Portfolio Hypervisor Scheduling: Towards Resource Portfolios

M. Alexander[1]

[1]Department of Management Science
TU Wien

M-HPC Topic Presentation, 26.8.2013

Table of Contents

Xen Hypervisor
Characteristics

Paravirtualization main resources

- Dom_0: I/O: disk, network drivers et al.
  - Memory
  - XEN 4.1 Dom_0 is a 32bit paravirtual VM (upper memory constraint)
- Dom_U
  - vCPU
  - CPU
  - Memory

Related Work

Subtitle text

MPT-based portfolio resource manager (compare load-leveler, scheduler) load-balancing appears to be new.

”first to apply portfolio scheduling to data centers and scientific workloads” [2]

Portfolio Scheduling

- Base Assumptions: Returns on resources
  - differ over time
    - steady-state maybe under constant workload type, sequenced jobs
  - Cross-correlation coefficients ϱ_{ij} can be estimated
  - Returns normally or elliptically distributed
  - Objective is to minimize return variance
  - Linear relation between resource return and variance
  - Scheduling on the granularity of multiple resource portfolios provided sufficient workloads to plan against leads to efficient resource utilization
  - Computationally feasible as overlay (job)scheduler - $O(n^3)$

Portfolio Scheduling

Non-pass-through Xen I/O known to be Dom_0, Dom_U
CPU-bound

- Objective function: aggregate throughput measured on Dom_0 egress interface [IP Mpackets/s]
- Lends itself to
  - heterogenous nodes $[\varrho_{ij}] \neq 1$
  - batch processing with multiple queues
- Global optimization to put portfolio on the efficient frontier, project current one towards $min\ \sigma^2$ or $max\ R$
  - scheduling policy?, utility?
  - period as $f(job\ length)$?

Markowitz
Mean Value Optimization

$min\,\sigma^2 \oplus max\,R$: Markowitz Modern portfolio theory (MPT) [3]

- Correlation matrix $\mathbb{P} = [\rho_{ij}]$
- Covaraince matrix Σ

$$\Sigma = [\sigma_{i,j}] = \begin{bmatrix} \sigma_{11} & \cdots & \cdots & \sigma_{n} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \sigma_{n1} & \cdots & \cdots & \sigma_{nn} \end{bmatrix}$$

Markowitz
Mean Value Optimization

- QP problem for min σ^2

$$\min \quad \sigma_{ij}^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j \rho_{ij} \sigma_i \sigma_j \tag{1}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} w_i = 1 \quad , \quad w_i \geq 0 \tag{2}$$

$$\mathbb{E}(R) = \sum_{i=1}^{n} \mathbb{E}(R_i) w_i \tag{3}$$

Relations
Mean Value Optimization

asset	resource types: CPU, memory (multiple instances)
return	MPI throughput | blocksize
return correlation	throughput correlation
variance	variance throughput Interpretation: packet delay variance: jitter
weights	weights
prices	TBD
efficient frontier	$\mathbb{E}[R], \sigma[R]$

Properties-Interpretations

- Main objective is to $max(R)|\sigma^2$
- Why diversify TBD
  - Fairness within a queue
- Reduce Variance - jitter
- Tangent portfolio/Sharpe ratio TBD

Special Cases

- Uncorrelated resources: with $n \rightarrow \infty$ Variance $\sigma^2(R) \rightarrow 0$
- Equality weighted, correlated resources: with $n \rightarrow \infty$ Variance $\sigma^2(R) \rightarrow 0$

Caveats

Mind

- Work sharing/stealing cost {transaction cost, switching environmental context}
  - Optimal portfolio period?
  - Xen *credit* scheduler fixed at 30ms period
- VM {computational, I/O} throughput not necessarily normally distributed
- Xen scheduler (*credit* scheduler as of 4.1) complex global load-balancing interaction
  - Intra-portfolio period changing μ, σ, ϱ
  - t_n static cases does not make a dynamic case

Dom_0 parameters

- dom0_mem=512M
- dom0_max_vcpus=2

Initial Dom_u parameters

```
kernel = '/boot/vmlinuz-3.2.0-4-amd64'
vcpus = '8'
memory = '256'

workaround vCPU offline->online
echo 1 > /sys/devices/system/cpu/cpu{}/online
```

Explorative Experiments
Synthetic Workload

- Worker VMs Debian Jessie und nested Xen 4.1
- Experiment A
  - I/O-bound MPI data traffic, varying blocksizes
  - Depreciated due to inter-VM throughput $>$ 5 Gbps (!) on 2.3 GHz Core i7 3615QM possibly due to zero-copy algorithms with MMU
- Experiment B
  - CPU-bound prime number generation
  - multi-threaded with
  - Vary **vCPU**, active RAM

Does the Model Apply to the Problem?

- Scheduler - Resource management
- Might map to heterogenous environments
  - How to make it comparable - reference workload?
  - How to model affinity, pinning?
  - Might its value be in the dynamic case as computational throughput risk proxy: $\sigma(t)$?

Possible Extensions

- Transcedent memory
- Balloon driver parameters
  - /sys/devices/system/xen_memory/xenmemory0
- Fairness, efficiency
- Global optimum including Dom_0
- Xen scheduler modification
  - Portfolio scheduling as demonstrated
  - Gang/co-scheduling with portfolio granularity

References

Michael Alexander.

Load-balancing for loosely-coupled heterogeneous nodes using data envelopment analysis.

In *4th IEEE International Symposium on Parallel and Distributed Processing with Applications*, page 499508, Heidelberg, 2006. Springer-Verlag.

Kefeng Deng, Ruben Verboon, and Alexandru Iosup.

A periodic portfolio scheduler for scientific computing in the data center (forthcoming).

In *Proceedings of the IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, Boston, MA, May 2013. IEEE.

Harry Markowitz.

Portfolio selection.

The Journal of Finance, 7(1):7791, March 1952.

Graeme West.

An introduction to modern portfolio theory: Markowitz, CAP-M, APT and black-litterman.

Technical report, New Zealand, June 2006.