Editor: **Eoin Woods**
Endava
eoin.woods@endava.com

Architectural Principles for Energy-Aware Internet-Scale Applications

Rabih Bashroush and Eoin Woods

DIGITAL-TRANSFORMATION INITIATIVES HAVE led to major efficiencies and cost savings, including the transition from paper-based processing to electronic documents and the use of traffic-routing algorithms for vehicle navigation. However, the software performing this "magic" consumes nearly 10 percent of the world's electricity.[1] Today's cloud-based applications span multiple continents, consuming energy in servers, networks, cooling and power facilities, storage, and user devices.

Here, we present three simple design principles software architects can use to address system-level energy efficiency. A case study illustrates the energy savings possible with a holistic approach.

From Hardware to Software

Over the past decade, researchers have been studying IT infrastructure energy consumption, working to increase datacenter, network, and hardware efficiency. Datacenter energy efficiency has improved considerably. For example, in the US, public-sector datacenters are now expected to operate at a *power usage effectiveness* (PUE) of less than 1.5, whereas a PUE of 2 was considered normal only a few years ago.

PUE is a datacenter's total energy consumption divided by its IT energy consumption, usually measured over one year. A PUE of 1.5 indicates that for every 1 KWh of IT load, a datacenter requires an additional 0.5 KWh.

Hardware has experienced a similar trend; computations per joule of energy have doubled every 1.57 years over the past two decades.[2] Yet, limited progress has been made in addressing the entire software system's energy efficiency. However, software engineering research is now focusing on a system-wide approach.

The State of the Art

To increase efficiency, we must be able to measure it. That is, we must be able to measure the useful work our software applications produce and the amount of energy this takes—and then optimize the ratio between the two. However, although the datacenter world has metrics such as PUE, no comparable metrics exist for software.

A further complication is that modern applications run across multiple platforms (user devices, networks, computers, storage, and so on). Optimizing energy consumption across all these platforms will require a range of special-

0740-7459/17/$33.00 © 2017 IEEE

ists to collaborate across traditional design boundaries.

Optimization must also consider key quality properties such as resilience (because redundancy in system designs is usually a major contributor to energy consumption), usability, and performance. In reality, however, we have no design trade-off tools that let us conduct such analyses.[3]

Despite these challenges, energy efficiency has been gaining traction in software engineering. Much of the early research focused on measuring applications' energy consumption[4] and tried to define useful work so as to allow the creation of useful metrics (for example, the DC4Cities project; www.dc4cities.eu). In parallel, other researchers have explored compiler optimization to decrease energy consumption or have evaluated design patterns' energy efficiency.

All these efforts have helped us begin to understand and optimize software applications. However, improving today's Internet-scale systems will require a more radical approach that considers the whole system. Such an approach is inherent to software architecture work.

The Three Principles

On the basis of early experiences and research in the field, we propose three simple architecture principles for achieving energy-efficient systems:

- Principle 1. Energy efficiency metrics must relate business transactions to energy consumption in a meaningful way to key system stakeholders.
- Principle 2. Identifying sources of energy waste at the system level produces the biggest savings.
- Principle 3. Addressing the energy optimization problem requires a cross-disciplinary team.

We now examine these principles in more detail.

Relating Business Transactions to Energy Consumption

Energy efficiency must be measured in a way that system stakeholders can understand. Ensuring that the metrics are meaningful is necessary to convince senior management to sponsor optimization projects. Ultimately, suitable metrics can help achieve holistic system tuning and drive revenue and cost optimization.

> The software performing technological "magic" consumes nearly 10% of the world's electricity.

Identifying Sources of Waste

Focus effort where it will be the most effective. For example, redundancy is a commonly overlooked source of energy consumption. To support resilience, redundancy is usually applied at all levels, including facilities, hardware, and software. Without system-level evaluation of resilience requirements, redundancy might be applied too generously. So, matching redundancy to actual requirements is a huge opportunity to achieve energy savings that would be difficult with local optimizations.

Employing Cross-Disciplinary Teams

Energy optimization requires design work across traditional design boundaries. For example, optimizing the design of resilience requires collaboration among infrastructure engineering, application development, and business teams. Without such collective efforts, improvements will be restricted to local optimizations—which often miss the bigger opportunities for savings.

A Case Study: eBay

The online-auction company eBay used principles such as those we've outlined to achieve significant energy savings.

As part of eBay's commitment to reducing its environmental footprint while decreasing costs and increasing performance, it introduced the Digital Service Efficiency (DSE) initiative.[5] DSE relates business metrics such as customer buying and selling transactions to their energy consumption and environmental impact. DSE provides a set of easily understood metrics to help eBay understand, communicate, balance, and tune its energy consumption (principle 1). These metrics include buy transactions/kWh, sell transactions/kWh, revenue/MW; and CO2 emissions/million users.

eBay identified reducing infrastructure redundancy as one of the main opportunities to save energy. To explore this opportunity, eBay rethought its entire system architecture (principle 2), taking into consideration its business needs and re-

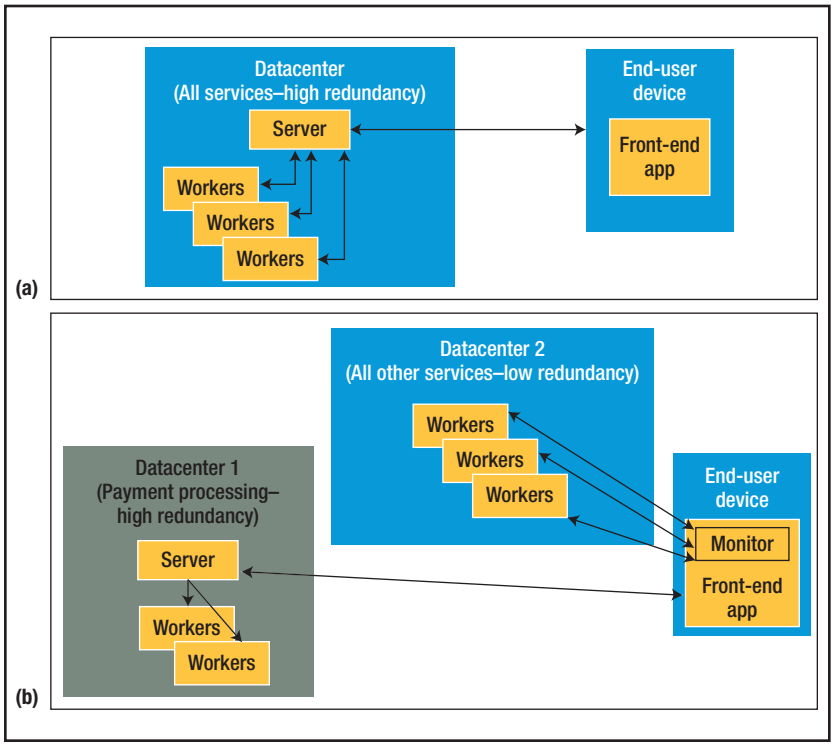

FIGURE 1. eBay's (a) original system architecture and (b) new design. The new design allows for reduced datacenter redundancy while maintaining overall system performance and resilience.

dundancy costs. eBay realized that no matter how resilient its back-end system was, if the end-user system failed, the whole session failed. This meant that responsibility for system resilience could be moved to the weakest link: the end-user system.

So, the company introduced a new system architecture that includes a monitor in the end-user system. The monitor identifies when a transaction such as a search request exceeds a timeout, perhaps owing to a service failure. When this occurs, the monitor transparently reissues the request to another service location, without notifying the user. Thus, noncritical services can be less resilient because the monitor's operation can mask their failures.

Furthermore, on the basis of business analytics, eBay estimated that only about 10 percent of its transactions were critical, such as payments. Payment handling has specific regulatory needs and requires a highly available infrastructure. On the basis of this insight, eBay processes payments at a specific, highly resilient datacenter that's a fraction of the capacity of its original datacenter[6] while processing the remaining workload in a cheaper, less resilient infrastructure. This significant improvement in energy efficiency required cooperation across eBay's engineering, operational, and business teams (principle 3).

Figure 1 depicts eBay's original and resulting architectures.

eBay has achieved major capital and operational expenditure savings by adopting this new architecture. The new low-redundancy site's simpler requirements have substantially decreased the infrastructure, which in turn has significantly decreased datacenter build-out and fit-out costs and time scales. Redundancy costs are significant. For example, according to Steven Shapiro, the cost of building a Tier III datacenter is double that of a Tier II datacenter.[7] (The Tier Classification System is a widely used rating system for datacenter availability, with Tier I being the least available or redundant facility and Tier IV the highest.[8])

Even more important (from our perspective), eBay has reduced energy consumption by approximately 50 percent because the low-redundancy site requires fewer infrastructure components (for example, $N + 1$ rather than $2N + 1$ redundancy).[5] This has resulted in not only significant energy cost savings but also reduced maintenance and hardware refresh requirements, further lowering environmental costs.

There has been increased interest in reducing the significant energy costs of running large IT systems. However, software architects lack suitable tools and methods to address energy concerns when designing systems. With this challenge in mind, we've suggested our three principles, which architects can follow to make energy-related tradeoffs during system design even with today's limited knowledge and technology.

Despite these principles' simplicity, eBay's experience shows that they can yield significant cost and energy savings when applied to large-scale production systems. Savings of this scale are difficult to

achieve through local optimizations, so we must rely on software architects' skills to lead our efforts in this emerging area.

References

1. M.P. Mills, *The Cloud Begins with Coal: Big Data, Big Networks, Big Infrastructure, and Big Power*, tech. report, Digital Power Group, Aug. 2013.
2. J. Koomey et al., "Implications of Historical Trends in the Electrical Efficiency of Computing," *IEEE Annals of the History of Computing*, vol. 33, no. 3, 2011, pp. 46–54.
3. R. Bashroush, E. Woods, and A. Noureddine, "Data Center Energy Demand: What Got Us Here Won't Get Us There," *IEEE Software*, vol. 33, no. 2, 2016, pp. 18–21.
4. S. Islam, A. Noureddine, and R. Bashroush, "Measuring Energy Footprint of Software Features," *Proc. IEEE 24th Int'l Conf. Program Comprehension* (ICPC 16), 2016; doi.org/10.1109/ICPC.2016.7503726.
5. "Digital Service Efficiency," eBay, Mar. 2013; www.ebaytechblog.com/wp-content/uploads/2013/03/FINAL_DSE-Solution-Paper.pdf.
6. D. Nelson and R. Paquet, "How eBay's I&O Organization Is Supporting Business Initiatives," keynote interview at Gartner Data Center Conf. 2013, 2013.
7. S. Shapiro, "Myths and Realities about Designing High-Availability Data Centers," presentation at Data Centre World 2015, Sept. 2015; www.slideshare.net/Morrison Hershfield/myths-and-realities-about-designing-high-availability-data-centers.
8. "Tier Classification System," Uptime Inst., 2013; uptimeinstitute.com/tiers.

RABIH BASHROUSH is a faculty member and director of the Enterprise Computing Research Group at the University of East London. Contact him at r.bashroush@uel.ac.uk.

EOIN WOODS is the chief technology officer at Endava. Contact him at eoin.woods@endava.com.