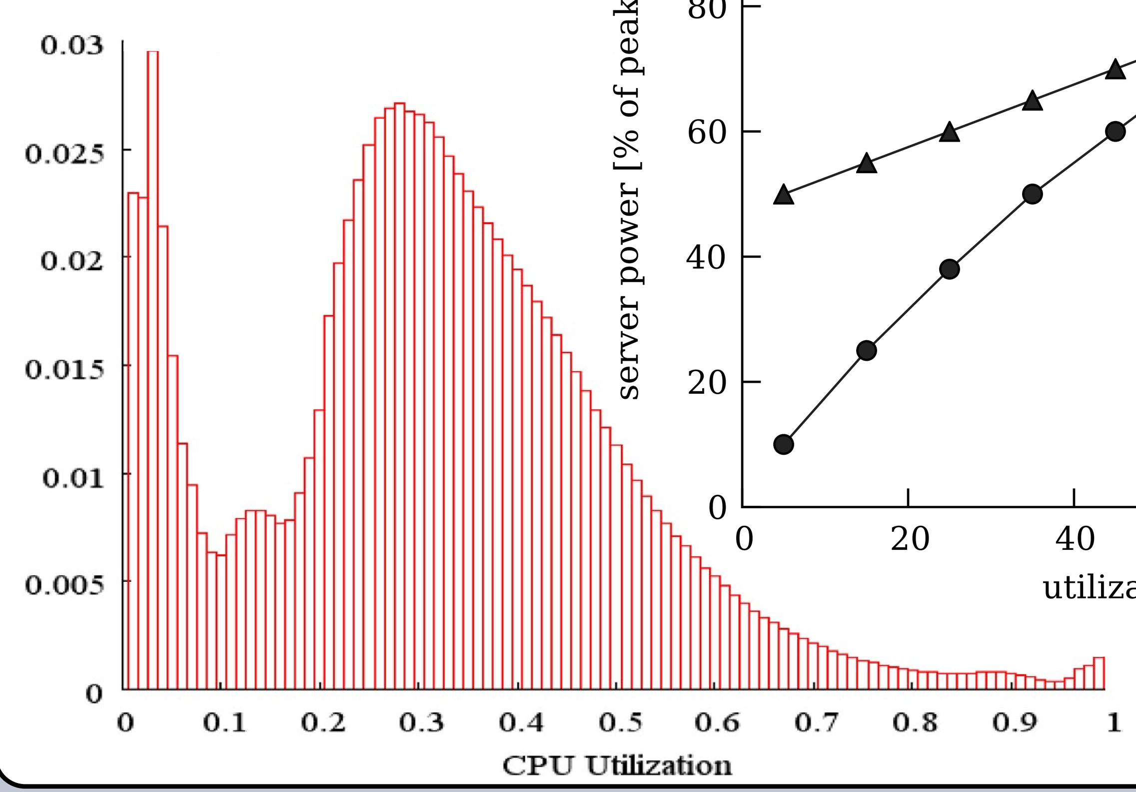
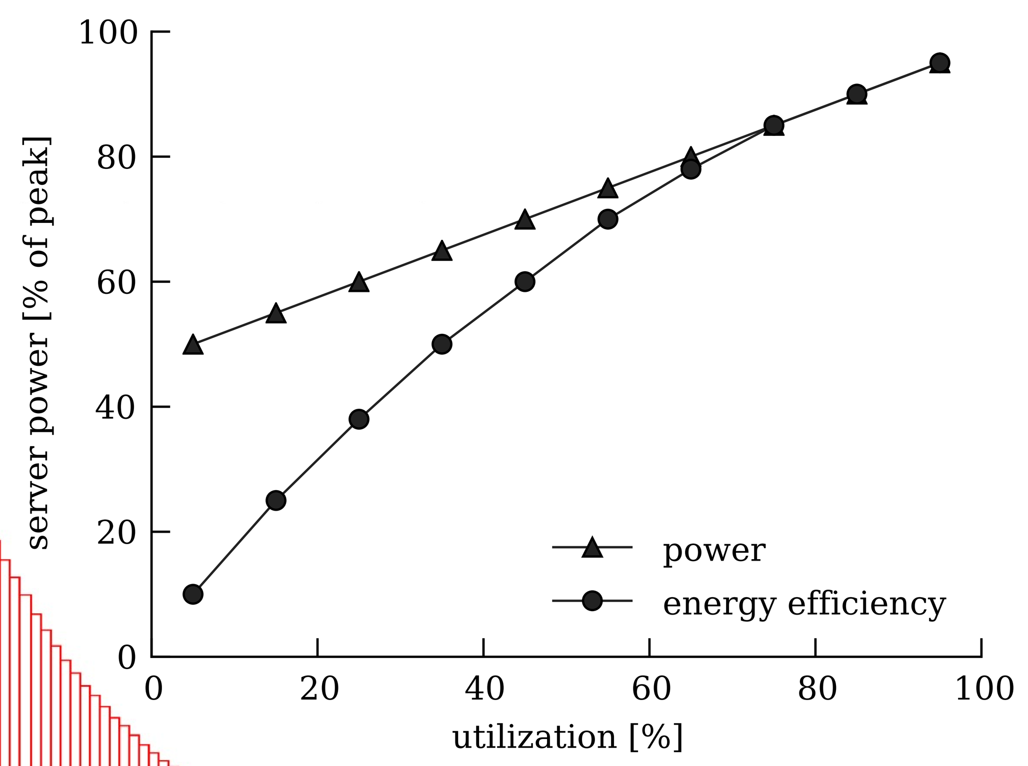
Consolidating web services for energy-efficiency

Thomas Knauth
Christof Fetzer
Technische Universität Dresden, Germany

Motivation



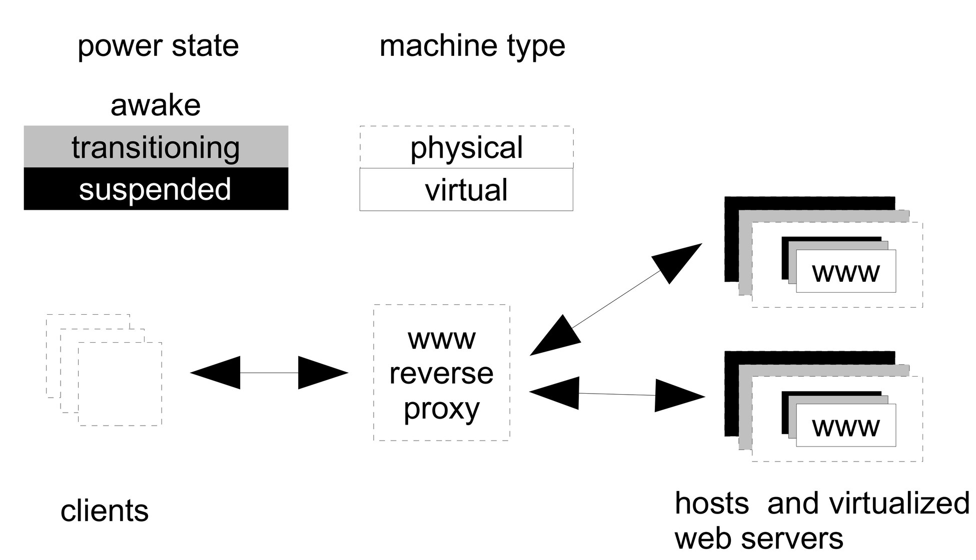


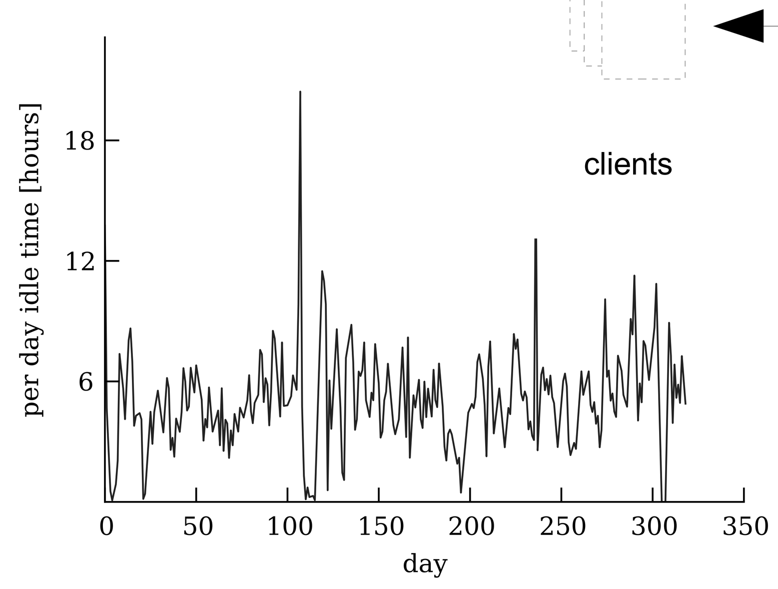
Servers exhibit low energy efficiency at low utilization levels and their typical operating region is between 10-50% [1]. Following the industry's best practices, most of the energy is already consumed by the compute infrastructure. Power usage effectiveness (PUE) cannot be decreased much further. Our optimizations target the servers.
Previous work on utilizing low power modes for energy efficiency focussed on batch instead of interactive workloads [2,3,4]. Batch workloads are less sensitive to intermittent performance degradation due to migration and resuming physical resources. For interactive workloads different trade-offs are necessary.
In our work we suspend, resume, and migrate virtualized web services to vacate physical servers. Unused servers are transitioned into a low power mode. One particular challenge is the timely instantiation of

General Idea

Virtual and physical machines are transitioned between active and idle state based on the observed request pattern. Previous work focussed on scaling the stateless web service application tier [3]. Scaling stateless components is comparatively easy as new resources can be acquired and released quickly. Stateful components, on the other hand, require the instantiation of 100s of megabytes of state prior to starting the service.
We modified the Apache web server to track the request pattern of backend web services. Configurable policies suspend the virtual machine encapsulating each web service. If a request arrives for a suspended web service, the corresponding virtual machine is resumed. By "parking" unused web services and migration we vacate physical servers. Empty physical servers can be transitioned into a low power mode. Achievable energy savings depend on the request inter-arrival time and for how long virtual/physical machines can be turned off. For example, the server hosting our group's web site could be turned off for 5 hours each day (on average).

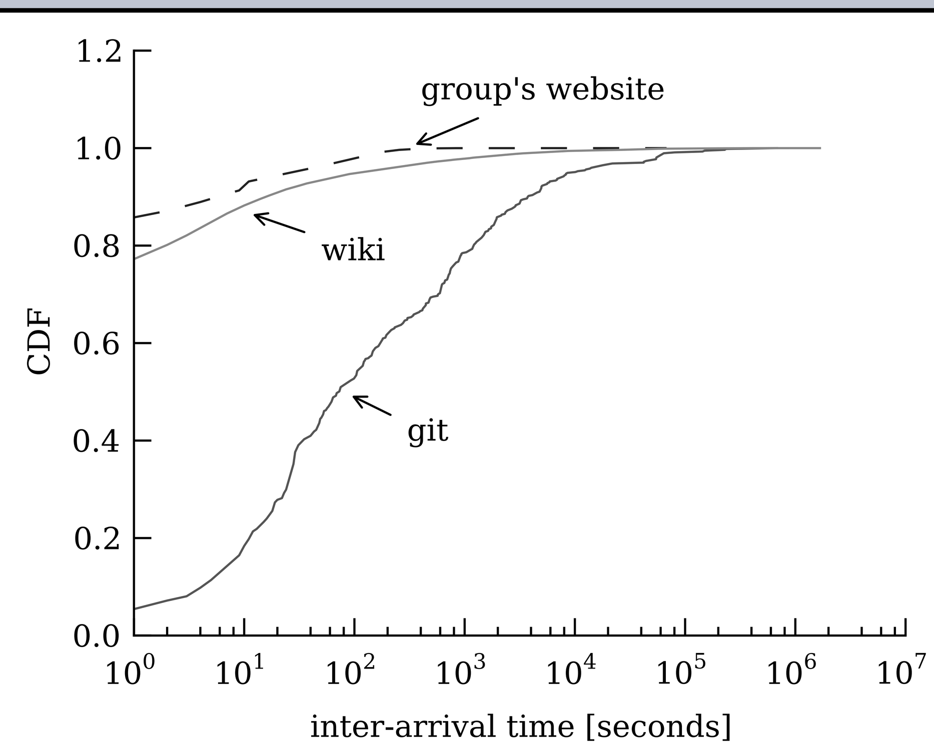
right: system architecture, an application level proxy transitions services and servers between active and idle state



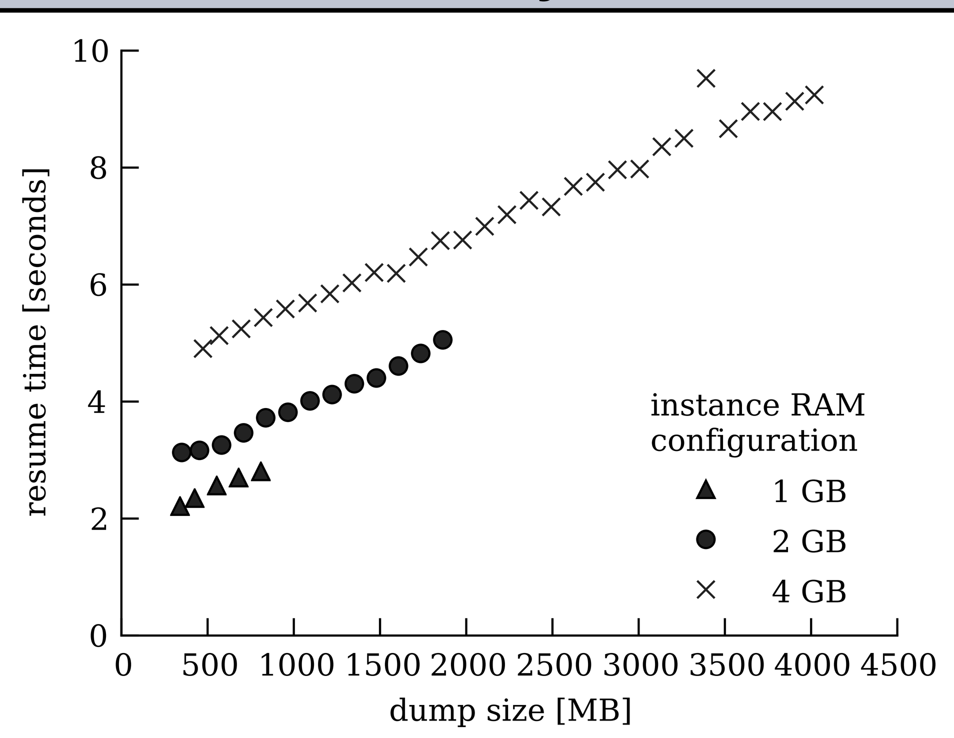


left: daily accumulated time of inactivity for our group's web server; on average, the server could be switched off for 5 hours each day

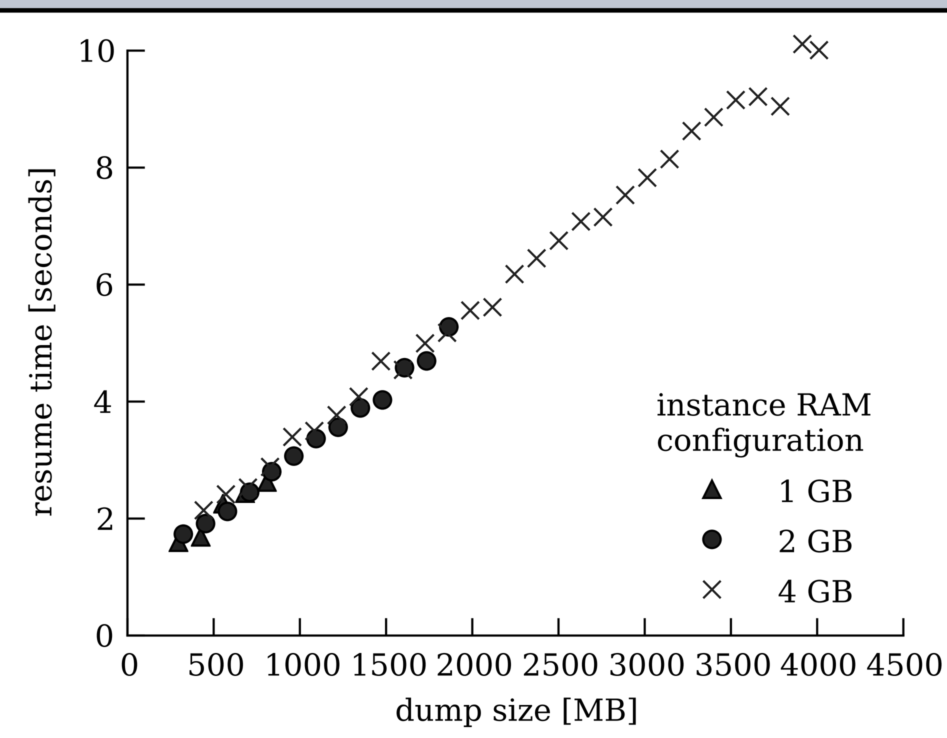
Preliminary Results



Cumulative distribution function (CDF) of request inter-arrival times for three exemplary web services. Idle periods of tens of seconds up to 100s of seconds exist for each service.



Virtual machine (VM) resume time as a function of the VM's state size and configuration. Small instances resume in about 2 seconds, with larger instances taking progressively longer.



qemu has potential for even faster resume times: here we disabled the gratuitous zeroing of empty pages. Resumes times for large instances can be further reduced by lazily reading the state from disk.

Literature

[1] L. Barroso and U. Hölzle. The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, Synthesis Lectures on Computer Architecture, 4(1), 2009
[2] F. Hermenier, et al., Entropy: a Consolidation Manager for Clusters, Conference on Virtual Execution Environments, 2009
[3] Y. Chen et al., Energy efficiency for large-scale MapReduce Workloads with Significant Interactive Analysis, EuroSys, 2012
[4] A. Krioukov et al., NapSAC: Design and Implementation of a Power-Proportional Web Cluster, ACM SIGCOMM Computer Communication Review, 2011

Acknowledgement

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under the ParaDIME Project (www.paradime-project.eu), grant agreement number 318693.

http://se.inf.tu-dresden.de/
thomas.knauth@tu-dresden.de
TECHNISCHE UNIVERSITÄT DRESDEN
PARA DIME