On the Combined Behaviour of Autonomous Resource Management Agents

Siri Fagernes and Alva L. Couch

June 24, 2010

The Vision of Autonomic Computing (AC)

Systems that are capable of

- self-management,
- adapting to changes by making their own decisions,
- based on status information sensed by the system itself.

Common Approach in AC

- Autonomic **control loops**,
- that operates to achieve defined system **goals**
- based on **predicted models** of system behaviour.

The Question of Knowledge

- Precise models of system behaviour require huge amounts of information.
- As dynamic behaviour and size of the systems increase, the **complexity** of information becomes overwhelming.
- Some of this information may not even be **knowable.**

Requiring less information for system management is beneficial!

Our Claims

- Minimal information can lead to **near-optimal** behaviour through use of **highly-reactive** management agents.
- Highly reactive agents can be **composed** without chaotic interactions.

Resource Management using Autonomic Operators

Exploring Resource Management Agents

- In 2009, prof. Alva Couch (Tufts University) proposed a theoretical model of **autonomic resource management**.
- The model **does not require complete information** of system behaviour, and still it is able to perform at near optimal levels.
- A high level of **reactivity** seems to compensate for lack of detailed knowledge.

This paper: can the agents be **composed** without chaotic interactions?

The Resource Management Model

- A system delivers a service with response time (performance) **P**
- Use of resources **R** with a cost **C**
- The service has a perceived value **V**
- System **goal**: balance **cost** and **value**

Basic Model

- One control loop affects the resource domain
- Influenced by unknown parameters that are built into the model
    - Load **L**
    - External influences **X** – "unknowable"

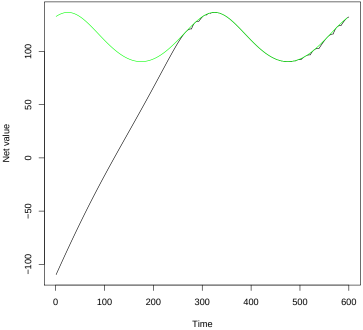
Basic Model - Dynamics

- The component in charge of controlling the resource usage receive feedback of the perceived **value** of the delivered service.
- Value feedback is used by the component to estimate whether it is beneficial to **reduce** or **increase** the resource usage.

Basic Model - Variables

- Performance P(R, L) = $\frac{L}{R}$
- Cost C(R) = R
- Value V(P) = 200-P

Results: measured net value



Green=optimum, black=actual

The Composition Problem

- How can we use several different control loops,
- That operate upon and influence the same system,
- At the same time?

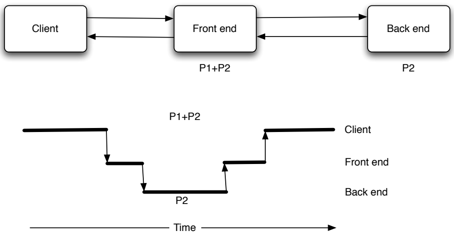
Model and Simulations

We Extended the Original Model

System performance depend on two resource variables R_1 and R_2:

$$P = \frac{L}{R_1} + \frac{L}{R_2}$$

Scenario: Front-end + Back-end



The total system response time depends on two processes. Transmission time is ignored.

Our Goal

- the variables should be updated without centralised coordination or (complete) coordinated knowledge

Performance and Value

Value function (for the overall system):

$$V = 200 - P = 200 - \frac{L}{R_1} - \frac{L}{R_2}$$

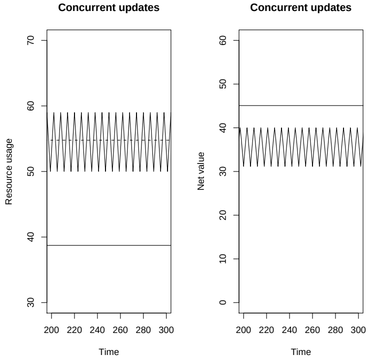
Choice of Algorithm

How should the variables $R1$ and $R2$ be updated?

- concurrently?
- taking turns?

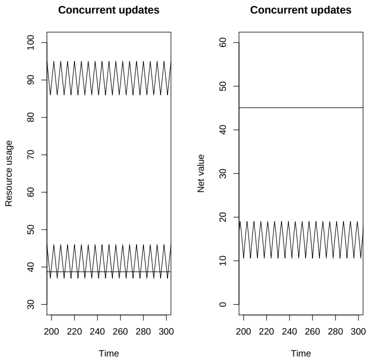
Results

Concurrency Leads to False Optima



Initial resource values: $R_1 = R_2 = 50$.

Concurrency Leads to False Optima (II)



Initial resource values: $R_1 = 1, R_2 = 50$.

'False Optimum'-Explanation

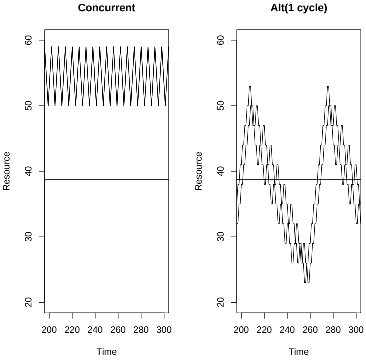
- Each of the variables get updated based on feedback of the global system's **overall** performance P.
- P depends on both $R1$ and $R2$.
- An estimate from $R1$ would not incorporate the cost of $R2$.
- Consequence: their individual estimate of the optimum is **wrong**.

Estimating the 'False Optimum'

- Each operator receives feedback of value $V = 200 - \frac{L}{R1} - \frac{L}{R2}$.
- Their individual estimate of total cost is $C(R1)$ (or $C(R2)$)
- In the special case where R1=R2=R, this could be represented by the following system (as seen from one of them):
  - $V(R) = 200 - \frac{2L}{R}$
  - $C(R) = R$

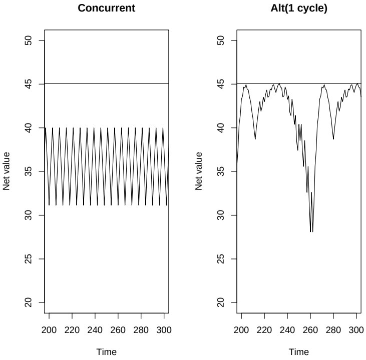
  which means that the net value function is $200 - \frac{2L}{R} - R$, which has the optimal value $R = \sqrt{(2L)}$.

Alternating Between Processes Lead to True Optima and Thrashing



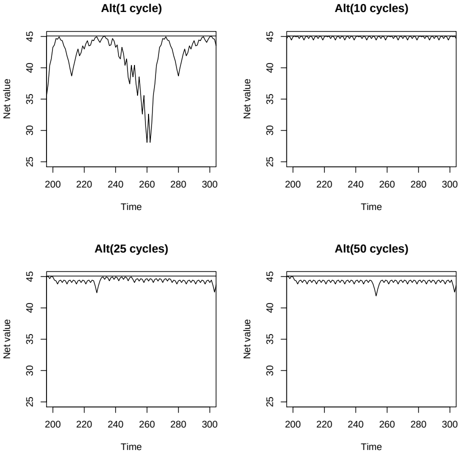
Initial resource values: $R_1 = R_2 = 50$.

Alternating Between Processes Lead to True Optima and Thrashing (II)

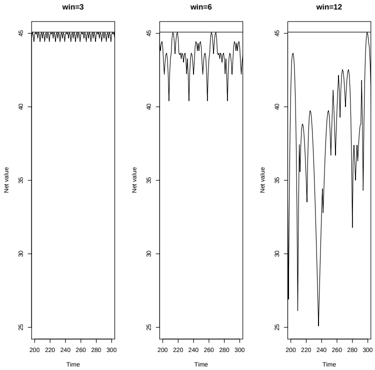


Initial resource values: $R_1 = R_2 = 50$.

The Best-Case Situation

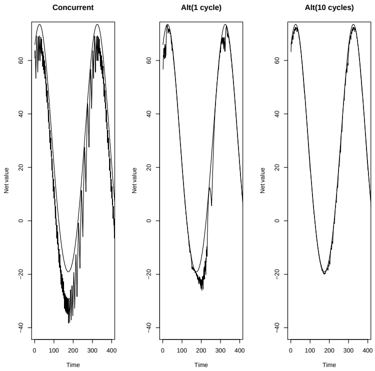


The Best-Case Situation (II)



Initial resource values: $R_1 = R_2 = 50$. Alternating for 10 cycles.

Varying (Sinusoidal) Load



Achieved net system value, sinusoidal load.

Observations

When resource variables are updated **concurrently**:

- If there is a significant difference in their initial values, the lowest value ends up **dominating**, while the highest value never converges to the optimal value.
- When both initial values are equal, both variables converge to the **false optimum** (which would be the optimal value if only one variable and the same system outcome as reported).

Observations

When resource variables are updated only **one at a time**:

- For all scenarios listed on the previous slide, both variables seem to converge to values in an interval between the optimum and the false optimum.
- Not affected by differences in initial values (important!)

Summary

- We have developed a model based on autonomic resource management agents.
- The current model requires very little exchange of information among the agents, and is still able to perform well when certain constraints are fulfilled.

Conclusions

- **Concurrent** updates of the variables leads to false optima.
- **Alternating** updates (with small increments) makes the variables oscillate around the actual optimum levels.
- **Oscillations** can be reduced by tuning certain parameters (small resource increments and short measurement windows).
- Updating the resource variables at **different times** (and hence makes them able to 'observe' each others' influence on the system) is important for robustness of the model.

Contact Information

- Siri Fagernes (siri.fagernes@iu.hio.no)
- Alva Couch (couch@cs.tufts.edu)