Towards Greener Software

Measuring Energy Efficiency of Enterprise Applications on Cloud Infrastructure

Graduate

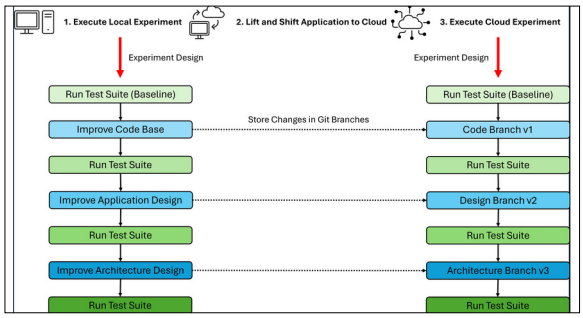
Jan Ruch

Initial Situation: Sustainability is an increasingly important aspect for our society. The energy consumption of software systems is a growing concern, especially in combination with the rise of cloud computing. This thesis sets out to elaborate on a practical approach to measure the energy consumption of Java-based enterprise applications on bare-metal infrastructures, both on-premises and in the cloud. It analyzes energy consumption hotspots, investigates optimization potential, compares their carbon footprint, and derives practical guidelines to improve the energy efficiency of Java enterprise applications and other software systems. ‘LakesideMutual’ is a service-oriented Spring Boot application that imitates a real enterprise application of an insurance company. Its main workflow allows users to request an insurance quote, receive insurance offers, and accept them to sign an insurance contract. The ‘PetClinic’ is a sample application in the Spring community do demonstrate Spring features. It is less complex and solely provides CRUD endpoints to manage pet owners, their pets, veterinarians, and visits.

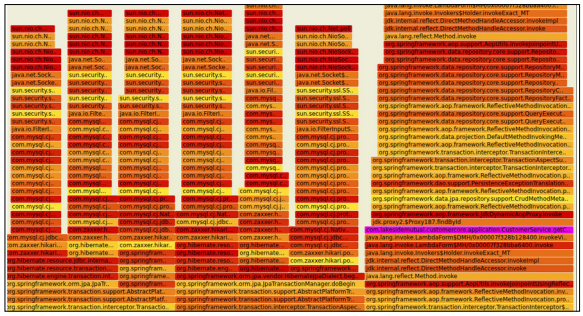
Approach / Technology: We leveraged JoularJX, JMeter, and Flamegraph in our experiments. JoularJX measured the CPU energy consumption of LakesideMutual reported by the RAPL interface in iterative cycles. JMeter simulated users interacting with the main business workflow of the application under test. We then converted the energy consumption for all call stacks reported by JoularJX into Flamegraph visualizations to identify energy consumption hotspots and derive optimizations on code-, design-, and architecture-level. We then repeated the measurements for each optimization to quantify its effects. Eventually, we deployed the application to a Scaleway bare-metal server, re-applied all optimizations, re-iterated all measurements, and calculated the Software Carbon Intensity (SCI) score. The SCI score includes the embodied and operational carbon emissions of the deployment and enables us to compare the carbon footprint of both deployment environments and each optimization.

Result: The result revealed that code-level optimizations led to the most significant energy savings while requiring the least invasive changes compared to design- and architecture-level optimizations. While software optimizations can significantly reduce the energy consumption of an application, the embodied carbon emissions of the underlying infrastructure have a substantial impact on the overall carbon footprint. Exemplary calculations suggest that scaling a small server horizontally is far more carbon-efficient than running one large vertically scaled server. We elaborated six practical guidelines, which developers can apply in real-world projects. The first guideline establishes an effective measurement approach leveraging the presented tools. Four guidelines focus on optimizing the energy efficiency of Java-based enterprise applications. The last one covers the deployment to a more sustainable cloud infrastructure. We validated these guidelines against the PetClinic application, which confirmed their applicability and effectiveness in reducing energy consumption and carbon footprint. Two selected guidelines are contributed back to the Green Software Patterns repository to share them with the community. Future work could contribute additional guidelines, explore other measurement tools for different technology stacks, or expand the measurement approach to other system components beyond the CPU.

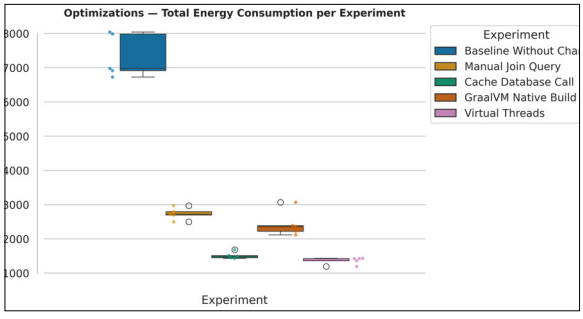
We measure the application locally, store all optimization changes in Git, and re-iterate the measurements on the cloud.
Own presentment



The Flamegraph illustrates the energy consumption, six flames for the transaction management and one for the DB fetch.
Own presentment



The diagram shows the guideline validation results. It illustrates the energy consumption reduction for the PetClinic.
Own presentment



Advisor
Prof. Mirko Stocker

Co-Examiner
Büttiker Leo, yonesu GmbH, Olten, SO

Subject Area
Computer Science, Software and Systems, Energy and Environment