Genetic Algorithms for Energy Efficient Virtualized Data Centers

6th International DMTF Academic Alliance Workshop on Systems and Virtualization Management: Standards and the Cloud

Helmut Hlavacs, **Thomas Treutner**
University of Vienna, Austria

26.10.2012

Table of Contents

Abstract

In A Nutshell

- **Efficiency by dynamic consolidation + workload forecasting**
- **Heterogeneous infrastructure in terms of power, resources**
- **Evaluation of real traces**, University of Vienna Central IT Dept.
- **CPU traces of $\approx$35 VMs, 4 weeks, 2h resolution, VMware**
- **Business infrastructure scenario:** Energy costs are just **one** of several parts of operational costs $\Rightarrow$**Use a cost model!**
- **Cost model, configurable penalties for several cost categories, minimize total weighted costs**
- **Multi-objective** combinatorial optimization problem
- **Comparison of total weighted costs: Balanced First Fit heuristic, Genetic Algorithm, Load Balancing**
- **Forecasting**: (S)ARIMA, Holt-Winters

Scenario

Scenario

- **Highly variable workload intensity, often periodic.**
- No (little?) number-crunching, its not a HPC cluster etc.
- **Minimize energy consumption while avoiding under-provisioning, <u>before</u> reaching 100% utilization!**
- **Queuing issues! Need resources for live migration!**
- **Status costs:**
  - Energy: Linearly correlated with CPU util, future work: SPECpower
  - **Overloads: Queuing, Bad QoS, loose revenue, <u>non-linear</u>, ideally continuous function!**
- **Reconfiguration costs:**
  - **Live Migration: Resource intensive process**
  - Server Boots/Shutdowns: Costs energy, puts mechanical/electrical strain?

Non-linear overload cost function

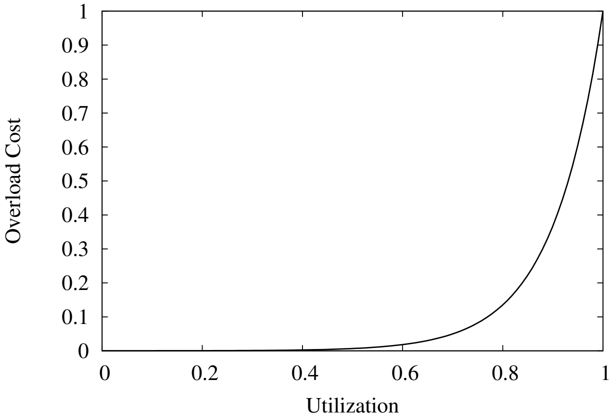
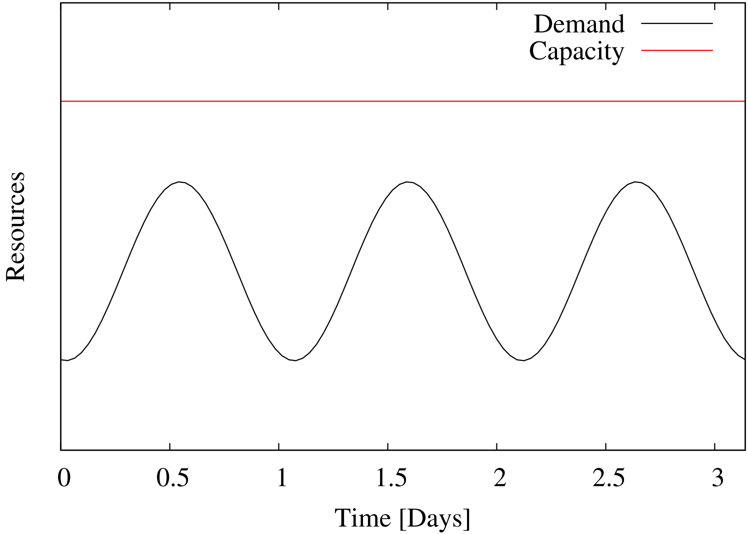
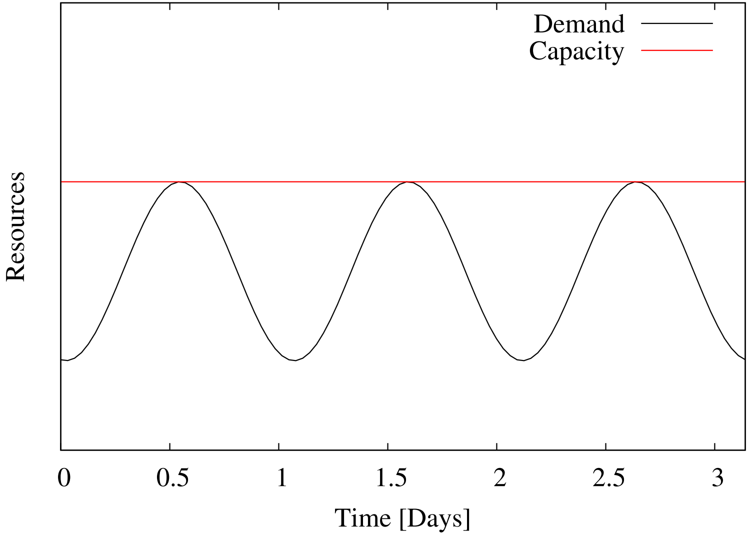


Figure: Markovian M/M/1 queue, $P(T > x) = 1 - F_T(x) = e^{-\mu(1-\rho)x}$, ρ as CPU util, service rate μ and max response time x must be supplied

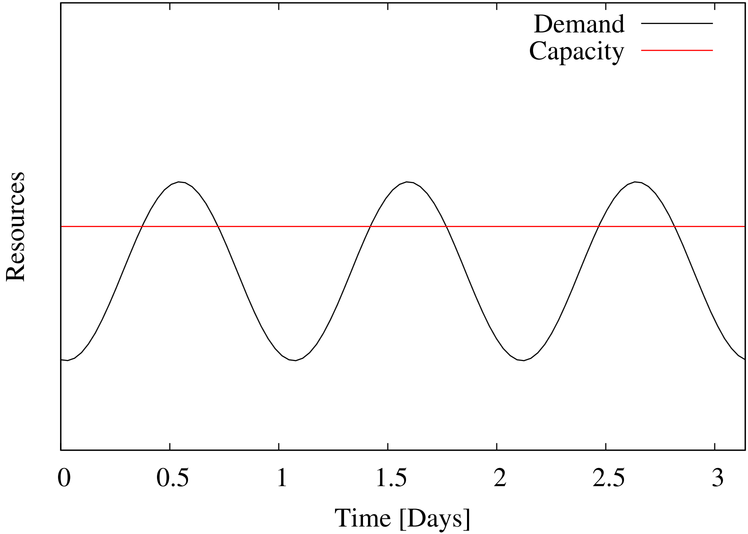
Static Over-provisioning



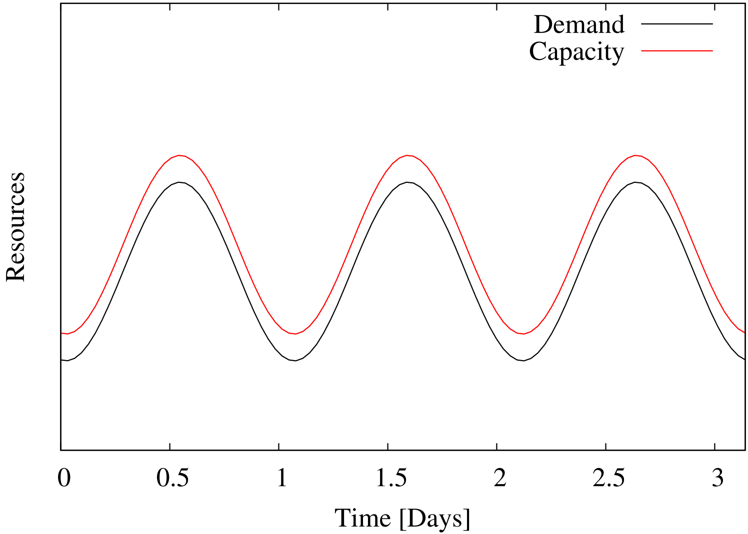
Peak-provisioning



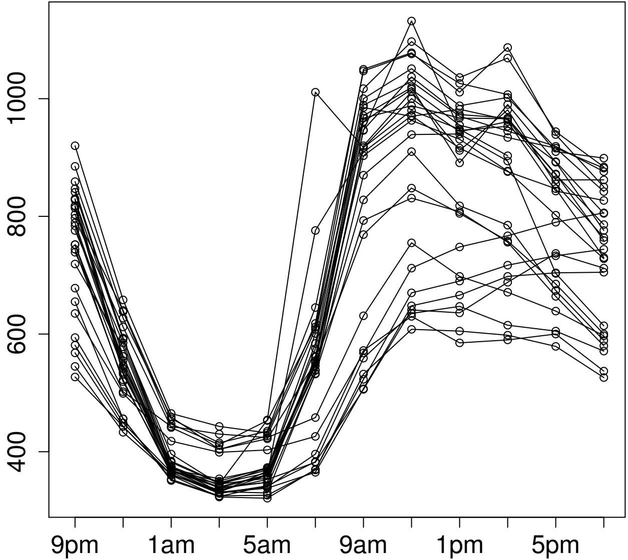
Static under-provisioning



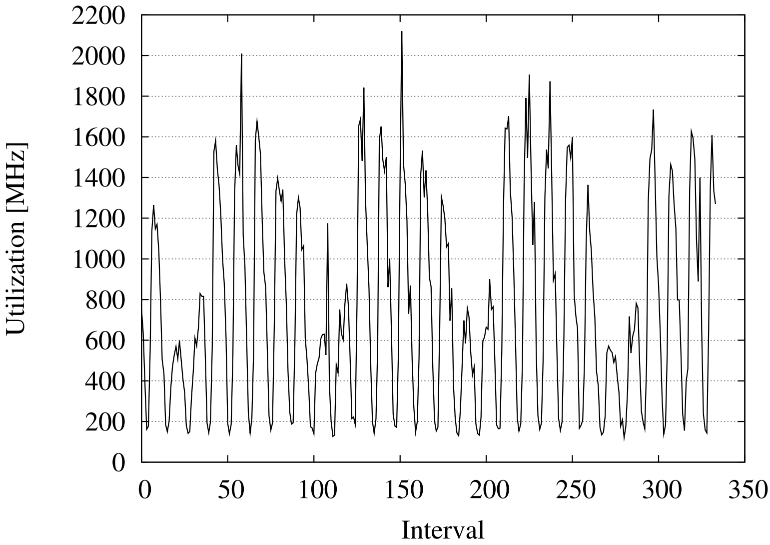
Dynamic Provisioning for Actual Demand



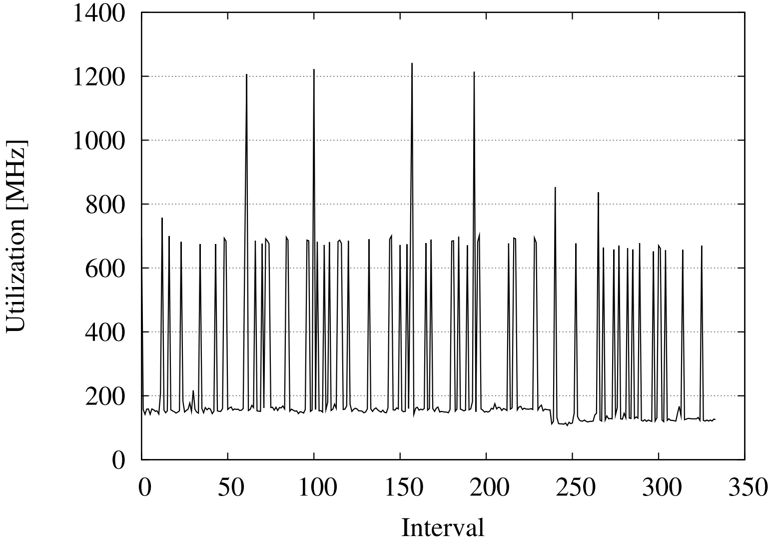
Diagnostic time series plot



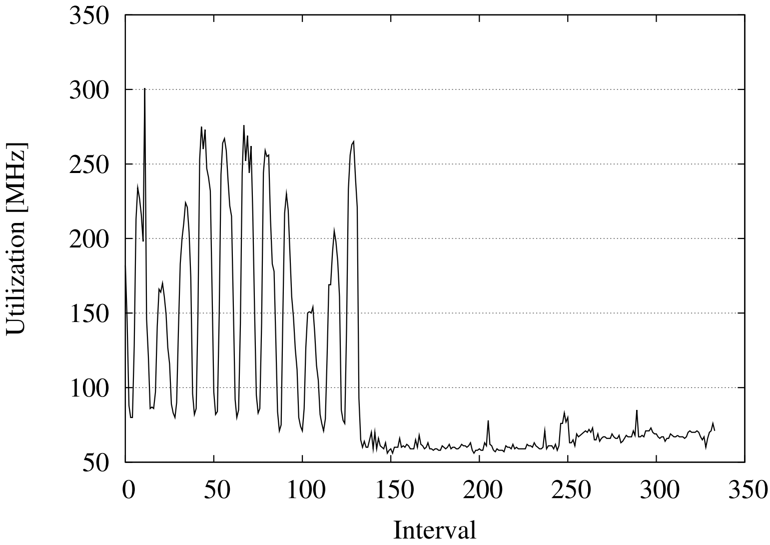
Periodic, seasonal resource demand



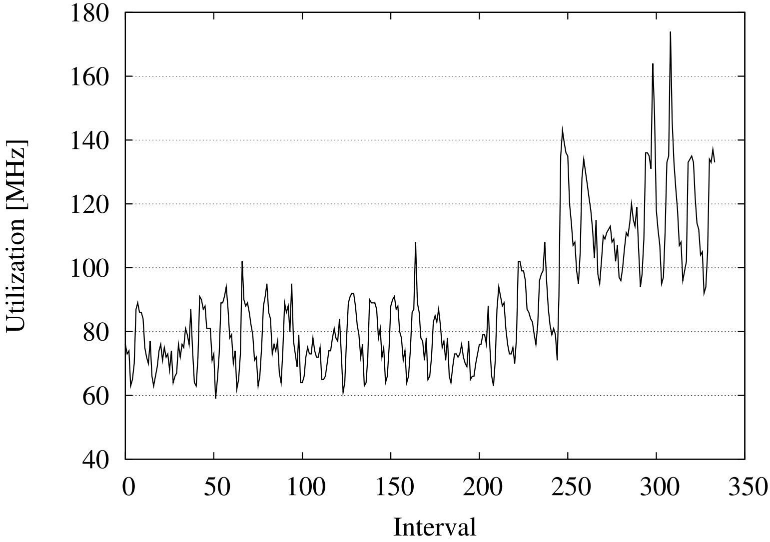
Bursty resource demand



Complete change in resource demand



Seasonal resource demand with a changing mean



universität wien

Balanced First Fit

- **Bin packing related heuristic, inherently inflexible**
- **Not influencable by cost model, but evaluated by it**
- Needs a sorting criteria for the bins, SPECpower_ssj2008 score
- In a nutshell, three phases:
  1. **Check servers for utilizations exceeding threshold**, if so, remove VMs resource-balanced until not overloaded, add VMs to *homelessVMs*
  2. **Try to map homelessVMs** beginning with most energy-efficient. Do this resource-balanced again. If still homelessVMs, force action.
  3. **Try to consolidate** less energy-efficient servers, only if **all** of its VMs can be migrated to more efficient servers, and `vmConsolidationInertia` reached
- Hysteresis control: Turn of servers if `rmIdleTimeout` reached

Genetic Algorithm

- **Meta-heuristic, directly influencable by cost model**
- **Fitness value is reciprocal to the cost of a solution**
- **Lower cost solutions have higher survival chances**
- Cross-over, Mutation, Evaluation, Selection
- Elitism Selection, Roulette Wheel Selection
- Max number of generations, stop if quality not increasing for *n* generations ⇒ Ensures good quality and runtime
- Multi-threading by using demes, randomly exchanging solutions
- **Several mutators defined**
- **Solution defined by mapping matrix**
  - Rows are servers
  - Columns are VMs

Genetic Algorithm: Crossover operator

Single point crossover

$$X_{Father} = \left(\begin{array}{cc|cc} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right); X_{Mother} = \left(\begin{array}{cc|cc} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{array}\right)$$

$$X_{Son} = \left(\begin{array}{cc|cc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{array}\right); X_{Daughter} = \left(\begin{array}{cc|cc} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{array}\right)$$

Genetic Algorithm: swapRM operator

Swap two rows

$$X_{old} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$X_{new} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Genetic Algorithm: swapVM operator

Swap two columns

$$X_{old} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$X_{new} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Genetic Algorithm: migrateVM operator

Migrate a VM

$$X_{old} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$X_{new} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Genetic Algorithm: consolidateRm operator

Move all “1“s to another row

$$X_{old} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$X_{new} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Parameters

- **28 days of trace, first 7 reserved for forecasting, 21 for eval**
- Hundreds of VMs, resampled from the trace data (scaling, memory alloc)
- **26 servers, taken from SPECpower_ssj2008, high diversity**
- Optional linear interpolation to "emulate" more frequent measurements ⇒ VMware export limitation
- **Optional forecasting**, GNU R, auto-model-building for each VM in each interval to consider change in workload pattern
- (S)ARIMA: Limit parameter search and data, takes very long
- **Use 95th upper bound as forecast, very conservative!**
- **Non-linear overload costs:** For every minute of an interval, for every VM running on an overloaded host, multiply cost function value with `rmUtilizationPenalty` and sum up ⇒ Penalize long intervals, as overloads are longer or harder detectable

Relevance	Parameter	Value
All	cpuUtilizationWarningLevel	0.6
	memoryUtilizationWarningLevel	0.8
	utilizationCostFunctionMu	10
	utilizationCostFunctionAllowedResponseTime	1
	energyPenalty	600
	migrationPenalty	1
	rmUtilizationPenalty	10
	bootPenalty	1
	shutdownPenalty	5
Load Balancing	variancePenalty	100000
BFF	rmIdleTimeoutSeconds	900
	vmConsolidationInertiaSeconds	600
GA and LB	numberOfThreads	4
	numberOfGenerations	200
	maxGenerationsOfFitnessNotIncreased	10
	sizeOfPopulation	800
	crossoverRate	0.5
	exchangeRate	0.1
	migrateVmRate	0.3
	swapRmRate	0.1
	swapVmRate	0.1
	elitism	true
Optional Forecasting	requiredPeriodsForForecasting	3

Simulation Input Data

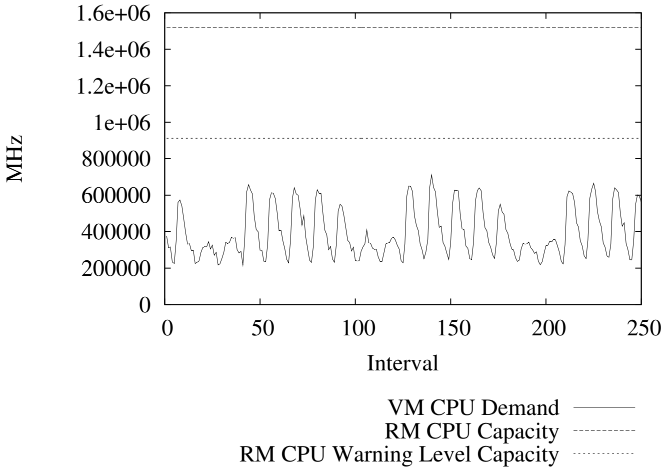
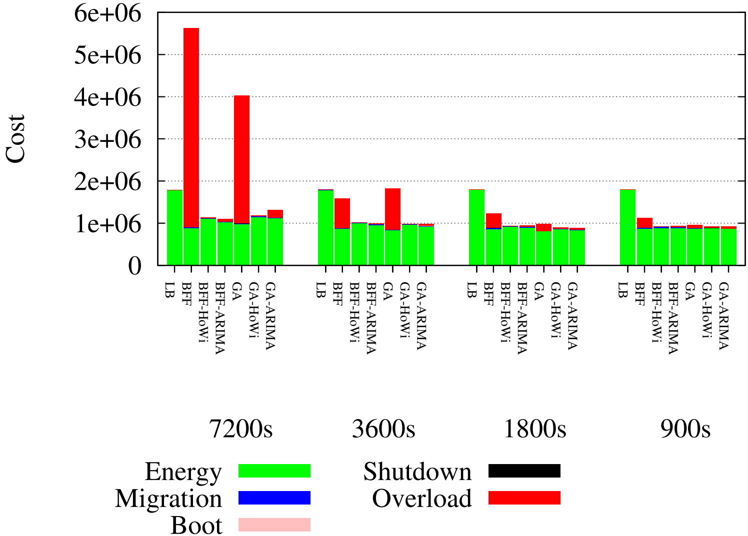
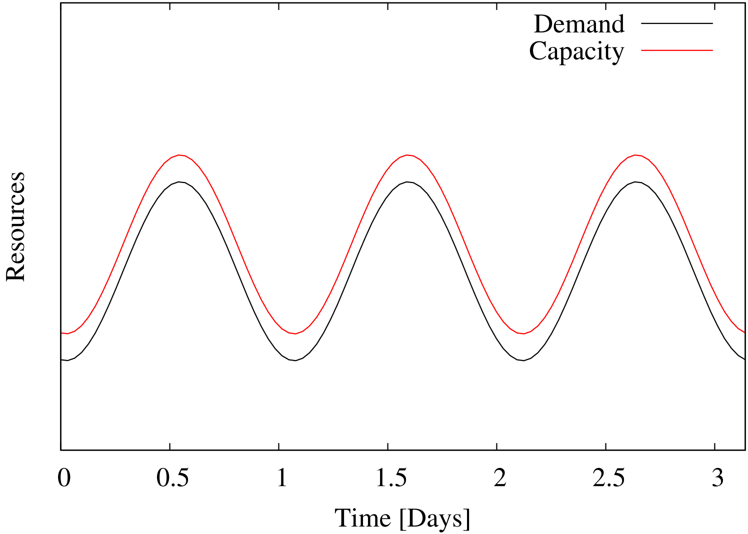


Figure: The time series of total VM CPU demand, server capacity and quota capacity within the warning level used in the simulations.

Total weighted costs



Dynamic Provisioning



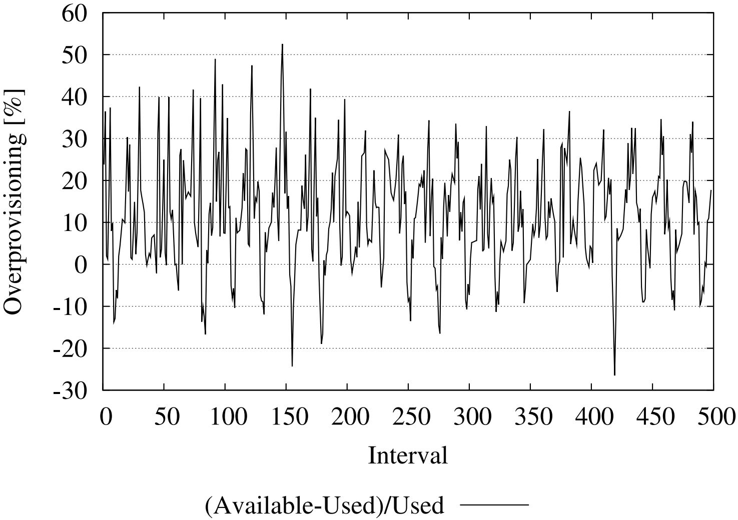


Figure: **Provisioning efficiency for an interval length of 3600 s and the BFF heuristic without forecasting.**

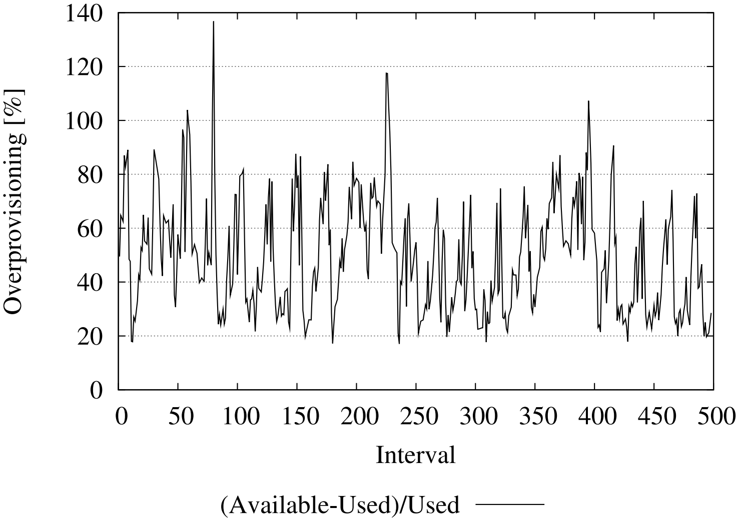


Figure: **Provisioning efficiency for an interval length of 3600 s and the BFF heuristic with Holt-Winters forecasting.**

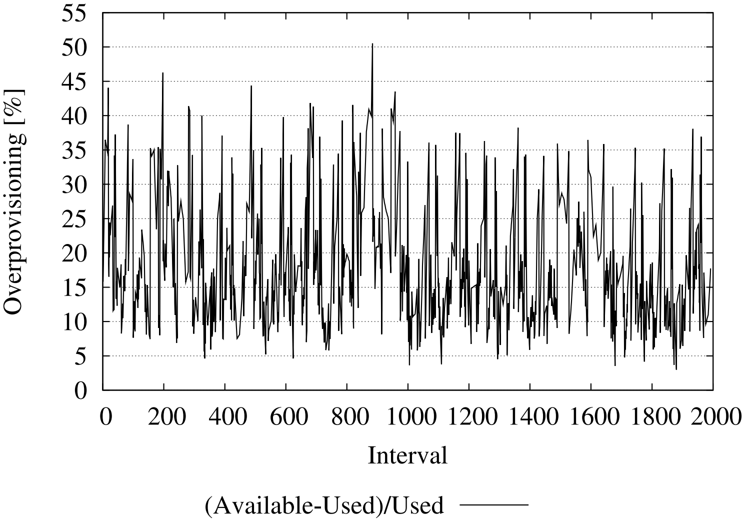
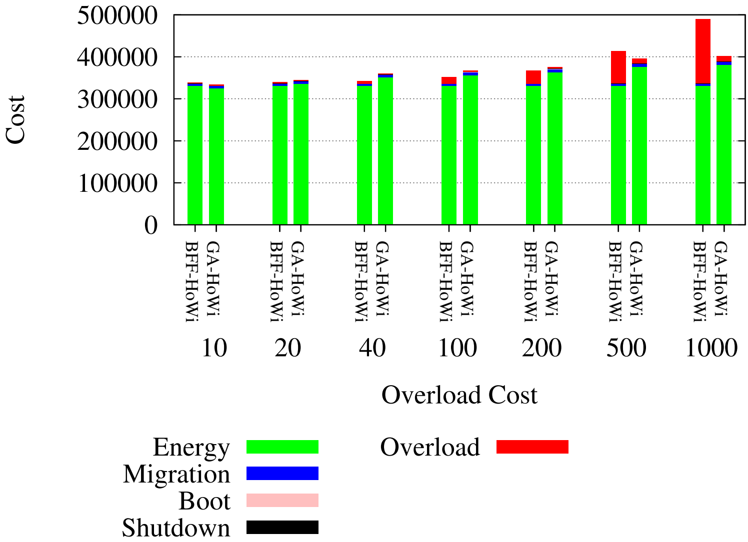
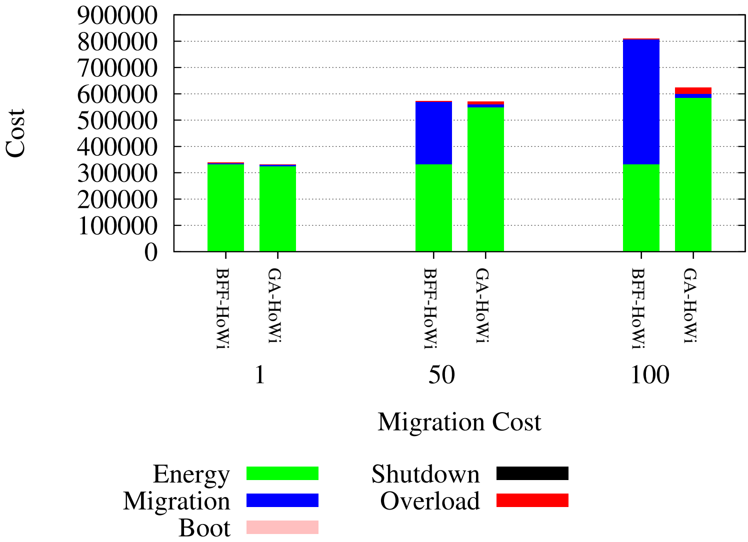


Figure: Provisioning efficiency for an interval length of 900 s and the BFF heuristic with Holt-Winters forecasting.

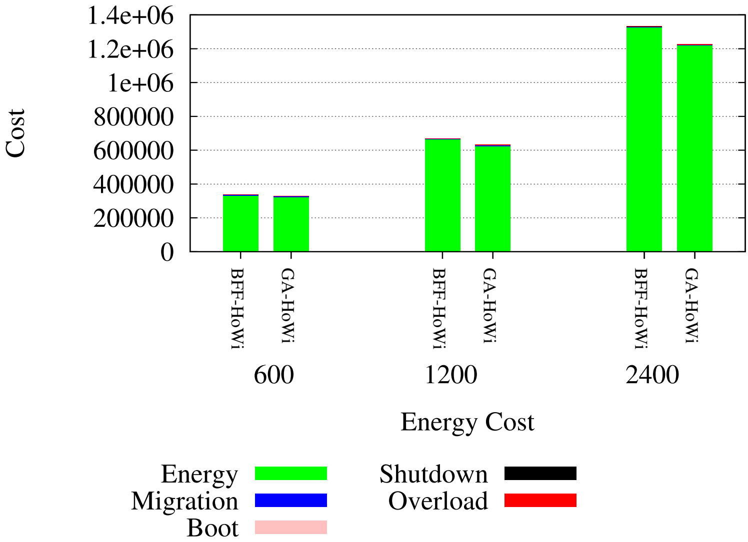
Changing the overload penalty



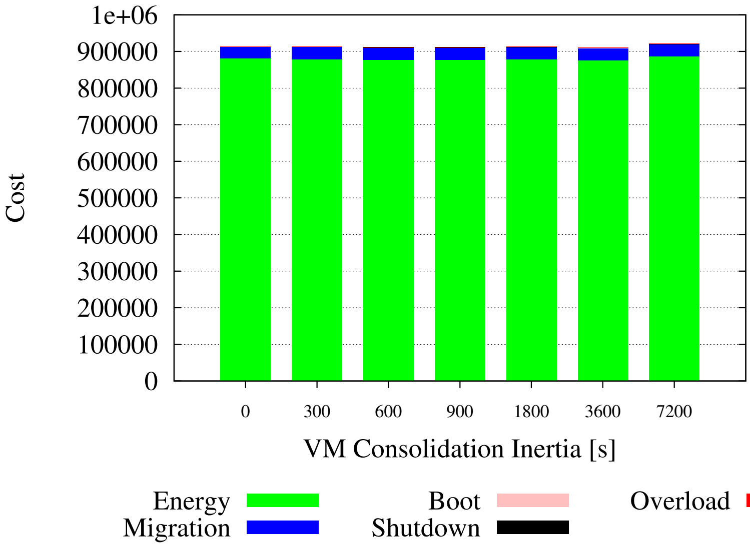
Changing the migration penalty



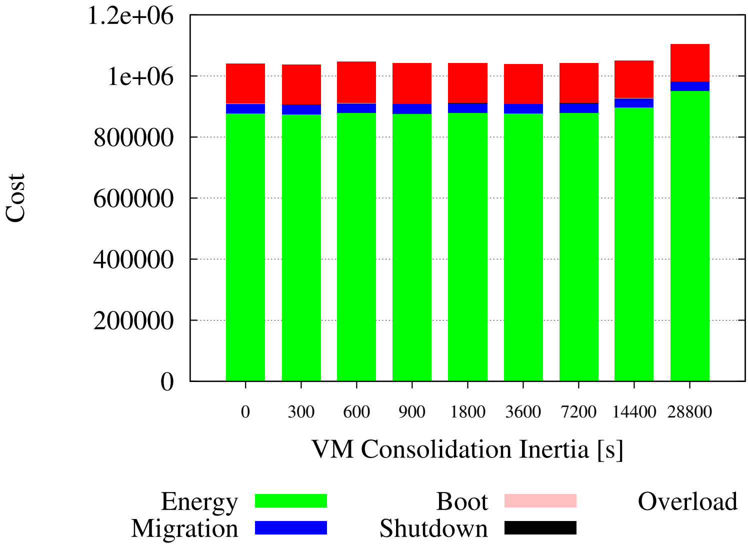
Changing the energy penalty



VM consolidation inertia, Holt-Winters forecasting



VM consolidation inertia, no forecasting



Performance: Hardware Platform Specification

Platform:	Low Power	Desktop	Server
CPU	AMD E-350	PhenomII X4 955	Intel Xeon E5-2670
CPU Frequency	1.6 GHz	3.2 GHz	2.6 GHz
CPU Cores	2	4	8
CPU L2-Cache	2x512 KiB	4x512 KiB	8x256 KiB
CPU L3-Cache	N/A	6 MiB shared	20 MiB shared
CPU TDP	18 W	125 W	115 W
Memory	2 GiB	16 GiB	64 GiB

Table: Description of platforms used in the performance evaluations.

Balanced First Fit

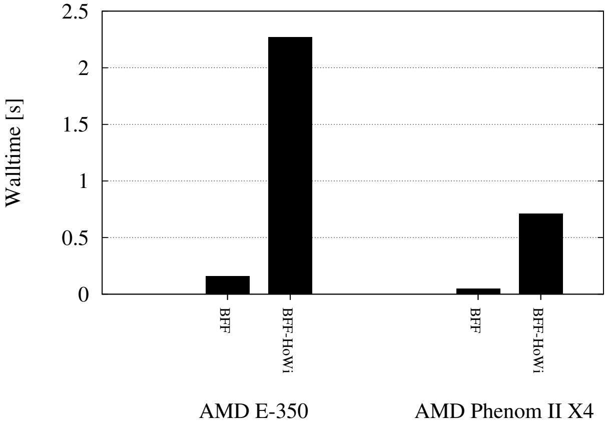


Figure: Runtime per interval of **BFF with/without Holt-Winters forecasting** on a low power CPU and a high-end desktop CPU.

Genetic Algorithm

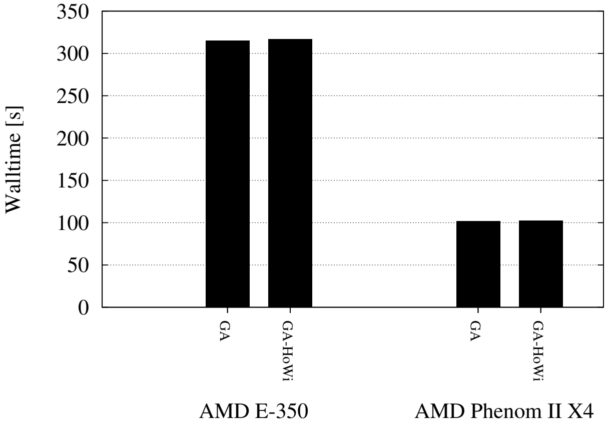


Figure: Runtime per interval of **GA with/without Holt-Winters forecasting** on a low power CPU and a high-end desktop CPU.

Genetic Algorithm Parallelization Speedup

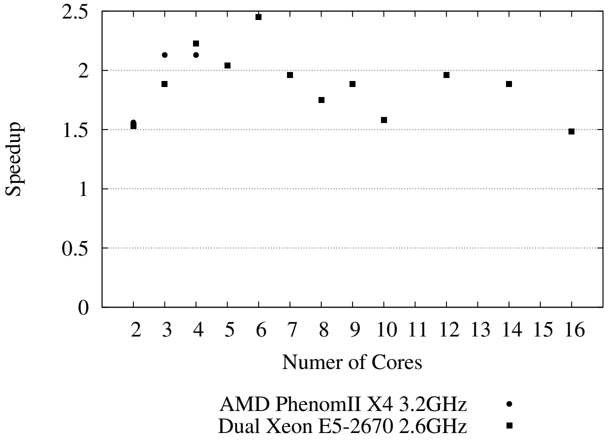


Figure: **Speedup achieved by parallelizing the genetic algorithm**.

Conclusions

- **Flexible cost model** feeding into a GAs fitness function
- **Easy adaptation to diverse optimization demands**
- **Case study parameter sets, drastic reduction of total costs**
- **For long intervals, forecasting is essential**
- **Heuristics faster, but inflexible** to changing parameters (energy price, overload costs etc.)
- **GA can do load balancing** by changing single parameter
- **Future Work:**
  - Non-linear, heterogeneous live migration costs
  - Heterogeneous VM overload costs (*priorities*)
  - Penalizing co-existence of VM pairs on a host (customer isolation, performance issues, security)
  - Speed up GA by storing final solution of the last *n* intervals, replaying them to solution population
  - GA multi-threading speedups?!

Q&A

Thank you for your attention!

http://www.metavisor.org was sponsored by the Internet Foundation Austria within the Netidee 2012 funding programme - http://www.netidee.at