



Grüner wird's nicht

Die Programmiersprache, die Sie gerade verwenden, hat eine Energie-Ineffizienz von 3,14 und liegt damit im Mittelfeld.

Zum Ausgleich dieser klimaschädlichen Sprache müssen Sie pro 1000 Zeilen Code einen Baum pflanzen. Seien Sie froh, dass Sie nicht eine Sprache verwenden, die am Ende der Liste steht. Dann müssten Sie aufgrund der Energie-Ineffizienz von fast 80 25-mal so viele Bäume pflanzen.

Übrigens ist die Übereinstimmung des Wertes 3,14 mit den ersten drei Ziffern der Kreiszahl Pi kein Zufall. Sie wissen schon: Bill Gates, 5G, Microbots.

In Zeiten des Klimaschutzes wird die Softwareentwicklung oftmals als Heilsbringer betrachtet. Zum Beispiel sorgt die Optimierung von Transport, Steuerung von Maschinen, Stromerzeugern und -netzen für Einsparung von Energie.

Aber man darf nicht vergessen, dass Software auf Rechnern läuft und diese Strom verbrauchen.

So benötigt das Internet mit all den Servern und Routern eine erkleckliche Menge an Strom. Ganz zu schweigen von den Kryptowährungen, deren Mining den Strombedarf mancher Kleinstadt in den Schatten stellt.

Als Softwareentwickler haben Sie es aber in der Hand, den Stromverbrauch der IT zu minimieren. Beispiel Cache: Daten, die nicht angefordert werden müssen, weil sie schon in einem lokalen Cache liegen, verbrauchen bei der Beschaffung weniger Energie. Bei vielen Daten, die alle paar Sekunden von vielen Clients angefragt werden, kommt da schon eine Menge Energie zusammen.

In einer Studie hat ein portugiesisches Forscherteam schon 2017 ermittelt, welche Programmiersprachen wie viel Strom verbrauchen, wenn sie gewisse Aufgaben ausführen [1]. Hierzu mussten die 27 Sprachen Algorithmen wie binäre Bäume oder Mandelbrot ausführen. Dabei wurden Stromverbrauch, Ausführungszeit und Speicherbedarf gemessen.

Stromsparsieger war die gute alte Sprache C vor Rust und C++. Normiert man den Wert des Strombedarfs von C auf den Wert eins, kam C# auf die oben schon genannten 3,14. Die Schlusslichter bilden Python (75,88) und Perl mit 79,58. Wenn Sie also besonders stromsparend unterwegs sein wollen, dann hat es sich ab sofort ausge-.NET-et. Zurück zum Ursprünglichen, zurück zu C. Aber ob Sie den Preis für die gewonnene Energieeffizienz bezahlen wollen? Immerhin müssen Sie bei C auf Annehmlichkeiten wie einen Garbage Collector verzichten.

Viel Spaß mit der dotnetpro

Tilman Börner
Chefredakteur dotnetpro

[1] Rui Pereira et. al., *Energy Efficiency across Programming Languages*, www.dotnetpro.de/SL2201Edi1



Andreas Maslo

zeigt, dass sich auch in Konsoleprogrammen GUI-Dialogboxen einsetzen lassen (S. 8)



Christian Sender

ist begeistert, was sich mit der Grafikbibliothek SkiaSharp alles umsetzen lässt (S. 58)



Janek Fellien

gibt ein Update, welche Änderung die aktuelle Implementation CQRS 3 mitbringt (S. 70)