Optimisation of virtual machine garbage collection policies

ASMTA'11

Simonetta Balsamo Gian-Luca Dei Rossi Andrea Marin

Dipartimento di Scienze Ambientali, Informatica e Statistica
Università Ca' Foscari, Venezia

June 22, 2011

Outline

- Context
- Prerequisites
- Model
- Examples
- Heuristic
- Conclusions

Memory management in HLLs

Automatic memory management (Garbage Collection)

- Easier
- Safer
- Performance issues

Many different technologies

- Algorithms
- Number of phases
- **Blocking** activities (*stop-the-world* approach)
- Activation timings

Performance optimisation strategies:

- Changing algorithms or implementations
- Reducing blocking phases
- **Tuning the activation timing**
    - Service time degradation vs. blocking activities

Model: Assumptions

- Customers (reqs) arrival poisson process, parameter λ
- Scheduling discipline: *Processor Sharing*
- Memory divided in B blocks
- At each customer arrival, b blocks are allocated, according to a discrete random variable probability distribution.
- Service rate μ_i depends on the number i of allocated memory blocks.
- Garbage collector is activated periodically (rate α_i) or when the memory is full.
- The garbage collector frees unused allocated memory blocks with rate γ_i.
- During the garbage collection phase all services are suspended
- The garbage collector stops unconditionally after a random delay, with rate β_i.
- When the system is empty (no customer), the memory is freed instantaneously.

Model: states space

State: a triplet (c, i, g), where

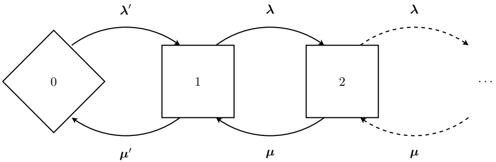
- c is the number of customers in the system
- i is the number of allocated memory blocks
- g is the state of the GC: ON (active) or OFF (non active)

When there is no customer in the system, i.e., $c = 0$, memory is always completely unallocated and the garbage collector is inactive, i.e., $i = 0$, $g =$ OFF. Formally

$$E = (0, 0, \mathsf{OFF}) \cup \{(c, i, g) | c \in \mathbb{N}_{>0}, i \in \{1 \dots B\}, g \in \{\mathsf{ON}, \mathsf{OFF}\}\}.$$

The model is a *Quasi-Birth-Death Process* and is solvable using a *Matrix Geometric* method [3, 2].

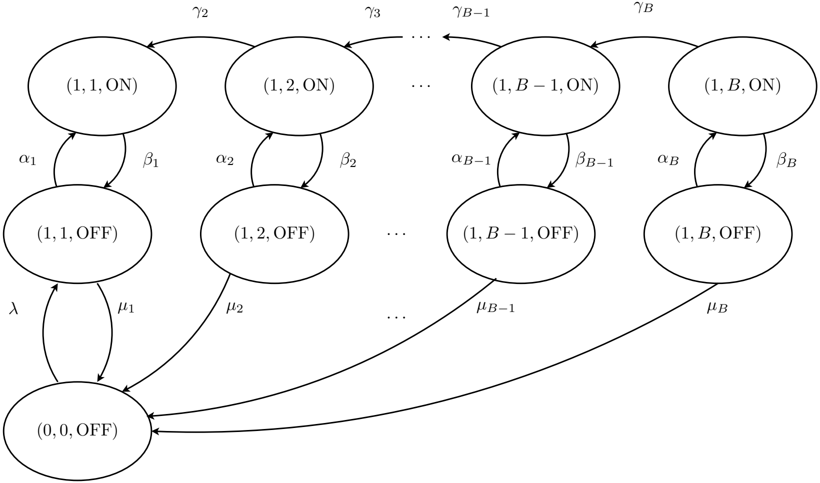
Quasi-Birth-Death Processes



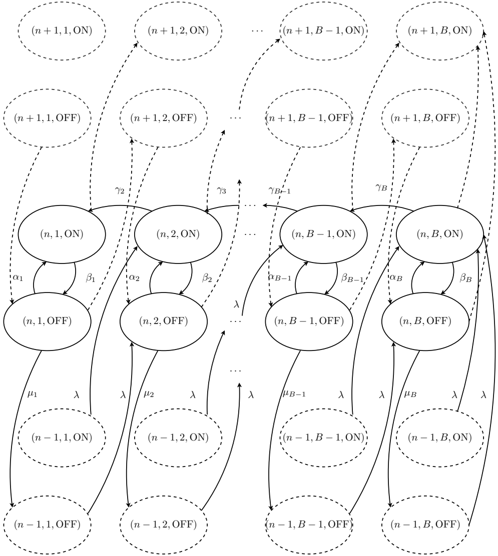
$$\mathbf{Q} = \begin{pmatrix} \mathbf{B}_{00} & \mathbf{B}_{01} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{B}_{10} & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{0} & \dots \\ & & & \ddots & \ddots & \ddots & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

- States are grouped in *levels*
- Transitions are permitted only between states in the same level or in adjacent levels.
- Levels can be represented by square matrices
- Transitions between levels are also represented by matrices
- After an optional initial phase, all levels and transitions have an identical structure.

Model: initial state



Model: regular block of states



Model: matrix representation

$$\mathbf{B}_0(i) = \begin{cases} \mu_{\frac{i+1}{2}} & \text{if } i \text{ is odd} \\ 0 & \text{otherwise} \end{cases} \qquad \mathbf{B}_1(1) = -\lambda \qquad \mathbf{B}_2(j) = \begin{cases} \lambda & \text{if } j = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{A}_0(i,j) = \begin{cases} \mu_{\frac{i+1}{2}} & \text{if } i = j \text{ and } i,j \text{ are odd} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{A}_1(i,j) = \begin{cases} \alpha_{\frac{i+1}{2}} & \text{if } j = i+1 \quad \wedge \quad i \text{ is odd} \\ \beta_{\frac{i}{2}} & \text{if } j = i-1 \quad \wedge \quad i \text{ is even} \\ \gamma_{\frac{i}{2}} & \text{if } j = i-2 \quad \wedge \quad i \text{ is even} \\ -\sum\limits_{\forall k \neq i} (\mathbf{A}_0(i,k) + \mathbf{A}_1(i,k) + \mathbf{A}_2(i,k)) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$\mathbf{A}_2(i,j) = \begin{cases} \lambda & \text{if } j = i+2 \\ \lambda & \text{if } (i = 2B \vee i = 2B-1) \quad \wedge \quad j = 2B \\ 0 & \text{otherwise} \end{cases}$$

Model: performance analysis

The model is a QBD processes

- Matrix Analytic Methods for steady state probabilities
  - Closed forms for $E[N]$ and $E[R]$
  - More performance indices, e.g., GC overhead, using iteration.

Performance indices in function of a variable parameter

- How a performance index, e.g., the average response time, vary over the GC activation rate?
- To simplify the examples, we assume $\alpha_i = \alpha, \beta_i = \beta \; \forall i \in \{1 \ldots B\}$
- $\overline{R}(\alpha)$: mean response time of the model as function of α
- Numerical search for a minimum

Model: minimum search

Where to search for a minimum?

Proposition

If the optimisation problem $\alpha^* = \textit{argmin}_\alpha \overline{R}(\alpha)$ *admits a solution, then the following inequality holds:*

$$0 < \alpha^* < \frac{(\beta + \gamma)(\mu^+ - \lambda)}{\lambda},$$

where $\mu^+ = \max_i(\mu_i)$, $0 \leq i \leq B$.

ATM no proof for minimum existence or uniqueness.

- Experimental evidence seems to suggest so

Numerical Examples

Where not stated otherwise, the parameters are the following

Parameter Name	Value
B	50
λ	3.0
β_i	$3 \ \forall i \in \{1 \dots B\}$
γ_i	$25 \ \forall i \in \{1 \dots B\}$
μ_i	5.0 for $1 \leq i \leq B/2$, 0.05 for $B/2 < i \leq B$

Table: Parameter Values in Numerical Examples

Numerical examples: $\overline{R}$

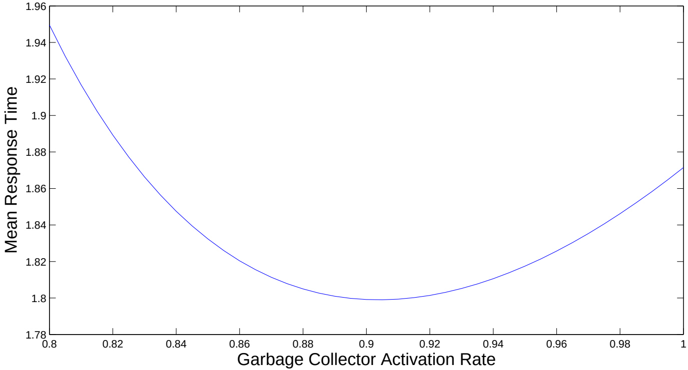


Figure: An example of the function $\overline{R}$

Numerical examples: effects of increasing customer arrivals

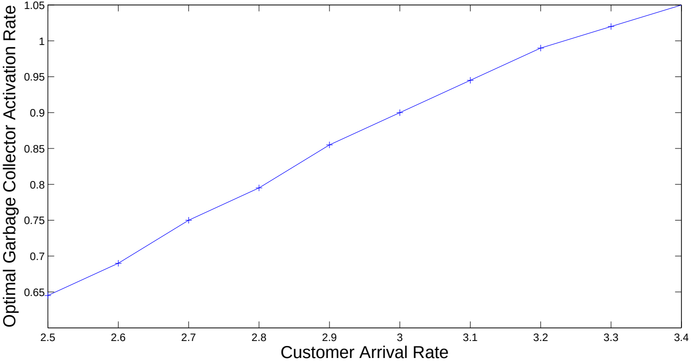


Figure: Optimal α value in function of λ

Numerical examples: utilisation

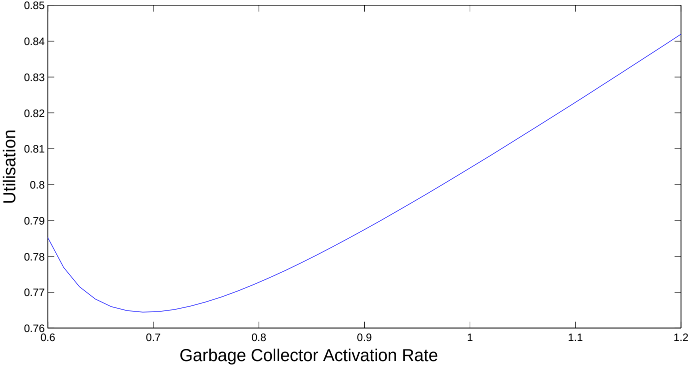


Figure: ρ value in function of α

A Heuristic for optimising GC Policies

Use the model to determine *on flight* the optimum garbage collector activation rates α_i for a real system with GC in order to minimise the average response time.

Suppose that we can measure:

- the number of available memory blocks $\bar{B}$
- the number of free memory blocks f
- the average customer arrival rate $\bar{\lambda}$
- the average service times $\bar{\mu_i}$
- the average rates at which the garbage collector can free a block of memory $\bar{\gamma_i}$
- the average rate at which the garbage collector returns control to the program $\bar{\beta_i}$

Where $i = \bar{B} - f$ is the number of allocated memory blocks.

Easily measurable stats, e.g., using the `-verbose:gc` option of the Sun JVM.

Heuristic: optimisation of α

Parameter α is unbound.

- We can solve the optimisation problem $\alpha^* = \text{argmin}_\alpha \overline{R}(\alpha)$
- We can also solve the problem over a function of α that determines the values of $\alpha_i\ \forall i$
- Other performance indices can be optimised using stationary probabilities.

Once determined the optimum α value, this is set as the activation rate for the GC of the real system.
What if some measured values changes, e.g., $\bar{\lambda}$, or more data is recovered, e.g., for new values of i?

- The model is parametrised again and new values for α_i are generated.

Conclusions

- We have proposed a queueing model for systems with garbage collection
- We have shown that the solution is numerically tractable
- We have proposed a heuristic for the optimisation of garbage collection activation rates
    - Easy to implement

Future works:

- Better validation of the model with experimental data
    - Results in [1] seem to be coherent with results from our model.
- Comparison with traditional policies for garbage collection activation
- Energy-aware optimisation of the activation rate

References

[1] Matthew Hertz and Emery D. Berger. Quantifying the performance of garbage collection vs. explicit memory management. *SIGPLAN Not.*, 40(10):313–326, October 2005.

[2] G. Latouche and V. Ramaswami. *Introduction to Matrix Anlytic Methods in Stochastic Modeling*. Statistics and applied probability. ASA-SIAM, Philadelphia, PA, 1999.

[3] M. F. Neuts. *Matrix Geometric Solutions in Stochastic Models*. John Hopkins, Baltimore, Md, 1981.

Any question?

Stationary distribution of the model

The minimum of the utilisation ρ is different from the minimum of the average response time $\overline{R}$.
This behaviour can be explained by the fact that, for this model, given two distribution $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$, after a certain k,

$$\boldsymbol{\pi}(k+i) < \boldsymbol{\pi}'(k+i) \text{ for } i \in \mathbb{N}$$

even if $\boldsymbol{\pi}(0) > \boldsymbol{\pi}'(0)$.

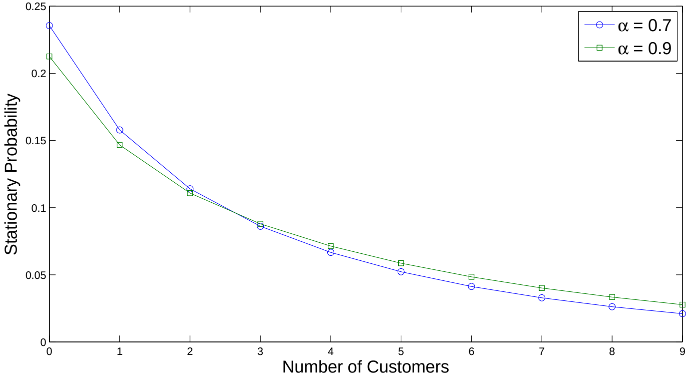


Figure: Vector $\boldsymbol{\pi}$ for two different α values and $\lambda = 3$