

# Wie viel Strom braucht Ihre Software?

Dominik Strub-Tiedt,  
Faysal Cherradi, Pascal Sieber

Der IT-Markt gilt angesichts des Wandels von der Industrie- zur Informationsgesellschaft als einer der dynamischsten Wachstumsmärkte weltweit. Die Globalisierung erhöht stetig den Wettbewerbsdruck, was Unternehmen dazu zwingt, effiziente Prozesse und Kostenstrukturen zu entwickeln. Nur so gelingt es ihnen, schnell genug auf die Marktveränderungen und Kundenbedürfnisse reagieren zu können.

Technologie spielt dabei eine entscheidende Rolle, denn skalierbare Geschäftsprozesse sind durch IT unterstützt. Dies kann einerseits zu Energieeinsparungen in der Abwicklung führen, andererseits steigt aber der Energieverbrauch in der IT enorm an. 2030 werden schätzungsweise 21 Prozent des globalen Stromverbrauchs durch IT konsumiert werden.<sup>1</sup> ClimaTiq geht davon aus, dass die Data Centers 2,5 Prozent bis 3,7 Prozent der globalen CO<sub>2</sub>-Emissionen verursachen. Die kommerzielle Luftfahrt verursacht im Vergleich dazu 2,4 Prozent (vgl. Abbildung 1).

85 Prozent dieser CO<sub>2</sub>-Emissionen können im Design und mit der Entwicklung der Software beeinflusst und damit reduziert werden. Dabei spielen die Architektur und Technologien entscheidende Rollen. Durch eine nachhaltige Gestaltung und Planung der Architektur können ökologische Prinzipien in den gesamten Lebenszyklus von Applikati-

onen und System-Landschaften appliziert werden, was die Einsparungen multipliziert.

## Kleine Komponenten brauchen weniger Strom

Eine effektive Strategie zur Verbesserung der Nachhaltigkeit der Software besteht in der Zerlegung von Softwarekomponenten in kleinere Stücke. Dieser Ansatz bietet den Vorteil, dass kleinere Stücke mit weniger Hardwareanforderungen auskommen können, was Ressourceneinsparungen und eine bessere Skalierbarkeit ermöglicht. Allerdings ist zu beachten, dass dies nicht immer der Fall ist, da einzelne Komponenten manchmal sehr hohe Hardwareanforderungen haben können.

Um den optimalen Schnitt der Lösung zu erreichen und klare Verantwortlichkeiten festzulegen, kann das Konzept des Domain Driven Designs (DDD) angewendet werden.

DDD legt den Fokus darauf, die Domäne und deren Fachlichkeit in den Mittelpunkt der Softwareentwicklung zu stellen. Durch die klare Abgrenzung der Domänen und die Definition wird eine bessere Modularisierung und Abstraktion erreicht.

Durch die Zerlegung der Softwarekomponenten in kleinere Stücke und die Anwendung von DDD können Softwarearchitekten und -entwickler effizientere und wartbarere Lösungen entwickeln. Dies ermöglicht eine bessere Skalierbarkeit, verbesserte Wiederverwendbarkeit von Komponenten und eine klarere Zuweisung von Verantwortlichkeiten, was letztendlich zu einer besseren Gesamtlösung führt.

## Datenspeicher brauchen Strom, auch wenn die Daten nicht mehr gebraucht werden

Eine Software, die kaum technische Schulden aufweist, aber auf Informationsquellen mit hohen Datenschulden angewiesen ist, kann zu einem hohen Ressourcenverbrauch bei der Speicherung sowie beim Datenaustausch führen.

Eine nachhaltige Datenarchitektur beinhaltet eine effiziente Verwaltung von Daten. Diese umfasst die Identifizierung und Löschung nicht mehr benötigter, duplizierter oder veralteter Daten, um Speicherplatz freizugeben. Daten sollten auch entsprechend ihres Wertes und ihrer Aktualität priorisiert und entsprechend gespeichert werden. Durch eine effiziente Datenverwaltung können Speicherressourcen optimiert und unnötiger Energieverbrauch vermieden werden. In der Cloud kann man die Daten beispiels-

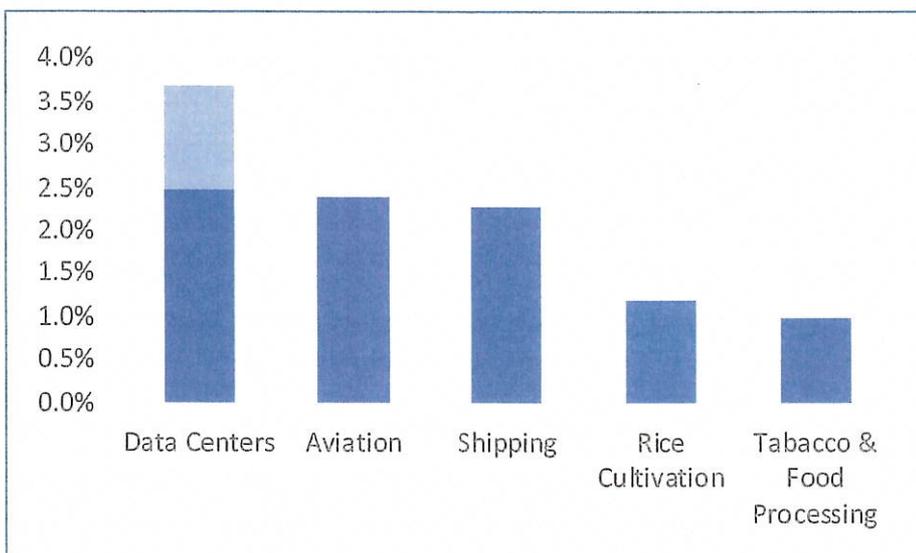


Abbildung 1: Greenhouse Gas Emissions in Data Centres<sup>2</sup>

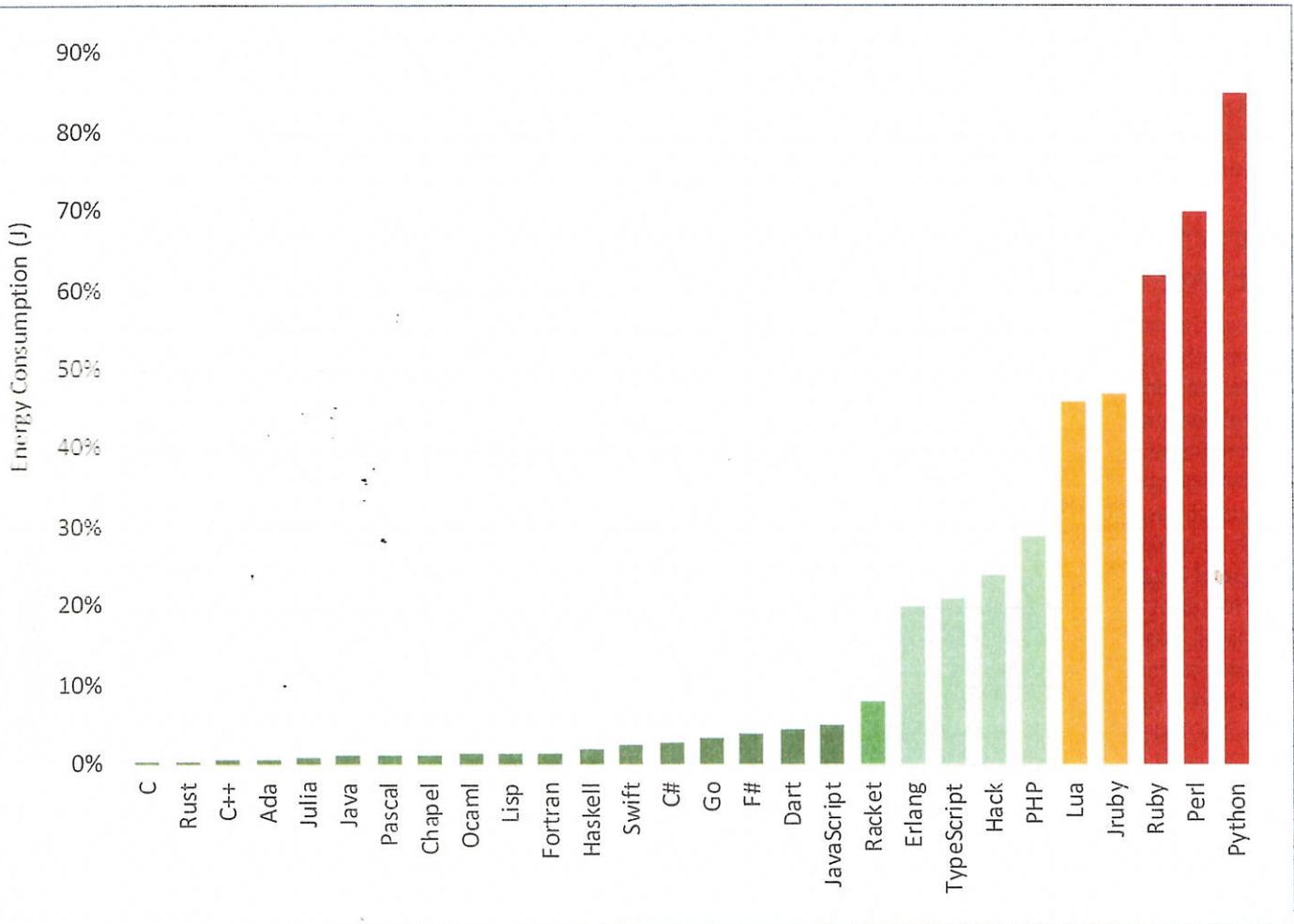


Abbildung 2: Ranking programming languages by energy efficiency.<sup>3</sup>

weise nach heißen, warmen und kalten Daten klassifizieren, um die Daten im jeweils energieeffizientesten Speichermedium abzulagern. Durch die Komprimierung von Daten kann der Speicherbedarf reduziert werden. Dies führt zu einer geringeren Anforderung an physische Speichergeräte und somit zu weniger Energieverbrauch. Insgesamt ist es wichtig, die Datenspeicherung im Einklang mit Nachhaltigkeitsprinzipien zu betrachten. Durch Datenkomprimierung und eine effiziente Datenverwaltung kann der Ressourcenverbrauch reduziert und die Nachhaltigkeit der Datenspeicherung verbessert werden.

### Maschinen-nahe Sprachen brauchen weniger Strom

Die Auswahl der Programmiersprache kann den Energieverbrauch in Software-Systemen beeinflussen. Programmiersprachen, die auf eine effiziente Ausführung und geringe Ressourcennutzung ausgerichtet sind, können zu einem geringeren Energieverbrauch während der Programmausführung beitragen. Zusätzlich können Programmiersprachen,

die Abstraktionen für Hardware-Interaktionen bieten und Funktionen zur Energieverwaltung unterstützen, den Energieverbrauch optimieren. Auch die Verfügbarkeit von Tools zur Energieprofilierung und Bibliotheken für die Energieanalyse und -optimierung spielen eine Rolle. Jedoch darf man natürlich nicht die Anforderungen des Projekts ausser Acht lassen, zum Beispiel, Hardwareentwicklung unterliegt häufig strengen Ressourcenbeschränkungen wie begrenztem Speicher, begrenzter Verarbeitungsleistung und Energieverfügbarkeit. Daher werden in der Hardwareentwicklung häufig Programmiersprachen wie C und C++ bevorzugt, die eine einfache Steuerung und eine effiziente Ressourcennutzung bieten. Diese Sprachen ermöglichen eine direkte Hardwaremanipulation und eine effiziente Codeausführung. Wie auf der unterstehenden Grafik

dargestellt ist, gilt: Je näher die Programmiersprache zur Hardware ist, desto weniger Energieverbrauch verbraucht die damit programmierte Software (vgl. Abbildung 2). Im Allgemeinen beeinflusst die Wahl der Technologie den Ressourcenverbrauch. Es sollte also auf jeden Fall eine Tradeoff-Analyse zwischen Kosten für eine Reduktion und der Reduktion des Energieverbrauchs durchgeführt werden, um die richtige Technologie auszuwählen.

### Mit Cloud-Plattformen können unnötige Infrastrukturen abgeschaltet werden

Moderne Software-Entwicklungsplattformen und Cloud-Services bieten sogenannte

1 Vgl. GFT: Green Coding: <https://www.gft.com/int/en/technology/thought-leadership>.

2 Vgl. Lavi, H. (2021): Measuring greenhouse gas emissions in data centres: the environmental impact of cloud computing: <https://www.climateq.io/blog/measure-greenhouse-gas-emissions-carbon-data-centres-cloud-computing>.

3 Pereira, R. et al. (2021): Ranking programming languages by energy efficiency, Science of Computer Programming, Volume 205, 1 May 2021.

On-Demand-Services an. Damit kann die Ressourcenauslastung optimiert und die Skalierbarkeit verbessert werden. Dies trägt zur Nachhaltigkeit bei, weil Entwicklungs- und Test-Installationen abgeschaltet werden, wenn sie nicht gebraucht werden.

## Das richtige Monitoring hilft Strom zu sparen

Nach der Definition der Prinzipien und dem Entwurf der Software-Architektur muss die Implementierung überwacht werden, um herauszufinden, welche Codeteile mehr Energie verbrauchen und Potenzial für Energieeinsparungen bieten. Um den Source Code hinsichtlich des Energieverbrauchs kontinuierlich zu analysieren, können Automaten eingesetzt werden.

Cast hat einen Index für Green Software entwickelt, um den Energieverbrauch der implementierten Lösung zu messen. Dieser Index von 0 (geringe Umweltfreundlichkeit) bis 100 (hohe Umweltfreundlichkeit) ist ein direkter Durchschnitt der zugrunde liegenden Green-Scan- und Green-Survey-Ergebnisse zur Messung von Pro-

grammierpraktiken und technischen Prinzipien, die Software umweltfreundlicher machen.

Green Survey besteht aus gewichteten Fragen, die einen Zusammenhang mit Energieverbrauch zu tun haben. Zum Beispiel: Unterstützt die Applikation die Fähigkeit der künstlichen Intelligenz? Wie schon bekannt ist über Künstliche Intelligenz, das Training von KI-Modellen benötigt erhebliche Ressourcen. Dies liegt hauptsächlich an der komplexen Natur des maschinellen Lernens, die häufig bei der Entwicklung von KI-Modellen verwendet wird. Deswegen je grösser und komplexer ein Modell ist, desto mehr Rechenleistung wird benötigt.

Beim Green Software Scan wird die Technologiewahl sowie der Source Code bewertet. Im Source Code wird nach definierten Mustern gesucht. Diese Muster sind Algorithmen bzw. Bibliotheken, die potenziell mehr Energie im Vergleich zur anderen alternativen Lösung konsumieren. Beispielsweise verbraucht eine Anwendung, die Datenbank-Abfragen innerhalb eines Loop verwendet, mehr Energie als eine Anwendung, bei der

die Loop-Logik innerhalb der Abfrage programmiert ist.

Die Kunden erhalten einen Überblick von nachhaltigen Best in Class bis zu verbesserungswürdigen Applikationen in ihrer Landschaft. Zudem werden die Codeabschnitte identifiziert und mit konkreten Empfehlungen vorgeschlagen, mit welchem Aufwand man nachhaltiger implementieren kann.

## Fazit

Software hat ein wachsendes Potenzial zur nachhaltigen Entwicklung, da Software immer mehr Bereiche durchdringt. Eine nachhaltige Softwarearchitektur, effizientes Datenmanagement, die Wahl der richtigen Technologie und die Nutzung von On-Demand-Ressourcen tragen zur Nachhaltigkeit der Software bei. Die Planung, Kontrolle und Steuerung der Softwareentwicklung durch ein angemessenes Monitoring ist sinnvoll, um die Ressourcenbelastung und damit den ökologischen Fussabdruck im Verhältnis zu den Kosten und der Verfügbarkeit der Ressourcen zu optimieren. ■

# Vorteile der digitalen Rechnungsverarbeitung

Arcplace 

- + Kürzere Durchlaufzeiten
- + Mehr Prozesstransparenz
- + Höhere Kostentransparenz
- + Bessere Belegkontrolle

Referenzen:

**MPK**  
MIGROS-PENSIONSKASSE

**swisspro**

**Rhätische Bahn**

Erfahren Sie mehr und melden Sie sich unter  
044 501 22 00 oder [sales@arcplace.ch](mailto:sales@arcplace.ch)