Performance-Aware Reconfiguration of Software Systems (PARSY)

Moreno Marzolla[1] Raffaela Mirandola[2]

[1] Università di Bologna, Dip. di Scienze dell'Informazione
marzolla@cs.unibo.it
http://www.moreno.marzolla.name/

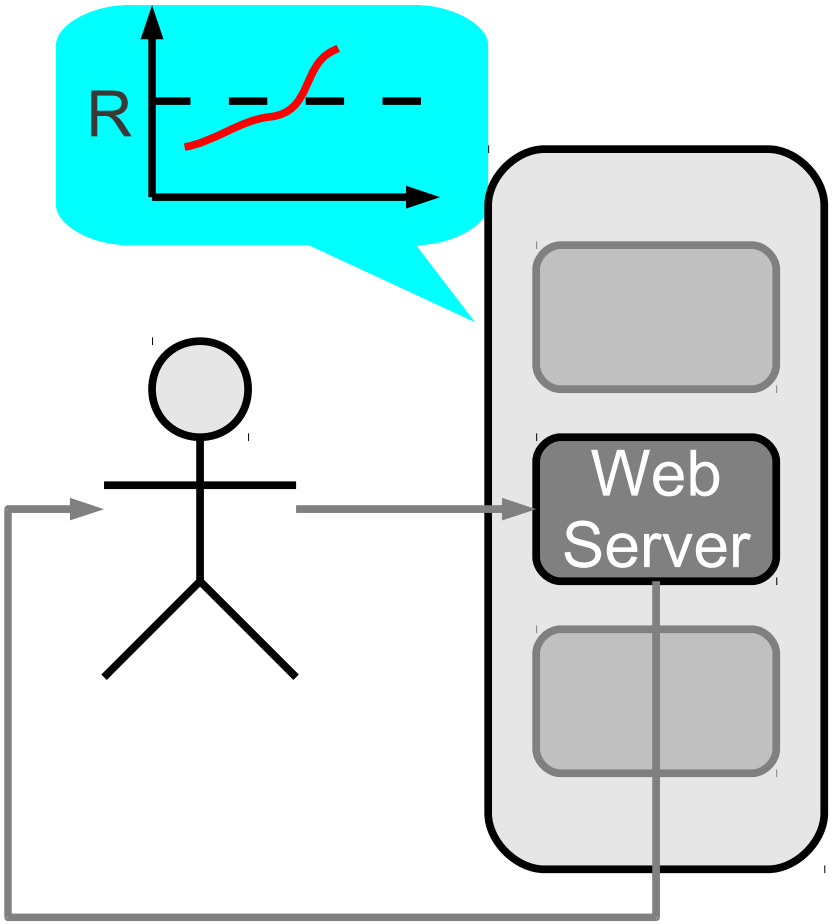
[2] Politecnico di Milano, Dip. di Elettronica e Informazione
mirandola@elet.polimi.it

Problem formulation

- We consider a component-based system which must be enhanced in order to provide an appropriate QoS level...
  - Specifically, the system response time R should not exceed a pre-defined threshold : $R < R_{max}$
- ...under variable workload
  - The number N of users accessing the system varies over time

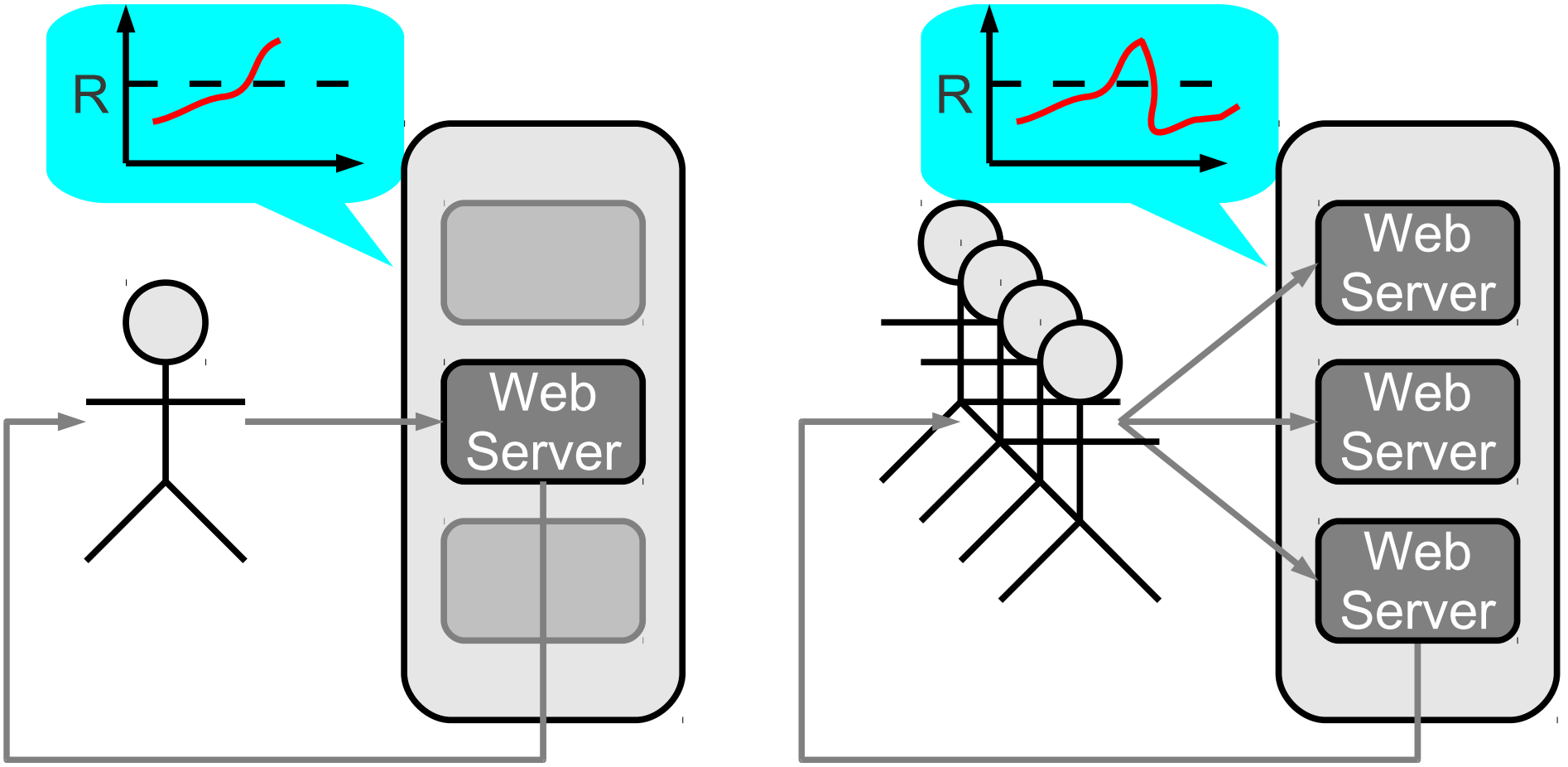
'a-la-Cloud solution

- Instantiate new components/services when necessary, to meet the increased workload



'a-la-Cloud solution

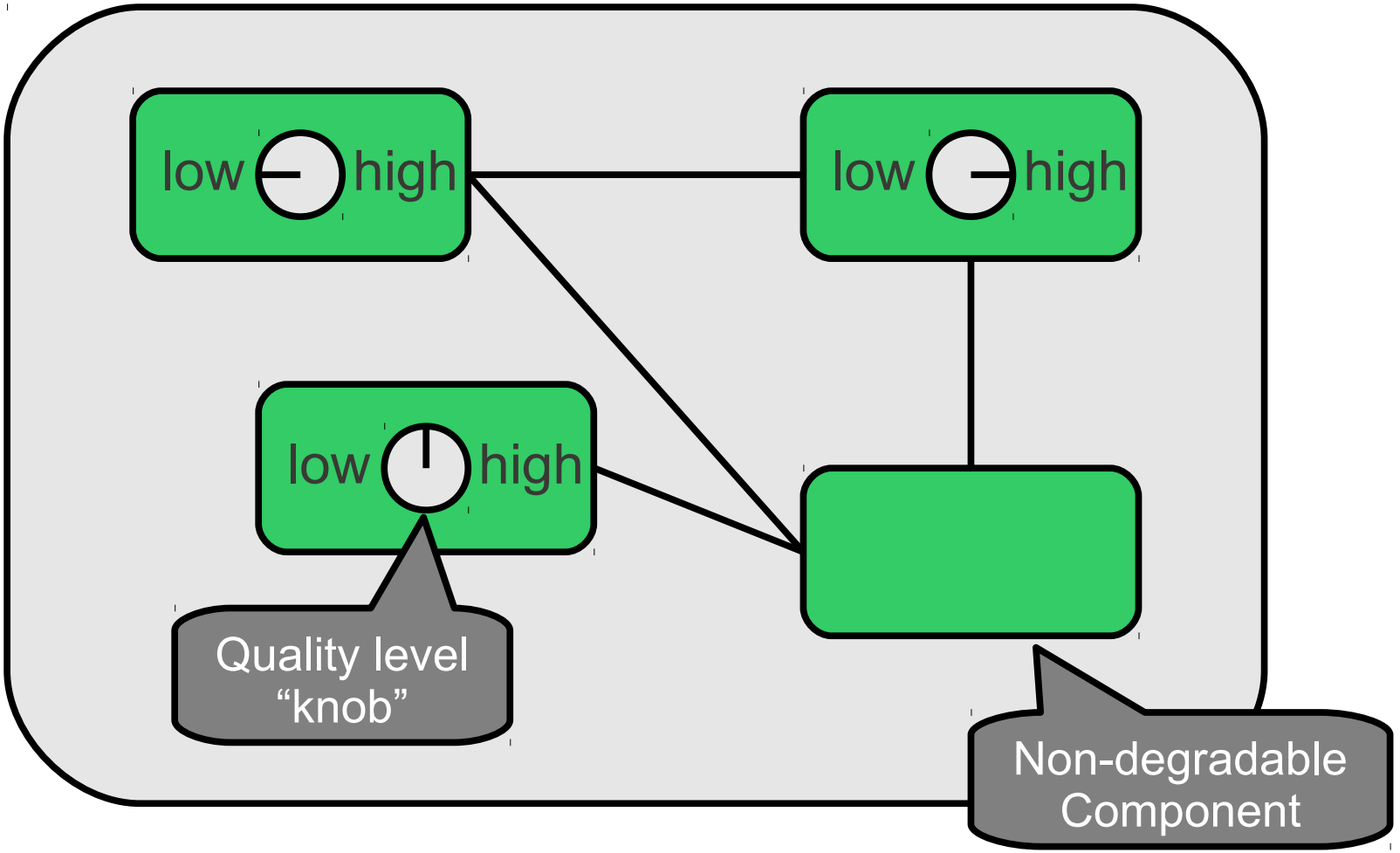
- Instantiate new components/services when necessary, to meet the increased workload



'a-la-Cloud solution
Possible Issues

- Additional resources are not available
- Your application is unable to scale easily
- Bringing up a new service instance can incur significant delays
  - Virtual machine allocation;
  - Virtual machine configuration;
  - Service configuration;
  - Service startup;
  - ...

PARSY
System model



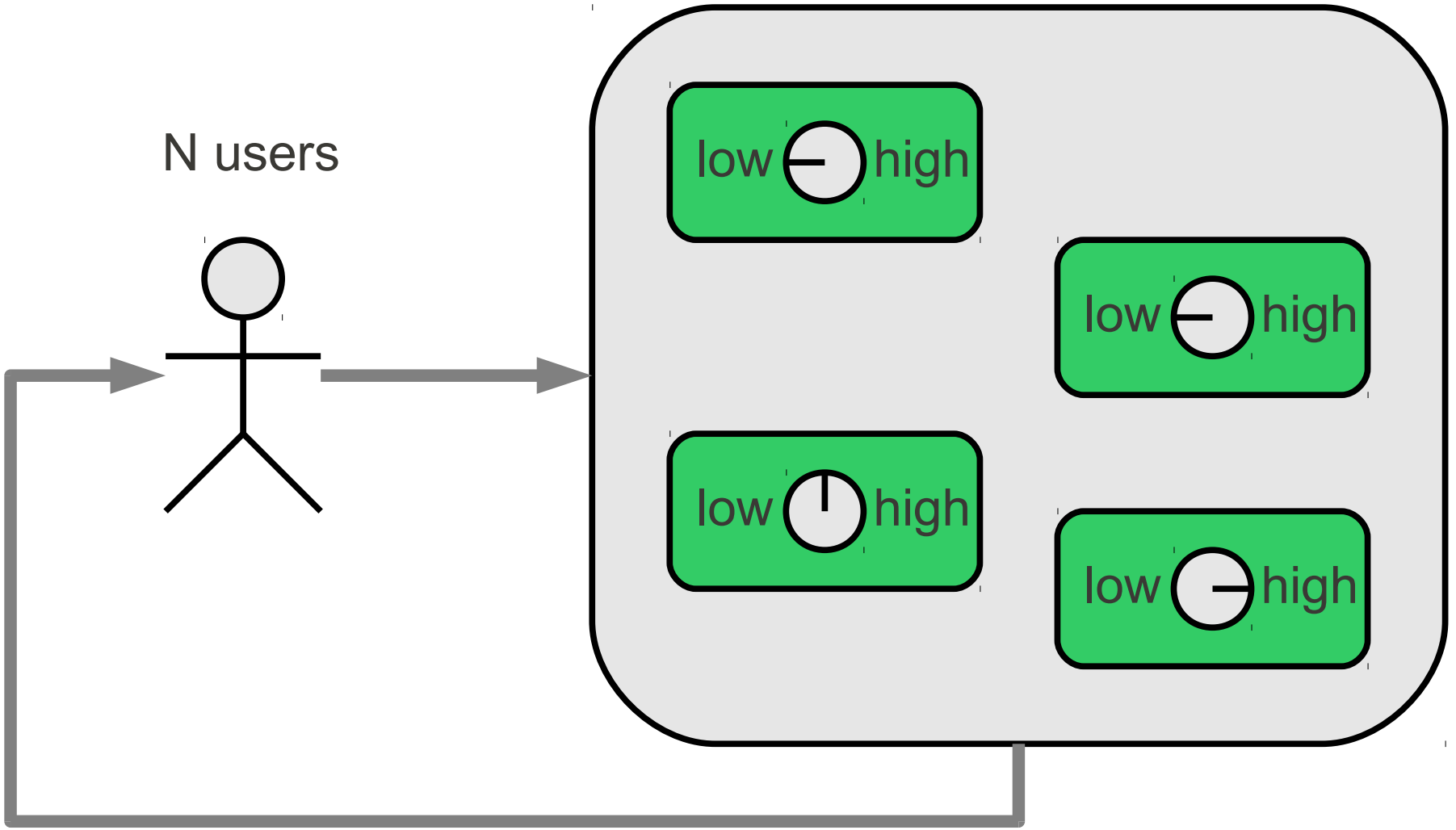
PARSY

- The system is assumed to be made of K components C_1, ... , C_K.
- Component C_k can be configured to provide service at different quality levels 1, ... , L_k (1=worst, L_k=best)
- D(k,j) is the average service demand of component C_k operating at level j
  - $0 < D(k,1) < D(k,2) < ... < D(k,L_k)$
- UT(k,j) is the utility of component C_k operating at quality level j
  - $0 < UT(k,1) < UT(k,2) < ... < UT(k,L_k)$

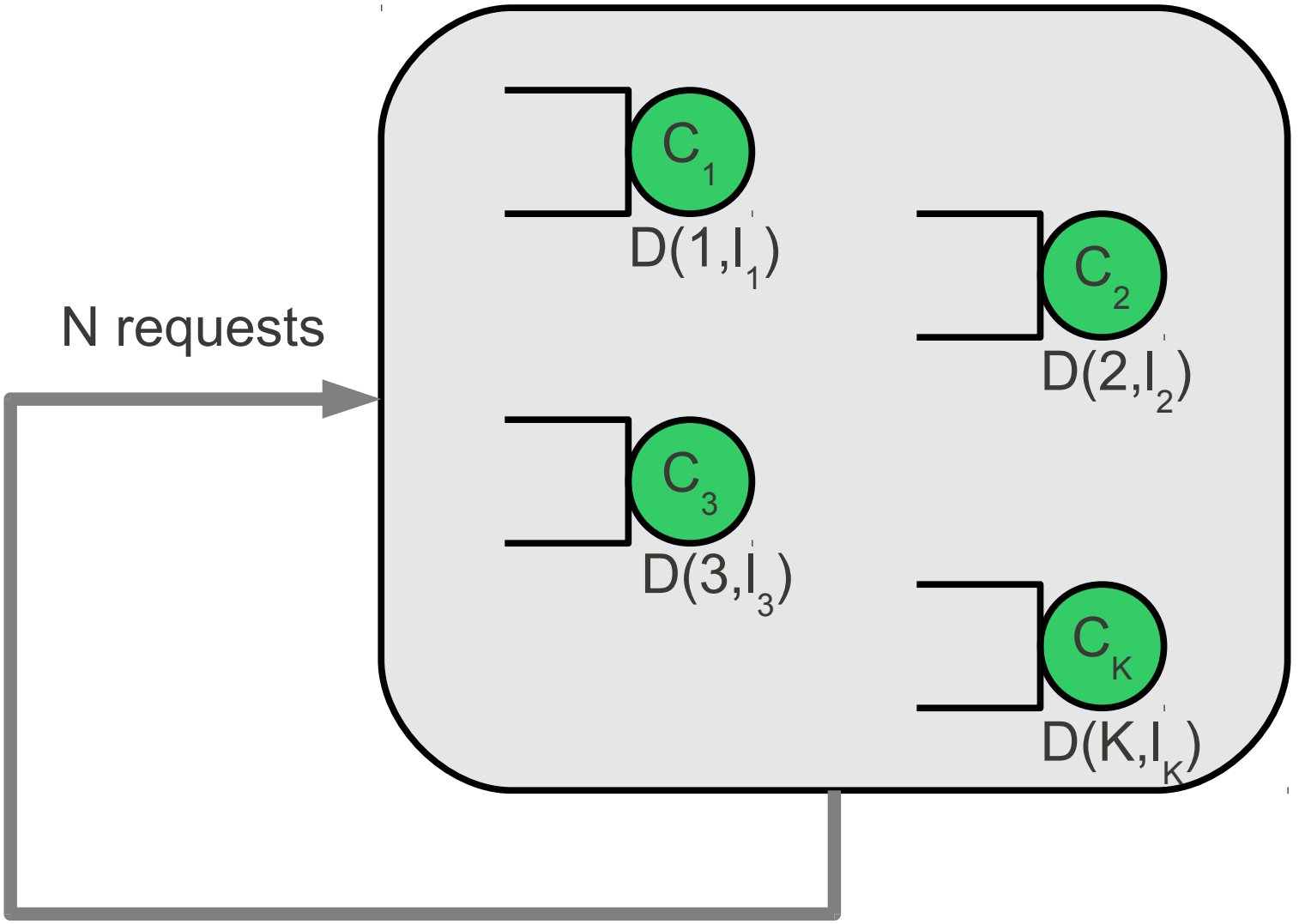
PARSY

- Goal:
  - Maximize: the total system utility
  - Subject to: *estimated response time R* < R_{max}

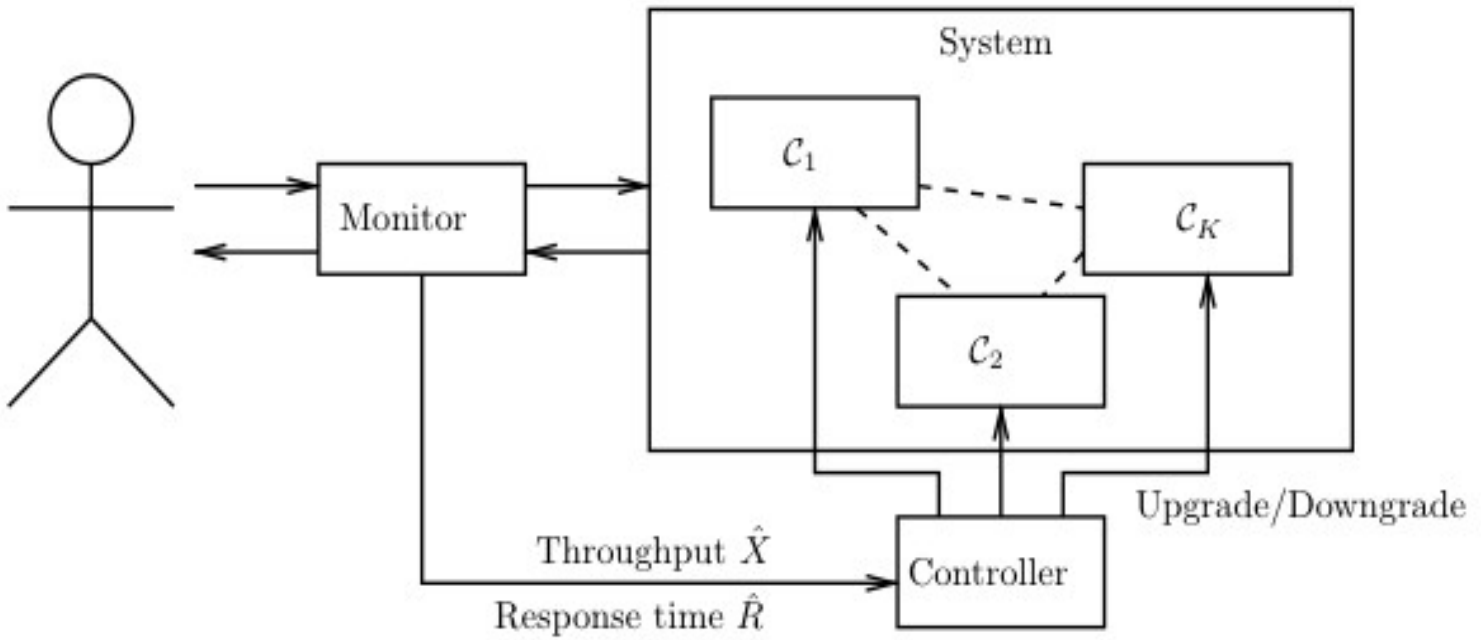
PARSY: system model



PARSY: Performance model



PARSY: System monitoring



- From Little's Law we can estimate N as

$$N = \hat{X}\,\hat{R}$$

Measured Throughput

Measured Response time

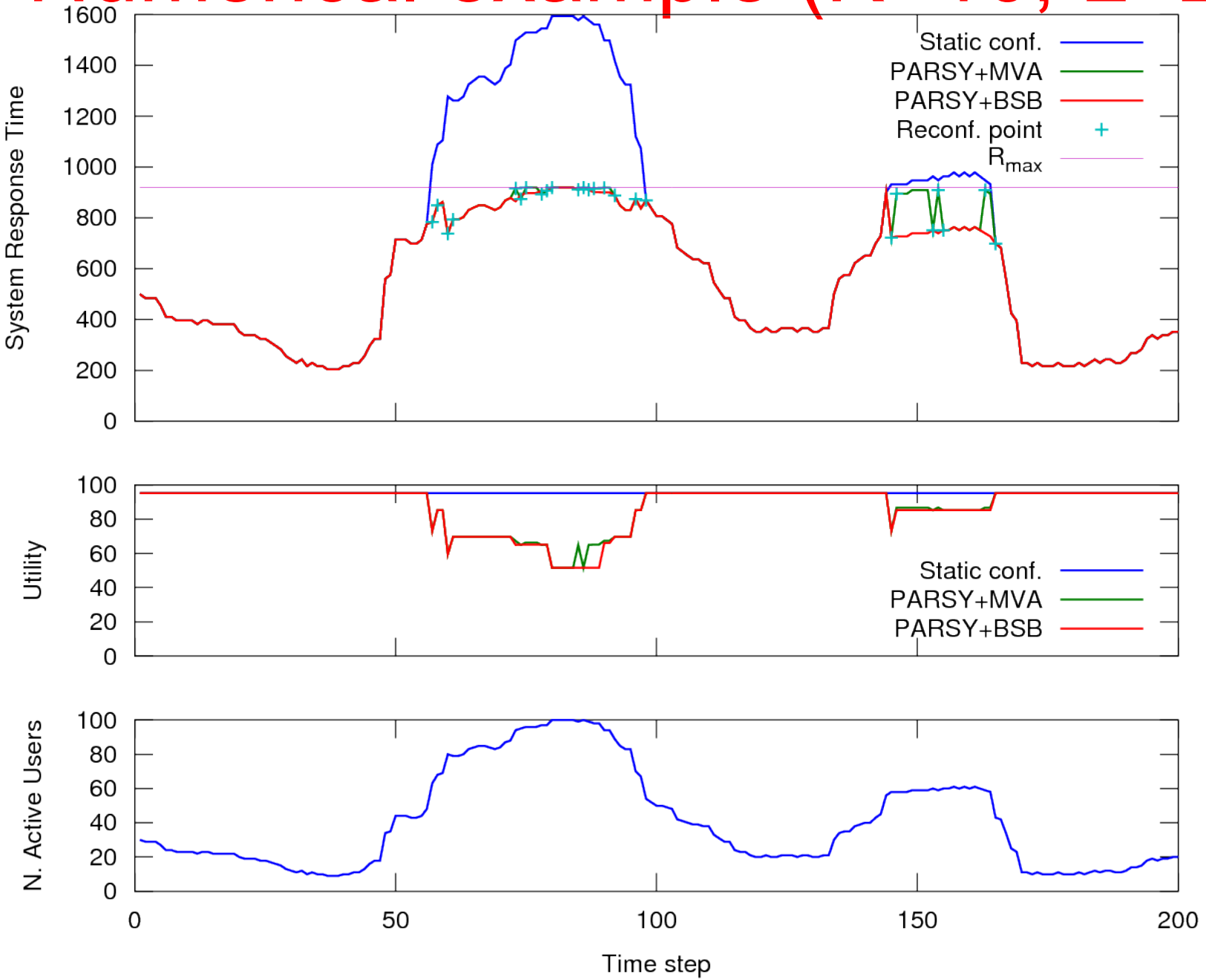
If $\hat{R} > R_{max} \rightarrow$ Degrade

1. Consider the set S of all components which can be degraded (those currently operating at level > 1)
   - If S is empty, stop
2. Select the component k in S with maximum ratio Demand / Utility
3. Degrade component k
4. Estimate new system response time R'
   - If $R' > R_{max}$, repeat from step 1
   - If $R' \leq R_{max}$, stop

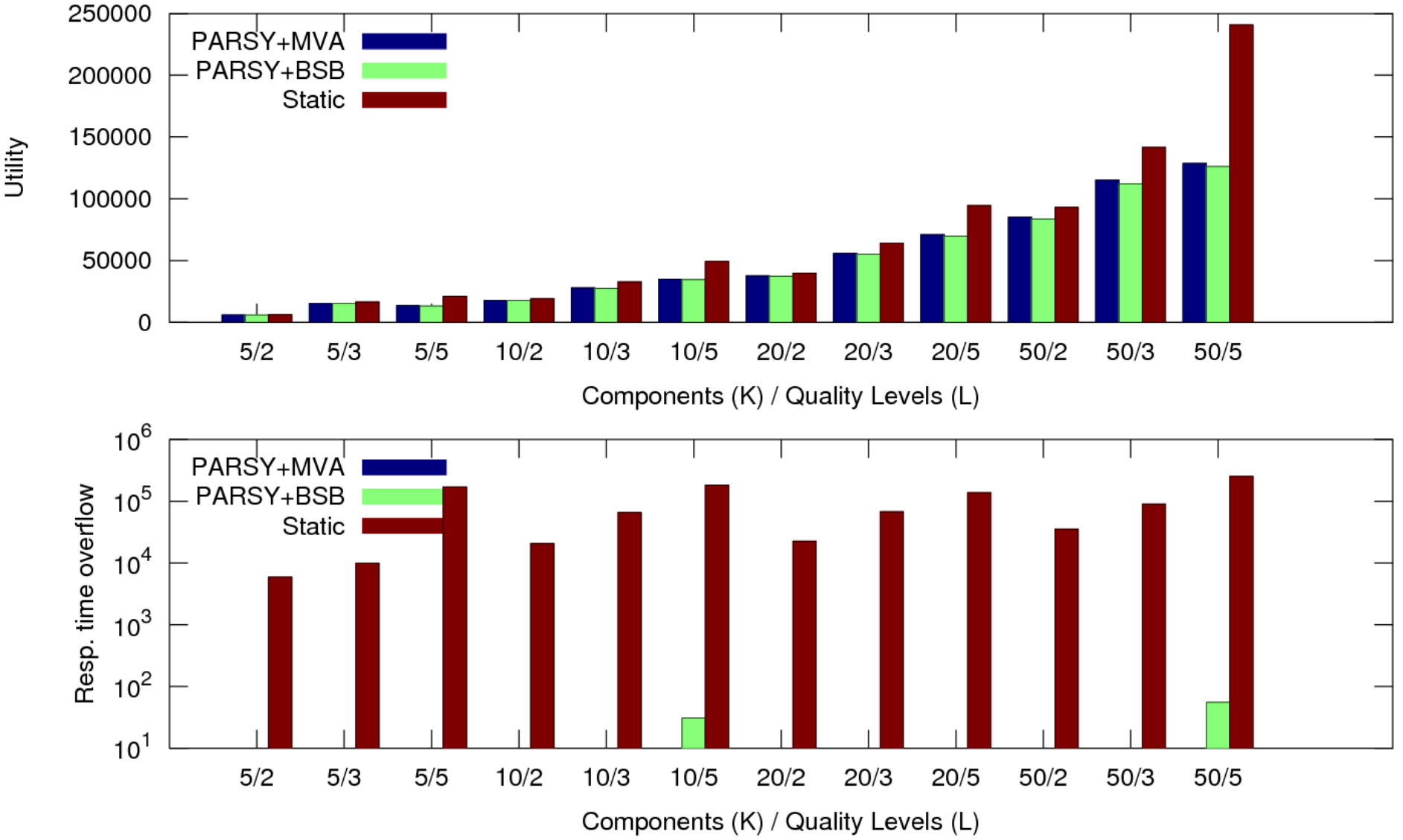
If $\hat{R} < R_{max} \rightarrow$ Upgrade

1. Consider the set S of all components which can be upgraded (those currently operating at level < L_k)
   - If S is empty, stop
2. Select the component k in S with minimum ratio New Demand / New Utility
   - *New Demand = service demand of upgraded component k*
3. Upgrade component k
4. Estimate new system response time R'
   - If $R' > R_{max}$, degrade (rollback) component k, remove k from S and repeat from step 1
   - If $R' \leq R_{max}$, repeat from step 1

Numerical example (K=10, L=2)



Numerical example



Conclusions and future works

- Experimental results are promising
  - PARSY+BSB produces only marginally worse reconfigurations than PARSY+MVA ← *That's a good news*
- TODO
  - What if D(k,j) are not known in advance?
  - What if D(k,j) vary over time?
  - What if multiple components share the same physical resource (CPU)? ← *QN model with multiple job classes?*
  - What if U(k,j) depends on the current quality level of other component(s)?