Sustainable Programming: Balancing Best Performance and Lowest Power Consumption

Applies to:

Performance, Scalability, Green IT, Programming, Sustainability, Performance-optimized coding

Summary

For software developers, being sustainable and contributing to Green IT means designing software programs that make efficient use of computing resources. For SAP programmers, this becomes even more imperative if we take the huge number of all business transactions worldwide into consideration that is handled by an SAP system in one way or another. The goals for a high performance are to ensure that software is scalable, and that it achieves stable end user response times and the expected throughput. Adhering to these goals helps when programming for a Green IT. But the equation "performance-optimized coding = power-optimized coding" is not always true – sometimes there is a tradeoff between power and performance. Looking more closely at specific performance requirements we can see different degrees of interdependence between high-performance program code and power consumption.

Author(s): Dr. Heiko Gerwens, SAP AG, Germany

Detlef Thoms, SAP AG, Germany

Company: SAP AG

Created on: 2 May 2011

Author Bio

Heiko Gerwens is the leading performance expert for SAP financial service applications. He joined SAP in 1997 and has worked in the Performance & Scalability team. During this time he has been involved in the development of many applications including customer projects that implement these applications. He supports customers, partners, consulting agents, and the field organization that deals with all aspects of performance, scalability, and sizing.

Detlef Thoms has twelve years of SAP experience that he has gained in Development Support, as SCM Senior Solution Consultant in numerous SAP implementation projects and in the SAP NetWeaver RIG as RIG Expert for the Master Data Management portfolio. In January 2011 he joined the Product Management as Product Expert for Performance & Scalability across the SAP portfolio.

Table of Contents

Overview

One of the greatest challenges relating to global warming is that greenhouse gases result—directly or indirectly—from almost every major human industry and activity. This chart shows these industries and activities, and the type and volume of greenhouse gases that result from them. It includes emissions estimates from a range of international data providers, in an attempt to account for all significant GHG emissions sources.

In 2005, total GHGs are estimated at 44,153 MtCO2 equivalents (million metric tons). CO2 equivalents are based on 100-year global warming potential (GWP) estimates produced by the IPCC. 2005 is the most recent year for which comprehensive emissions data are available for every major gas and sector*. [*World Greenhouse Gas Emissions in 2005; http://www.wri.org/chart/world-greenhouse-gas-emissions-2005]

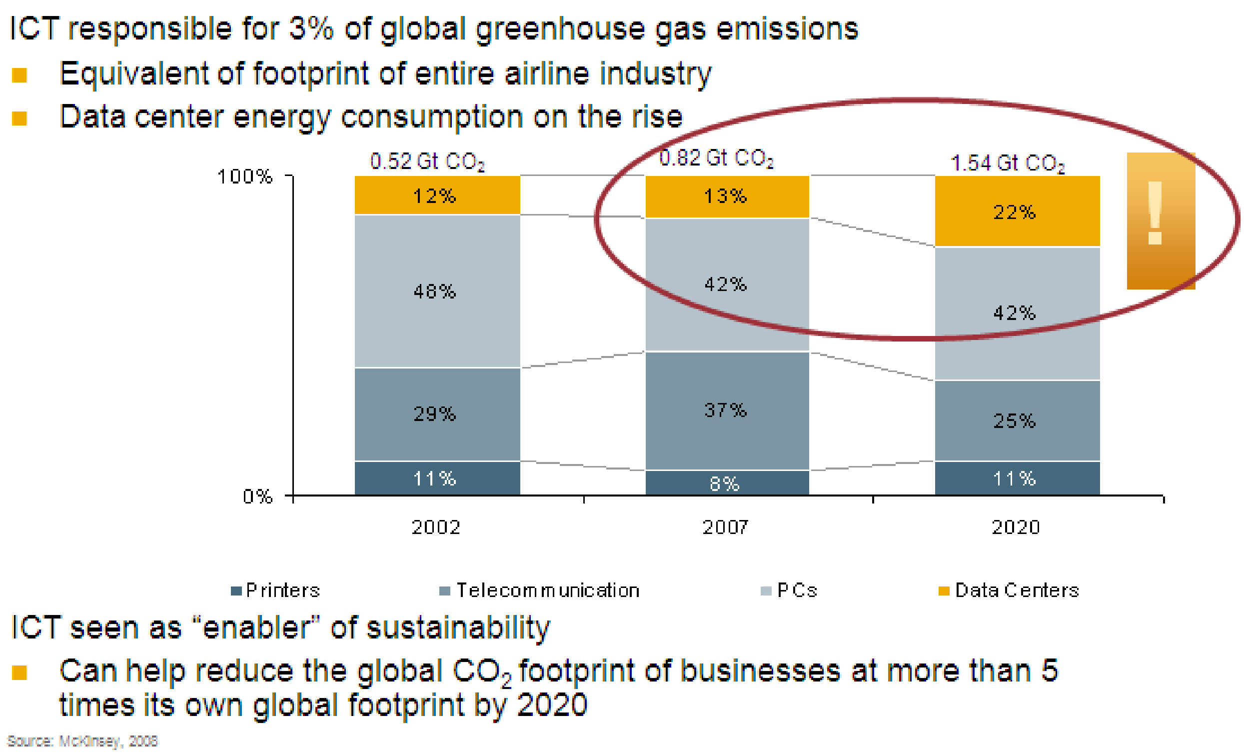
Energy use is the single largest contributor to the carbon footprint of the ICT sector. Still, greenhouse gas emissions from the ICT sector are small relative to the sector's share of the world economy. Despite tremendous efficiency improvements in electronic components, demands for new services are increasing, and so is the amount of total electricity consumed by ICT. ICT devices can have a negative impact on the environment, however the ICT sector can also be innovative and help curb emissions too. Data centers (also called server farms) are where internet service providers or e-commerce companies locate the hundreds or thousands of computer servers that provide their online services. Data centers use massive amounts of electricity; large ones can use megawatts of power, with each square meter using as much power as an entire average US home. Cooling is about 60 per cent of the power costs in a data centre because of inefficiency. The IT industry has realized the need for action which at the same time is of course a business opportunity for many*. [*United Nations Environment Management Group (EMG) [http://www.unep.org/climateneutral/Topics/Informationandcommunicationtechnologies/tabid/147/Default.aspx]

The Sustainability Rationale

For software developers, being sustainable and contributing to Green IT means designing software programs that make efficient use of computing resources. For SAP programmers, this becomes even more imperative if we take the huge number of all business transactions worldwide into consideration that is handled by a SAP system in one way or another.

Today, information & communication technology (ICT) is directly responsible for 3% of the global greenhouse gas emissions. This is equivalent to the footprint of the global airline industry. If business continues as usual, ICT energy consumption will increase by 30% to 1,54 GT CO_2 until 2020, and data center energy consumption will rise to be responsible for 22% of this growth (McKinsey, 2008).

On the other hand, IT can help reduce CO_2 emissions enormously: If the most efficient ICT technologies are adopted, IT can help reduce the global CO_2 footprint of businesses at more than 5 times its own global footprint by 2020. Developers can contribute to this effort by designing software such that the energy required by a system is reduced.



Power Consumers in an SAP System

In an SAP system, the main power consumers are CPU, memory and disks:

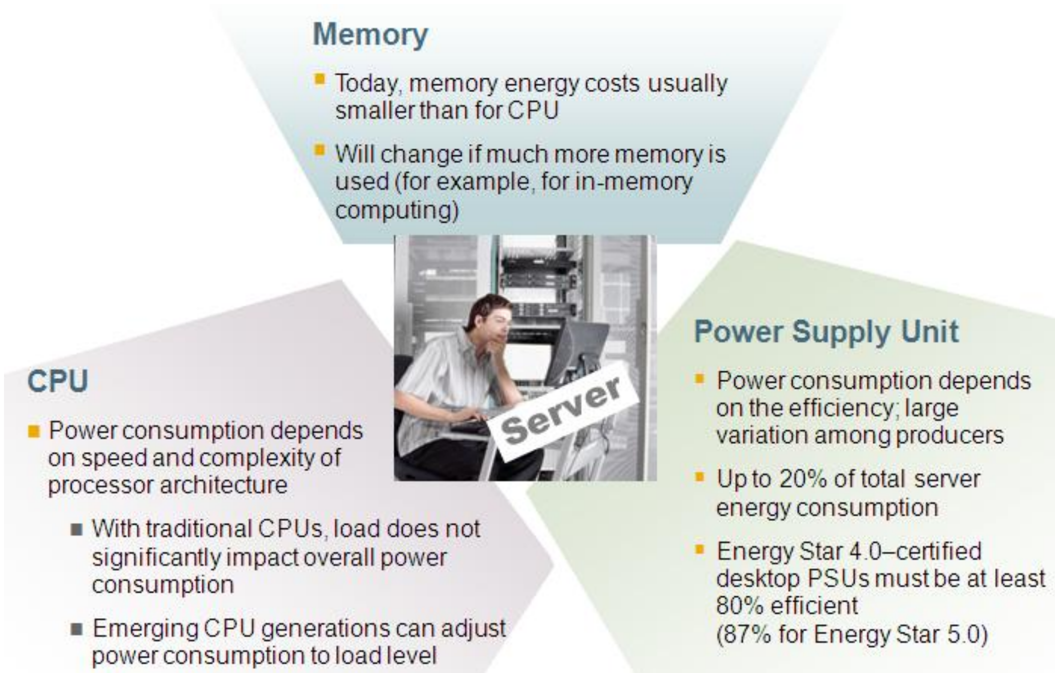
- **CPU:** This is the primary power consumer. The degree of power consumption depends largely on the CPU speed and the complexity of the processor architecture. Older CPUs have no means of adjusting their speed to the requirements of the system, and thus higher or lower load levels do not significantly impact overall power consumption. However, new CPU generations can adjust the clock rate, and thus the power consumption, to current system requirements.

- **Memory:** Next to CPU memory is a main power consumer. With new technology trends, e.g. for in-memory computing, more memory will be required. So power consumption of memory will become even more important.

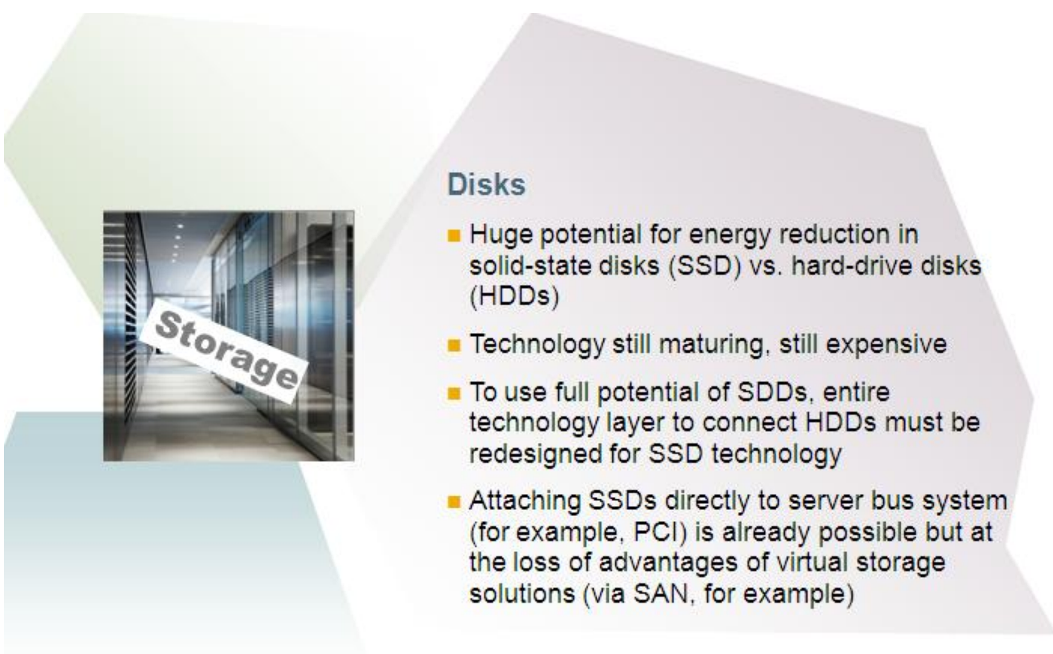
- **Disks:** There is huge potential for energy reduction in solid state disks (SSD) as compared to hard-drive disks (HDDs). SSDs have no moving parts and thus are much more power efficient. Moreover flash-integrated circuits have a much higher IO throughput per SSD device, so fewer HDD devices are needed

  For a comparison of SSD with hard disk drives please also see: http://en.wikipedia.org/wiki/Solid-state_drive

Power Consumers in an SAP System – Server



Power Consumers in an SAP System – Storage



Typical values for power consumption of technical components:

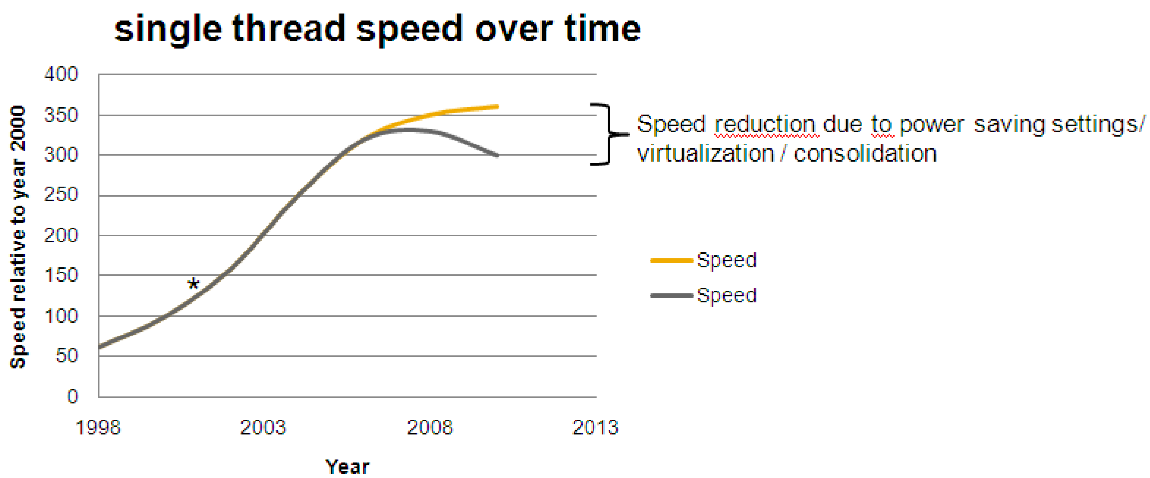
- CPU per core (~1000 SAPS) → 20 Watt
- Memory per Gbyte → 1 Watt
- Storage per 1000 IOPS (for 3000 SAPS)→ 100 Watt

- Typical landscape for the Suite:
- 2 socket 8 core → 200 Watt
- 100 Gbyte → 100 Watt
- Storage 4 Tbyte, 3000 IOPS → 800 Watt

Impact of single thread performance

When we analyze the impact of single thread performance it is obvious that “Free lunch is over!”. The reason is that power saving in servers is most efficiently done by reducing speed of processor (reduce P-states). This slows down the single thread performance.

In addition there is a trend to consolidate infrastructure on fewer servers to reduce costs. This typically goes in line with higher average utilization which again has a slight negative impact on response times. On top of this in many situations virtualization is used for consolidation which again creates some performance overhead. So all in all the current technology trends increases the pressure on development towards high performing software.



***** Start of multi-core architecture in 2001: A multi-core processor combines two or more single-core processors on one chip, allowing tasks to be run unilaterally while increasing the overall performance without requiring excessive energy or generating too much heat.

In transactional processing parallelization is very hard to impossible to achieve. Therefore adding new functionalities and capabilities has to be done with care!

- Which capabilities/ functionality are “must” and which are nice to have
- With which software architecture

Due to this the pressure for small CPU consumption is increased.

Takeaways:

- Compute resources are not endless!
- Compute resources are not for free!

Performance-Optimized Code and Power Efficiency

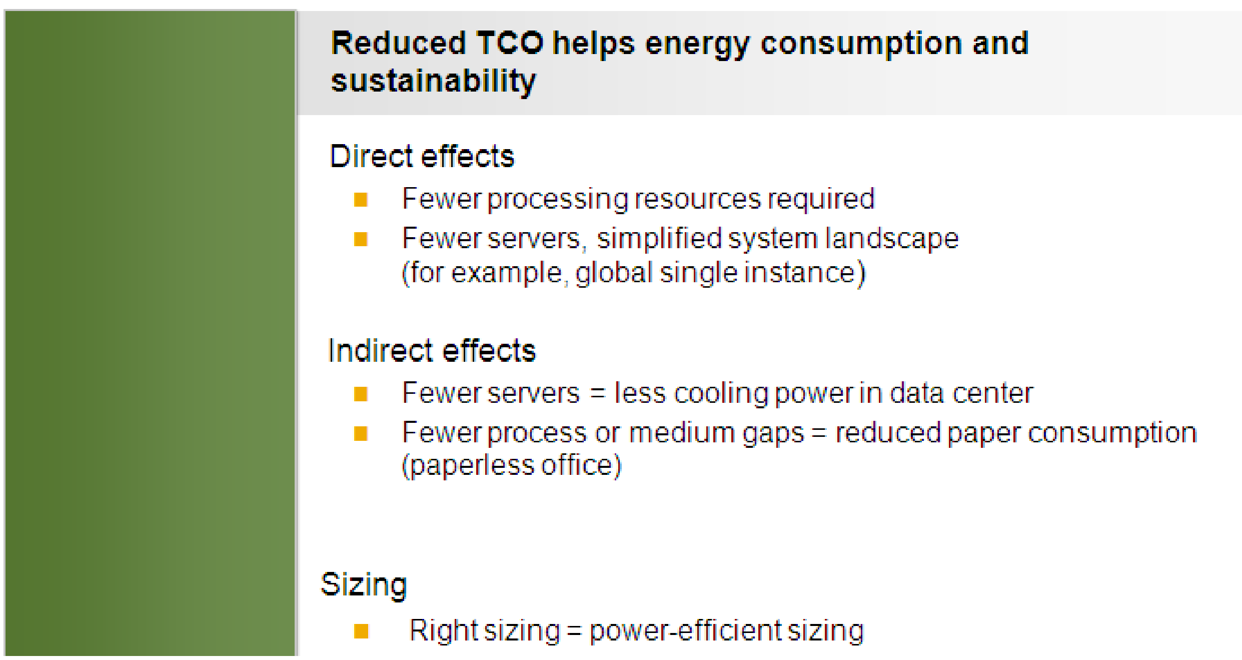
The goals for a high performance are to ensure that software is scalable, and that it achieves stable end user response times and the expected throughput**.**

Adhering to these goals helps when programming for a Green IT:
Performance-optimized programming often (but not always as we shall see below) equates to energy efficient programming.

General effects of performance optimizations on power consumption

Performance optimizations reduce power consumption in SAP system landscapes if they lead to:

1. **Fewer processing cycles** (CPU optimization)
2. **Less disk I/O** (indexes, buffers and caches)
3. **Optimized use of memory** (shared caches)
4. **Reduction of TCO and hardware sizing**
   If a performance optimization reduces the required CPU time for a very frequently used transaction say, from 2 to 1 CPU seconds, this significantly reduces the overall number of CPUs required in all application servers.
   Scalable software enables multi-client capabilities, which, in turn, helps reduce the number of servers running in an IT environment. For example, consolidating IT systems from various regional data centers onto a single global instance and data center brings immediate financial and environmental gains - through reduced cost for power consumption and maintenance, as well as an optimum 24x7 usage of data center resources. Consider also the knock-on effects of server reduction on the cost and resources spent on cooling the data center.



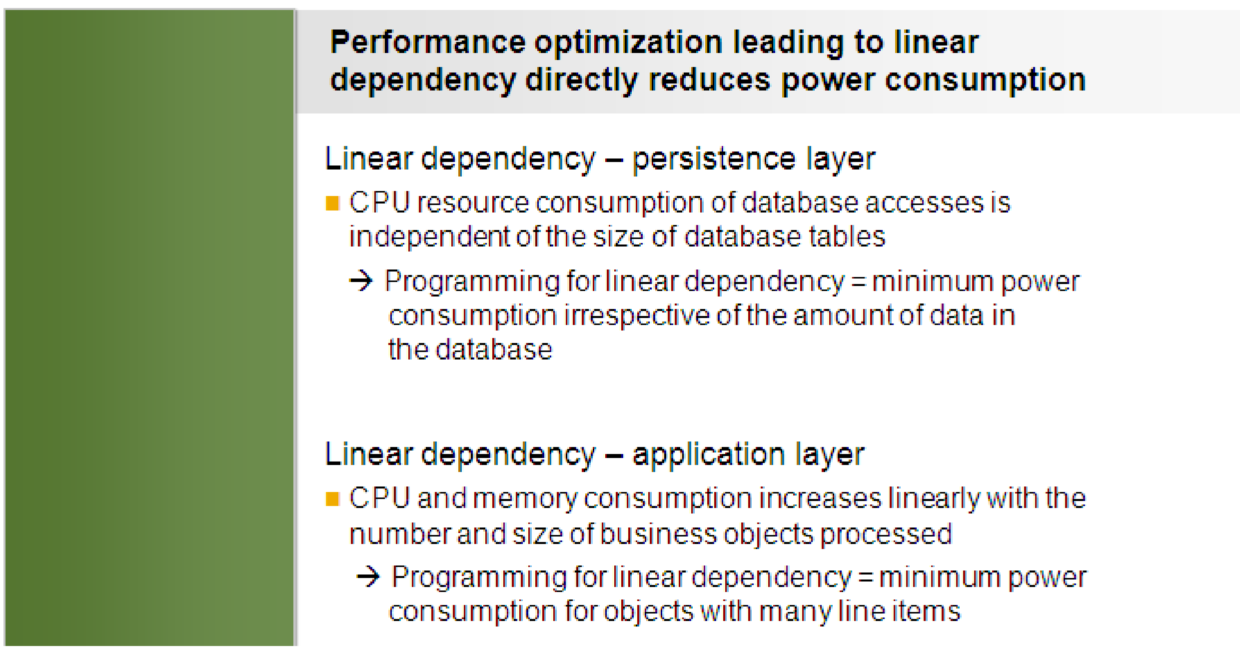
Specific effects of performance optimizations on power consumption

The equation “performance-optimized coding = power-optimized coding” is not always true – sometimes there is a tradeoff between power and performance. Looking more closely at specific performance requirements we can see **different degrees** of interdependence between high-performance program code and power consumption.

1. Performance optimization leading to linear dependency directly reduces power consumption.

   Any performance optimization that changes a program‘s resource consumption behavior from worse than linear to linear **immediately and directly** reduces power consumption. Non-linear runtimes and large amounts of data cause a huge increase in runtimes, CPU cycles and memory. This means significantly more or larger servers. Note the following:

   - On the application layer, CPU and memory consumption should, at maximum, increase linearly with the number and size of business objects processed.
     What you can do: To achieve this, avoid memory leaks and frequent scans on large internal tables as they cause non-linear behavior.
   - On the Persistence layer, resource consumption of the CPU should be independent of the size of database tables.
     What you can do: To achieve this, always use appropriate database indexes and complete where clauses when programming OLTP applications.



2. In a scalable architecture, efficient client-server programming reduces power consumption **in most situations**

   In a client-server architecture, central resources such as the database must be protected. The sustainability effect in reducing the number of roundtrips between database and application servers is based on the fact that the SQL protocol is quite expensive. By reducing the number and type of accesses to the database the SQL protocol cost can be minimized. Client server programming enables scalability which, as mentioned before, permits server consolidation.

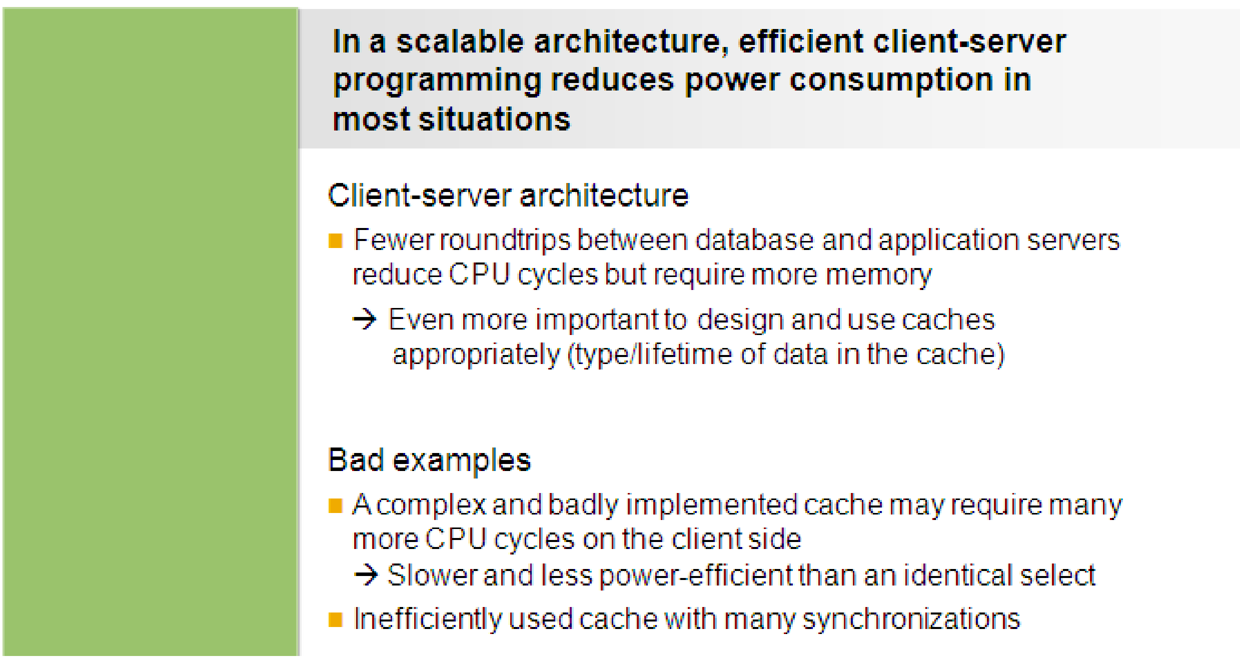
   Fewer roundtrips between the database and application servers reduce CPU cycles but require more memory.

If for example a cache for already selected data is implemented one has to make sure that this cache cannot grow endless but data also has to be removed from this cache with some algorithm. E.g., in a transactional model a typical cleanup point would be after finalizing a transaction, because it's unlikely the same data (e.g. some master data) will be used in the next transaction.

What you can do: Use buffers and caches appropriately:

- Use the buffer/cache infrastructure of the application server for defined sets of data that rarely change (e.g. configuration data, runtime information)
- Within a transaction, use temporary buffers or caches for larger sets of data that are likely to change more frequently (e.g. master data, business partner, material master)

However, be aware: A complex and badly implemented cache may require many more CPU cycles on the client side and therefore is slower and less power-efficient than an identical select. If for example the application implements a cache which will be scanned sequentially with every access these accesses might become very slow and therefore power consuming in case the cache grows large.



3. Parallel processing **can** lead to additional power consumption

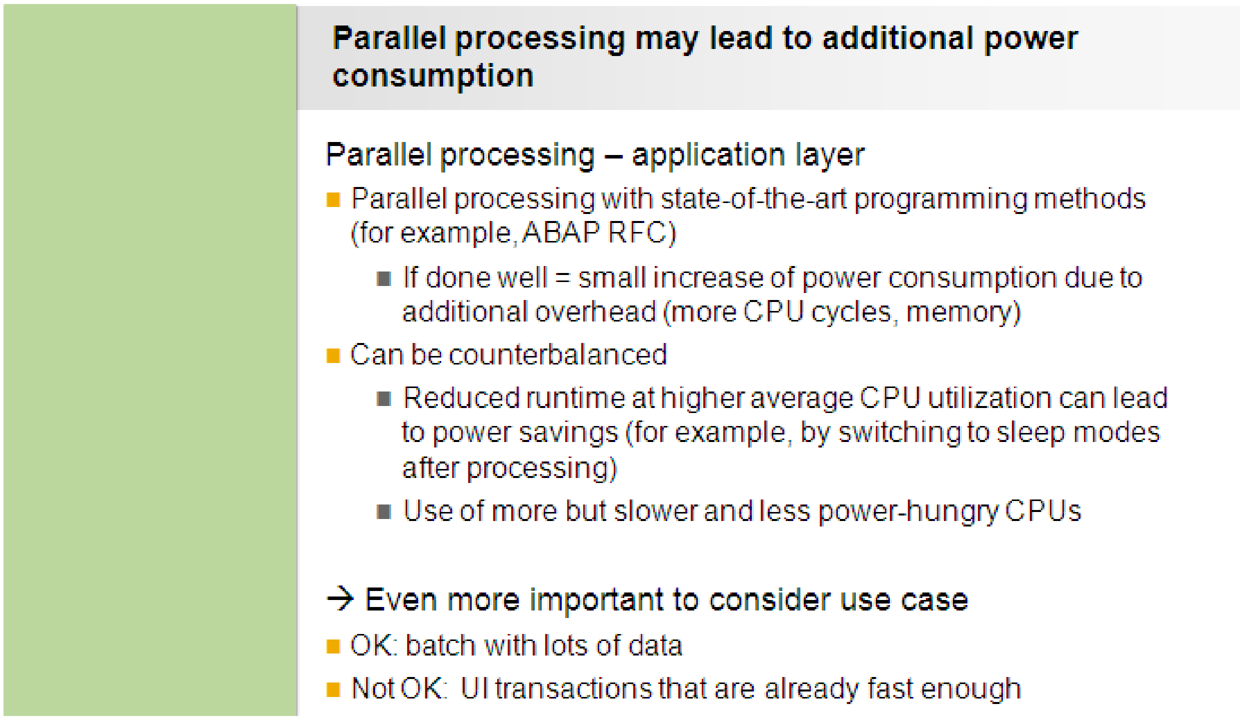
   Generally, if parallel processing is used, the performance – in the sense of response time or runtime – is improved. However, parallel processing must be done well to avoid too much overhead and thus a knock-on effect on sustainability: Each parallelization "costs" CPU cycles and memory, causing increased power consumption for processing the same business functionality.

   If done well, parallel processing with state-of-the-art programming methods creates a small increase of power consumption due to additional overhead on the application layer. This increase can be counterbalanced, by the power saved e.g. through the reduced runtime at higher average CPU utilization (e.g. by switching to sleep modes after processing), or the use of more, but slower and less power-hungry, CPUs.

   What you can do: It is even more important for you to consider the use case carefully: use parallelization, for example, for batch with lots of data, but not for UI transactions that are already fast enough.

This means you have to evaluate if parallel processing is necessary to reach your KPI versus the overhead for parallel processing.

- In terms of power consumption parallel processing is the second best choice
- Best choice is to optimize architecture and code so that it easily fulfills response without parallel processing



One example for Performance-Optimized Code and Power Efficiency

Adhering the above mentioned goals for a high performance are to ensure that software is scalable, and that it achieves stable end user response times and the expected throughput some remarkable results can be achieved. Let’s have a look on the following example.

Baseline:

SAPS: SAP Application Performance Standard (SAPS) is a hardware-independent unit of measurement that describes the performance of a system configuration in the SAP environment. It is derived from the Sales and Distribution (SD) benchmark, where 100 SAPS is defined as 2,000 fully business processed order line items per hour.

1000 SAPS = 1kSAPs is defined as 20,000 fully business processed order line items per hour.

SAP Server Power Benchmark: Power Efficiency Indicator - Server (watts/kilo SAPS): **18,3 watts/kilo SAPS**

Energy density (From Wikipedia, the free encyclopedia): Energy density is a term used for the amount of energy stored in a given system or region of space per unit volume. Often only the useful or extractable energy is quantified, which is to say that chemically inaccessible energy such as rest mass energy is ignored. Quantified energy is energy that has some sort of, as the name suggests, quantified magnitude with related units. **Gasoline : 46.4 (MJ/kg)**

The energy in kWh / l depends on the respective density and the lower heating value of the fuel. If one takes typical characteristics for each fuel grades the results are for about **8.6 kWh / l.** (Number taken from a gasoline producer)

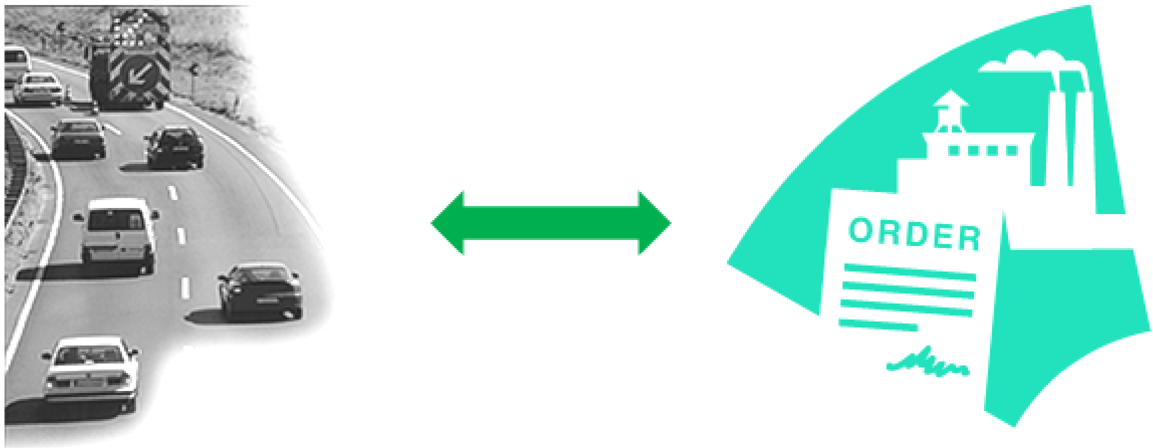
Typical midsize car with a patrol consumption of **6l petrol /100km**.

Per 100km 6l x 8.6 kWh / l = **51,6 kWh** are consumed. If you compare this to the SD Benchmark you can fully business process [(51,6 *1000/18,3) *20000=] 56,39 Million **order line items**

This means if you drive in average per year 30.000 km it is equivalent to the energy consumption of processing [(300*6l x 8.6 kWh/l)*1000/18,3) *20000=] **16.918.032.786 order line items.**

1 l petrol results in a production of around 2,32 kg CO_2 means this is equivalent to 1800x 2,32 = 4176 kg CO_2.

So summing it up 30,000 yearly car kilometers compare to a quite impressive long number - 16.918.032.786 SAP SD postings.



30,000 yearly car kilometers

16.918.032.786 SAP SD postings

It is interesting to see how energy efficient SAP systems can be operated.

Please let us know what you think from a software developer's perspective, being sustainable and contributing to Green IT by designing software programs that make efficient use of computing resources?

How important is this topic within your company?

Your insights will help us immensely.

Please give us feedback via the corresponding SDN blogs or via mail:

heiko.gerwens@sap.com

detlef.thoms@sap.com

Related Content

Related SDN site:

https://www.sdn.sap.com/irj/sdn/performance-analysis

Related Websites:

http://service.sap.com/performance

http://service.sap.com/sizing

http://service.sap.com/benchmark

www.sap.com/benchmark

Copyright