Anwendungsbeispiels zeigen wir, was bei der Installation und Konfiguration des Datenbanksystems zu beachten ist.

Grundsätzlich wird bei Datenbanksystemen zwischen

- relationalen (bspw. MySQL, PostgreSQL) und
- nicht-relationalen Systemen (NoSQL, „Not only SQL“ wie bspw. MongoDB, Redis)

unterschieden. Relationale Datenbanken speichern die Informationen in strukturierten Tabellen, die zueinander in festgelegten Relationen stehen. Bekannte Vertreter der nicht-relationalen Systeme, wie die dokumentenorientierte Datenbank MongoDB, speichern die Daten als Dokumente (Documents) innerhalb von Sammlungen (Collections). Dabei müssen die Datensätze nicht in einer festen Relation stehen und auch nicht dem gleichen Schema folgen. Denken Sie dabei vielleicht an eine Sammlung von Büchern in einem Bücherregal. Jedes Dokument in einer Sammlung kann andere Informationen enthalten und in der Größe variieren. Dies bringt Flexibilität gegenüber den relationalen Modellen und kann in der

Praxis eine schnellere Implementierung bedeuten, was wiederum nützlich für eine Prototypenentwicklung sein kann.

Bei der kontinuierlichen Erfassung und Speicherung von Sensordaten besteht die größte Herausforderung in der Skalierbarkeit der beteiligten Systeme. Betrachten wir bspw. ein System mit 20 Sensoren, die sekundlich einen Datenwert liefern. In 30 Tagen liefert dieses System also 51,8 Millionen Datensätze, die nicht nur auf den Speicher geschrieben werden müssen, sondern auch indiziert werden, um möglichst schnell bei Abfragen wieder gefunden werden zu können. Systeme wie MongoDB können hier im Vergleich mit relationalen Systemen ihre Vorteile nicht ausspielen. Selbst bei speziell auf die Systeme angepassten Implementierungen, etwa dem Zusammenfassen von mehreren Datensätzen in einem Dokument, hängt die Performance im Vergleich zurück⁴. Zudem ist der Speicherverbrauch nicht aus dem Fokus zu verlieren. Im Gegensatz zu relationalen Modellen, bei denen der Datentyp für jede Spalte einer Tabelle im Voraus definiert wird und somit optimal abgelegt werden kann, ist bei der Ablage der gleichen Daten in ein NoSQL-System ein Mehrverbrauch um den Faktor 10 möglich.

Selbst relationale Datenbanksysteme wie PostgreSQL brechen in der Performance ein, sobald der Datenbestand zu groß wird⁵. Beim Schreiben

jedes neuen Datensatzes müssen Indizes angepasst werden. Dies ist nur mit hoher Geschwindigkeit möglich, wenn der Index im Arbeitsspeicher (RAM) gehalten werden kann. Wächst jedoch der Datenbestand und damit auch der Index zu stark, so kann dieser nicht mehr im flüchtigen Speicher gehalten werden. An dieser Stelle setzen auf Zeitreihendaten spezialisierte Systeme mit weiteren Optimierungen und zusätzlichen Strategien an.

TimescaleDB, eine Erweiterung für PostgreSQL spezialisiert auf den Umgang mit Zeitreihendaten, implementiert das Konzept von Hypertabellen (Hypertables), welche die eigentliche Datentabelle im Hintergrund für den Nutzer unbemerkt in kleinere Tabellen partitioniert. Somit ist es möglich, den Index zu jeder Zeit im Arbeitsspeicher zu halten.

Unser Tipp:

- Standardmäßig erfolgt die Unterteilung der Tabellen wöchentlich. Doch je nach Anwendungsszenario oder in Abhängigkeit von der Größe des Arbeitsspeichers kann dieses Intervall zu groß sein. Behalten Sie deshalb den Datenbestand und die Ausführungszeiten von Abfragen im Auge und passen Sie die Parameter für die Hypertables bei Bedarf an.

Im Anwendungsbeispiel haben wir die Erweiterung TimescaleDB in ein PostgreSQL 12 installiert. Danach

ist jede Tabelle einmalig durch das Ausführen einer Anweisung in eine Hypertabelle zu überführen. Den Rest der Arbeit erledigt TimescaleDB von nun an im Hintergrund.

Von Datensätzen zur Visualisierung

In den vorangegangenen Absätzen haben Sie erfahren, welche Voraussetzungen an ein Datenhaltungssystem für Zeitreihendaten gestellt werden und wie eine praktische Umsetzung aussehen kann. Im Folgenden betrachten wir die Visualisierung der Daten näher.

Unsere Node-Red-Anwendung schreibt die über Modbus ausgelesenen Daten der Multifunktionsmessgeräte in jeweils eine Tabelle pro Gerät. Der Vorteil: Indizes wachsen nicht unnötig an.

Um die Daten zu analysieren und bspw. Stillstandszeiten, Muster, Zusammenhänge mit anderen Daten, möglichen Verschleiß oder Abweichungen zu erkennen, eignet sich eine grafische Visualisierung der Daten. Auch für diesen Schritt greifen wir auf freie Software zurück.

Die plattformübergreifende Anwendung Grafana ist ein Werkzeug zur Visualisierung von Rohdaten in verschiedenen grafische Anzeigeformen und eignet sich perfekt für die Abfrage und Visualisierung von Zeitreihendaten.

Wie auch bei PostgreSQL handelt es sich bei Grafana um eine Server-Anwendung, die Sie einmalig zentral installieren. Danach ist Grafana über den Webbrowser auf anderen Geräten im Netzwerk erreichbar.

```

1 SELECT * FROM public."pac4200_10-54-120-88"
2 ORDER BY "time" DESC LIMIT 100
3

```

time	spannung_11	spannung_12	spannung_13	strom_11	strom_12	strom_13	gesamtscheinleistung	gesamtwirkleistung	gesamtblindleistung
[PK] timestamp with time zone	real	real	real	real	real	real	real	real	real
1	234.59055	232.2417	232.36159	3.5624125	0	0	835.7083	492.3575	-84.26233
2	2021-02-10 14:08:43.086+01	234.5154	232.20741	232.36218	3.5561829	0	833.9796	493.36353	-84.63307
3	2021-02-10 14:08:41.087+01	234.53583	232.19553	232.34476	3.5727956	0	837.9486	495.3488	-84.30739
4	2021-02-10 14:08:40.084+01	234.53793	232.0725	232.31468	3.5805957	0	839.7855	496.2991	-83.96396
5	2021-02-10 14:08:39.085+01	234.5859	232.18916	232.40244	3.5913599	0	842.4824	498.0411	-83.78642
6	2021-02-10 14:08:38.085+01	234.48715	232.10133	232.32724	3.6068897	0	845.77167	498.66934	-83.50982
7	2021-02-10 14:08:37.086+01	234.54556	232.15845	232.38628	3.6144855	0	847.76154	501.04724	-84.93158
8	2021-02-10 14:08:36.087+01	234.64284	232.22089	232.47554	3.6202707	0	849.4692	501.8868	-85.179306
9	2021-02-10 14:08:35.088+01	234.62221	232.14791	232.40341	3.6297452	0	851.61884	503.86642	-84.99029
10	2021-02-10 14:08:34.086+01	234.59007	232.15923	232.38393	3.6346216	0	852.6462	505.0143	-85.48883
11	2021-02-10 14:08:33.086+01	234.62328	232.12966	232.37637	3.6577191	0	858.18604	508.06534	-85.59442
12	2021-02-10 14:08:32.084+01	234.63504	232.14467	232.40474	3.6673596	0	860.5333	509.29767	-86.24506

Abb. 2: Energiedaten in der Hypertable

Unser Tipp:

- Die Konfiguration von Grafana erlaubt es, einzelne Diagramme und Visualisierungen in bereits bestehende Dashboards und Webanwendungen im Unternehmen einzubinden und die vorhandenen Anwendungen so aufzuwerten.

Neben Linien- und Flächendiagrammen bietet die Anwendung eine Vielzahl weiterer Visualisierungsoptionen (Abb. 3).

Diagramme lassen sich bildschirmfüllend darstellen oder in variabler Größe und Anzahl zu ganzen Dash-

boards kombinieren. Ob mehrere Y-Achsen, das Überschreiben von Datenreihennamen oder die Anzeige von Alarmbereichen – Grafana bietet zahlreiche Möglichkeiten, die eigenen Daten je nach Bedarf abzubilden. Selbst bei Bestandsanlagen kann eine solche Visualisierung ganz neue Betrachtungswinkel eröffnen und helfen, Prozesse zu hinterfragen oder Details besser zu verstehen.

Die Abbildungen 4 und 5 geben einen Einblick über den Umfang der Funktionalität und sollen Ihnen einen Eindruck vermitteln, wie Sie die eigenen Daten durch geeignete grafische Darstellung neu erleben können.



Abb. 3: Visualisierungsoptionen

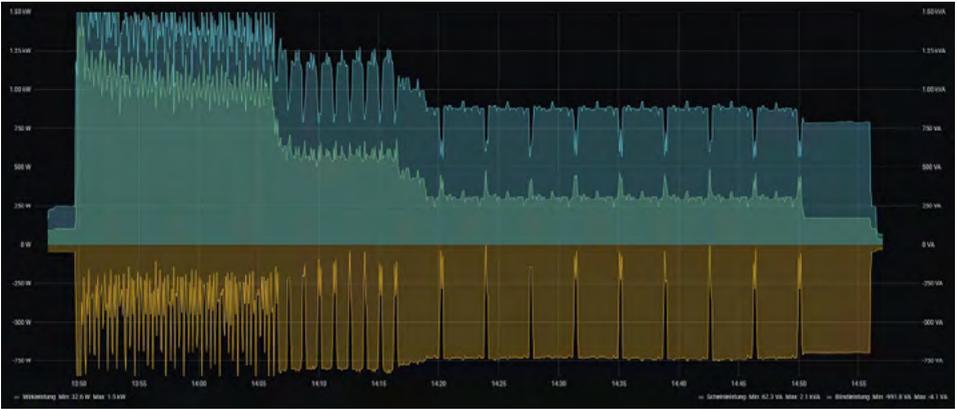


Abb. 4: Energiedatenvisualisierung in Grafana



Abb. 5: Dashboard in Grafana

Die Abfrage der darzustellenden Daten erfolgt über SQL. Neben der Option Ihre Datenbankanfragen direkt in SQL zu formulieren, gibt Ihnen Grafana auch die Möglichkeit, eine Art eingebauten Baustein-Editor zu benutzen, dessen Ausgabe im Hintergrund automatisiert in SQL überführt wird.

Die Unterschiede können Sie der folgenden Abbildung 6 entnehmen.

Unser Tipp:

- Machen Sie sich mit den eingebauten Aggregations- und Sonderfunktionen, die die Erweiterung

TimescaleDB mitbringt, vertraut. Sie ermöglichen Ihnen, die Performance Ihrer Abfragen stark zu erhöhen.

Werte zurückzuliefern, die bspw. aus Mittelwerten von festgelegten Zeitintervallen bestehen.

Unser Tipp:

Der Zeitbereich für die Anzeige kann in Grafana frei gewählt werden. Dabei ist zu bedenken, dass das sekundliche Ablegen eines einzigen Messwertes in einer Woche über 600.000 Datensätze erzeugt. Das Datenbanksystem ist in der Lage all diese Datensätze als Abfrageergebnis zurückzuliefern. Jedoch können Bildschirme aufgrund ihrer Auflösung die Datenpunkte nicht darstellen. Deshalb sollte die Abfrage dahingehend angepasst werden, nur einen Bruchteil der

- Grafana kann aus Anzeigzeitraum und Bildschirmgröße die Zeitintervalle ableiten und TimescaleDB anweisen, den betrachteten Datenbestand beim Ausführen der Abfrage in das berechnete Intervall zu unterteilen und nur einen Vertreter (für die Messwerte im Intervall) zurück zu liefern. Nutzen Sie diesen Geschwindigkeitschub.

The screenshot shows the Grafana SQL editor interface. It is divided into two main sections, A and B.

Section A: Contains a SQL query:

```
SELECT
  $__timeGroupAlias("time", $__interval),
  avg(gesamtwirkleistung) as "Wirkleistung",
  avg(gesamtblindleistung) as "Blindleistung",
  avg(gesamtscheinleistung) as "Scheinleistung"
FROM "pac4200_10-54-129-91"
WHERE
  $__timeFilter("time")
GROUP BY 1
ORDER BY 1
```

Below the query are several buttons: "Format as" (set to "Time series"), "Query Builder", "Show Help >", and "Generated SQL >".

Section B: Shows a configuration table for the query:

FROM	"pac4200_10-54-129-88"	Time column	"time"	Metric column	⊙	none
SELECT	Column: gesamtblindleistung	Aggregate: avg	Alias: gesamtblindleistung	+		
WHERE	Macro: \$__timeFilter	+				
GROUP BY	time (\$__interval, none)	+				

At the bottom of Section B are buttons: "Format as" (set to "Time series"), "Edit SQL", "Show Help >", and "Generated SQL >".

Abb. 6: Optionen für SQL-Abfragen



Best Practices

An dieser Stelle möchten wir wichtige Tipps für Sie zusammenfassen, damit die Speicherung und Auswertung von Sensordaten auch in Ihrem Unternehmen zum Erfolg wird:

- Identifizieren Sie existierende Datenquellen in Ihrem Unternehmen. Nicht immer ist zusätzliche Sensorik nötig.
- Verwenden Sie Aufzeichnungsintervalle, die genau zu Ihrem Szenario passen. Unnötig kurze Intervalle erzeugen immense Datenmengen.
- Wählen Sie das passende Datenbanksystem für Ihren Anwendungsfall. TimescaleDB ist für Zeitreihendaten eine gute Wahl.
- Dimensionieren und konfigurieren Sie den Datenbankserver so, dass benötigte Tabellenindizes im Arbeitsspeicher gehalten werden können.
- Behalten Sie Ausführungszeiten von Datenbankabfragen im Auge. Können Indizes nicht mehr im RAM gehalten werden, sinkt die Performance drastisch.
- Nutzen Sie optimierte Sonderfunktionen Ihres Datenbanksystems.
- Binden Sie erstellte Visualisierungen in schon vorhandene Dash-

boards und Webanwendungen im Unternehmen ein, um das daraus ableitbare Wissen breit nutzen zu können.

- Behalten Sie den wachsenden Datenbestand im Auge, um beteiligte Systeme frühzeitig skalieren zu können.
- Hinterfragen Sie die angenommenen Parameter zu gegebener Zeit. Gibt es Haupt- und Nebenzeiten mit unterschiedlicher Nutzung der Anlagen? Variable Intervalle für die Datenablage können das Speichervolumen stark reduzieren.
- Erhöhen Sie – wenn möglich – nur die Speicherintervalle, nicht die Abfrageintervalle der Sensoren. So können Sie noch mit der gleichen Geschwindigkeit auf Änderungen reagieren.

Eine solide Datenbasis ist die Grundlage für viele Optimierungen im Unternehmen, wie bspw. genauere Verfügbarkeitsplanung von Ressourcen, verbesserte Prozessplanung und -steuerung sowie jeglicher Einsatz von Anwendungen der Künstlichen Intelligenz. Sie konnten einen Anwendungsfall für den Einsatz von Sensordaten in Ihrem Geschäftsbereich identifizieren und wollen das Gelesene im Selbstversuch umsetzen? Dann stehen wir Ihnen für weiterführende Fragen gerne zur Seite.

Anmerkungen

¹ Bach, F. (2019): Neues Konzept zur skalierbaren, explorativen Analyse großer Zeitreihendaten mit Anwendung auf umfangreiche Stromnetz-Messdaten, Karlsruher Institut für Technologie.

² node-red-contrib-s7. Abgerufen von: <https://flows.nodered.org/node/node-red-contrib-s7> [01.03.2021]

³ node-red-contrib-ads. Abgerufen von: <https://flows.nodered.org/node/node-red-contrib-ads> [01.03.2021]

⁴ How to store time-series data in MongoDB, and why that's a bad idea. Abgerufen von: <https://blog.timescale.com/blog/how-to-store-time-series-data-mongodb-vs-timescaledb-postgresql-a73939734016/> [02.03.2021]

⁵ TimescaleDB vs. PostgreSQL for time-series: 20x higher inserts, 2000x faster deletes, 1.2x-14,000x faster queries. Abgerufen von: <https://blog.timescale.com/blog/timescaledb-vs-6a696248104e/> [02.03.2021]

Autor

Daniel Fischer ist wissenschaftlicher Mitarbeiter an der Professur Fabrikplanung und Intralogistik der Technischen Universität Chemnitz. Im Mittelstand 4.0-Kompetenzzentrum Chemnitz beschäftigt er sich mit den Themen Internet of Things, Datenerfassung, Automatisierungstechnik und Digitalisierung.

daniel.fischer@betrieb-machen.de

Weitere Informationen

Das Mittelstand 4.0-Kompetenzzentrum Chemnitz gehört zu Mittelstand-Digital. Mit Mittelstand-Digital unterstützt das Bundesministerium für Wirtschaft und Energie die Digitalisierung in kleinen und mittleren Unternehmen und dem Handwerk.

Was ist Mittelstand-Digital?

Mittelstand-Digital informiert kleine und mittlere Unternehmen über die Chancen und Herausforderungen der Digitalisierung. Die geförderten Kompetenzzentren helfen mit Expertenwissen, Demonstrationszentren, Best-Practice-Beispielen sowie Netzwerken, die dem Erfahrungsaustausch dienen. Das Bundesministerium für Wirtschaft und Energie (BMWi) ermöglicht die kostenfreie Nutzung aller Angebote von Mittelstand-Digital.

Der DLR Projektträger begleitet im Auftrag des BMWi die Kompetenzzentren fachlich und sorgt für eine bedarfs- und mittelstandsgerechte Umsetzung der Angebote. Das Wissenschaftliche Institut für Infrastruktur und Kommunikationsdienste (WIK) unterstützt mit wissenschaftlicher Begleitung, Vernetzung und Öffentlichkeitsarbeit.

Weitere Informationen finden Sie unter www.mittelstand-digital.de

IMPRESSUM:

Herausgeber:

Mittelstand 4.0-Kompetenzzentrum Chemnitz
Geschäftsstelle
c/o Technische Universität Chemnitz
Prof. Dr.-Ing. habil. Ralph Riedel
DE – 09107 Chemnitz
Tel: 0371 531 19935
Fax: 0371 531 819935
E-Mail: info@betrieb-machen.de
Web: www.betrieb-machen.de
www.kompetenzzentrum-chemnitz.digital

Redaktion & Gestaltung

Daniel Fischer & Diana Falke

Bildnachweis Titel:

© TU Chemnitz

Veröffentlichung:

Juni 2021