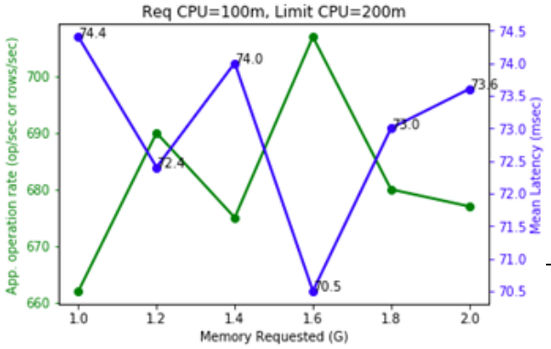
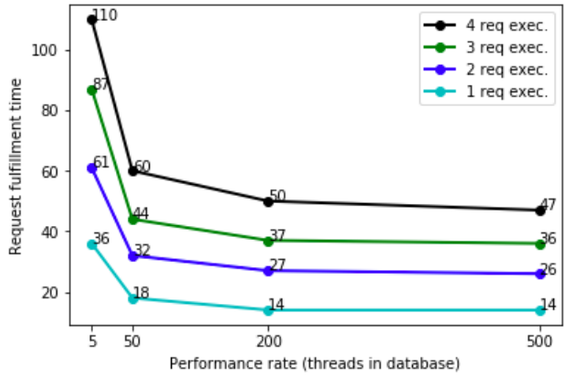
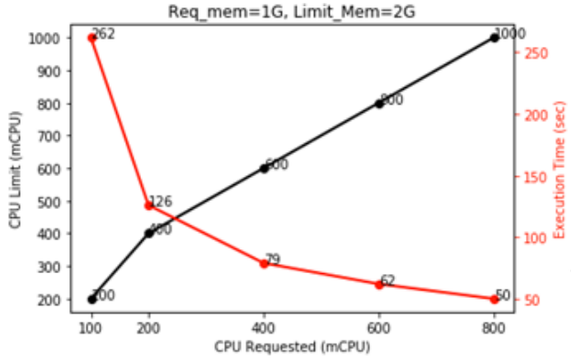
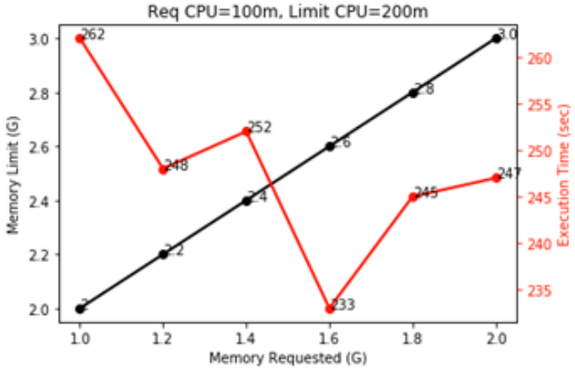
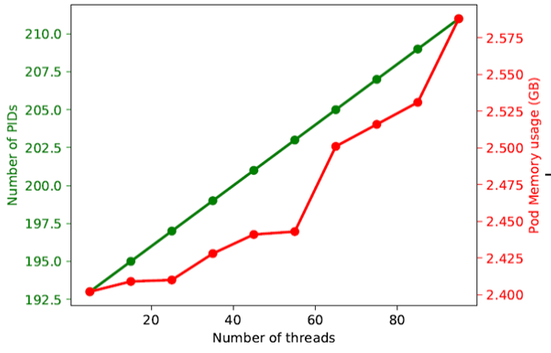
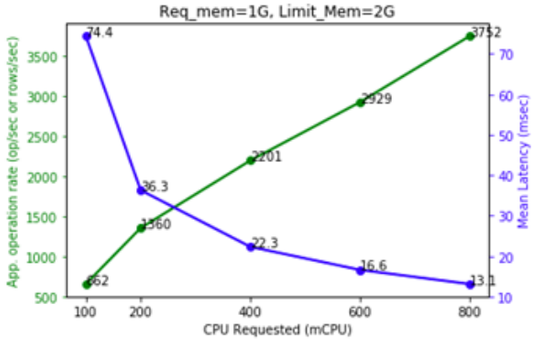












Optimization

Mapping or Regression Model

Output of Regression Model:

- Action selected
- Selection of pod
- Routing of new request

Input of Regression Model:

- Each action probability from ML
- Objective Function
- Constraints

Output: Probability of each action classification

Scaling
Resource Limit
Priority Defining
Disable
Rollback
Restart
Alert

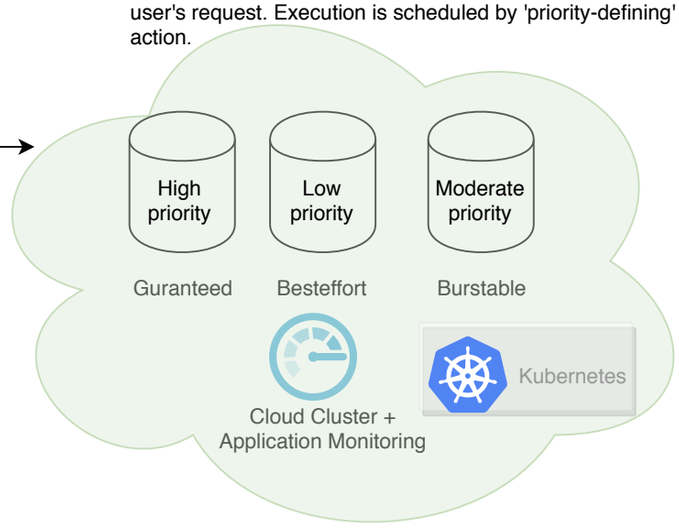
Open source tool LIME calculates such probability

Evolutionary Learning Model (Progressive Learning)

Guranteed: Pod for execution of premium user's request.

Besteffort: Pod for execution of normal user request.

Burstable: Pod for execution of both premium and normal user's request. Execution is scheduled by 'priority-defining' action.



Input features = $\{N_{pod}(t),\ M_{kub}(t),\ N_{pod_L},\ M_{kub_L},\ U_{type},\ r,\ \#\ of\ PID,\ CPU_{usage_{mean}},\ M_{app_{garbage}},\ errors,\ ....\}$

Actions that are independent of user activities: Rollback, Restart, Alert
Actions that do not require any change in application execution in cluster: Disable, Alert

Explanation:

Condier that new user's request arrive and ML prediction probabilities of each actions to fulfill the request are ranked as: scaling, resource-limit, priority defining.

Since scaling has highest probability, the optimization block checks whether scaling up is possible given the constraints. If cluster resources are available then scale the application pod. But resources are not available for scaling then determine the user-type.

If it is premium user then remove the 'best-effort' pod and scale the 'guranteed pod' then execute the request.

If it is normal user then choose the next action in the ML predicted probability ranking which is resource-limit. Mapping or regression model will determine which pod to select and how much performance rate to apply based on the mapping plot it possess.

Guranteed: Pod for execution of premium user's request.

Besteffort: Pod for execution of normal user request.

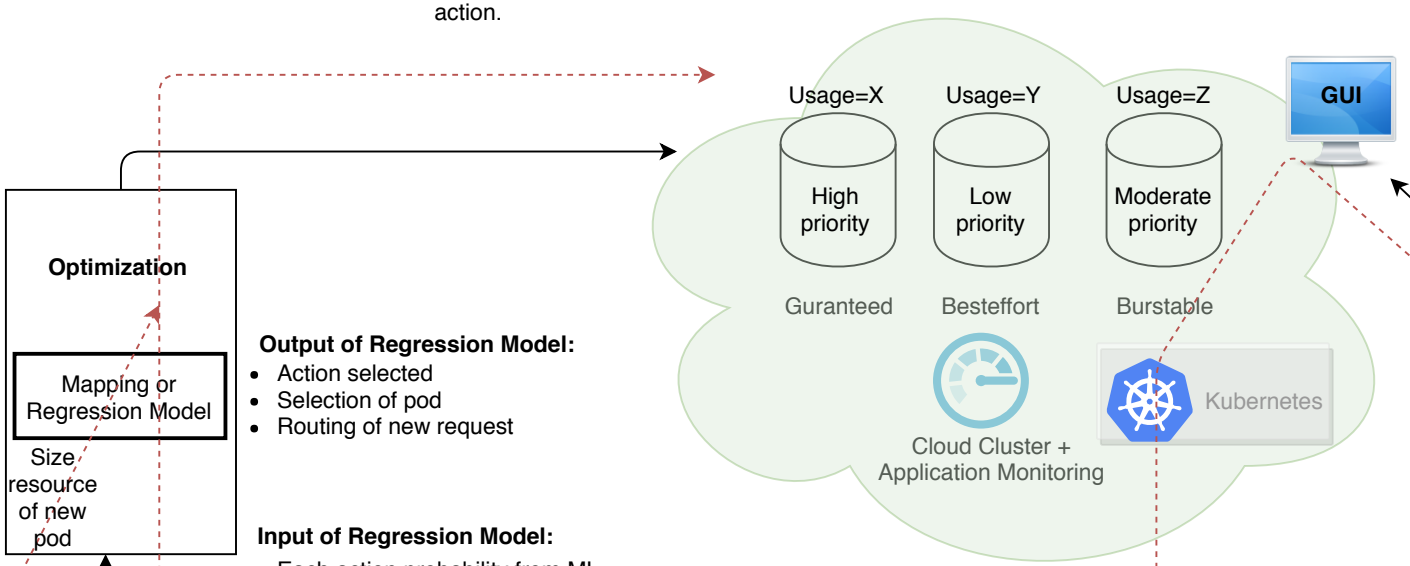
Burstable: Pod for execution of both premium and normal user's request. Execution is scheduled by 'priority-defining' action.

Objective:

1) Requests of premium users ($U_{type} = 1$) are fulfilled in T_{prem} time.
2) Requests of general users ($U_{type} = 0$) are fulfilled in T_{gen} time. ($T_{prem} < T_{gen}$)

Constraints:

1) Total number of application pods $\leq N_{pod_L}$.
2) Total cluster memory usage $\leq M_{kub_L}$.
3) Total cluster CPU usage $\leq C_{kub_L}$



Output of Regression Model:

- Action selected
- Selection of pod
- Routing of new request

Input of Regression Model:

- Each action probability from ML
- Objective Function
- Constraints

Scaling

Output:
Probability of each action classification

Evolutionary Learning Model (Progressive Learning)

Input features = $\{N_{pod}(t),\ M_{kub}(t),\ N_{pod_L},\ M_{kub_L},\ U_{type},\ r,\ \#\ of\ PID,\ CPU_{usage_{mean}},\ M_{app_{garbage}},\ errors,\ ....\}$

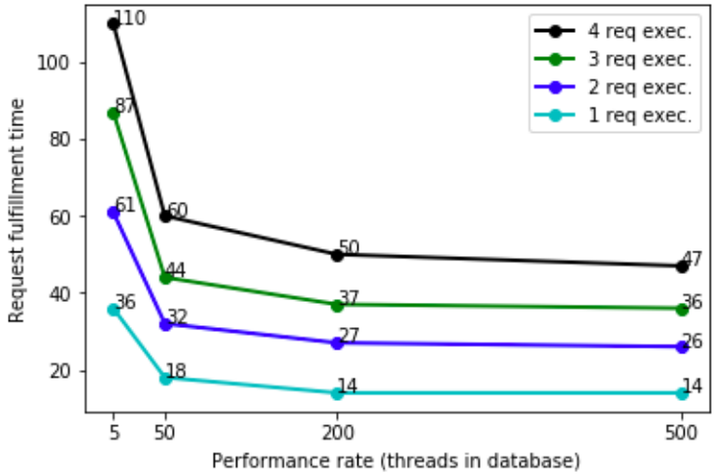
Scaling = a%
Resource Limit = b%
Priority Defining = c%
Disable = d%
Rollback = e%
Restart = f%
Alert = g%

Explanation:

Condier that new user's request arrive. ML will predict probabilities of each actions to fulfill the request. Consider a > b > c >d .....

If Utype = 1, request execute on Guranteed and Burstable.
if $(X_{new} + X + Y + Z)_{CPU} < C_{kub_L}$ || $(X_{new} + X + Y + Z)_M < M_{kub_L}$
Do scaling
else if $(X + Y + Z + Z_{new})_{CPU} < C_{kub_L}$ || $(X + Y + Z + Z_{new})_M < M_{kub_L}$
Do scaling

If Utype = 0, request execute on Besteffort and Burstable
if $(X + Y + Y_{new} + Z)_{CPU} < C_{kub_L}$ || $(X + Y + Y_{new} + Z)_M < M_{kub_L}$
Do scaling
else if $(X + Y + Z + Z_{new})_{CPU} < C_{kub_L}$ || $(X + Y + Z + Z_{new})_M < M_{kub_L}$
Do scaling



Type of pod to select

Guranteed: Pod for execution of premium user's request.

Besteffort: Pod for execution of normal user request.

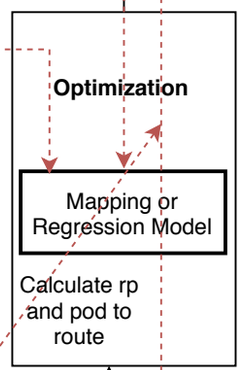
Burstable: Pod for execution of both premium and normal user's request. Execution is scheduled by 'priority-defining' action.

Objective:

1) Requests of premium users ($U_{type} = 1$) are fulfilled in T_{prem} time.
2) Requests of general users ($U_{type} = 0$) are fulfilled in T_{gen} time. ($T_{prem} < T_{gen}$)

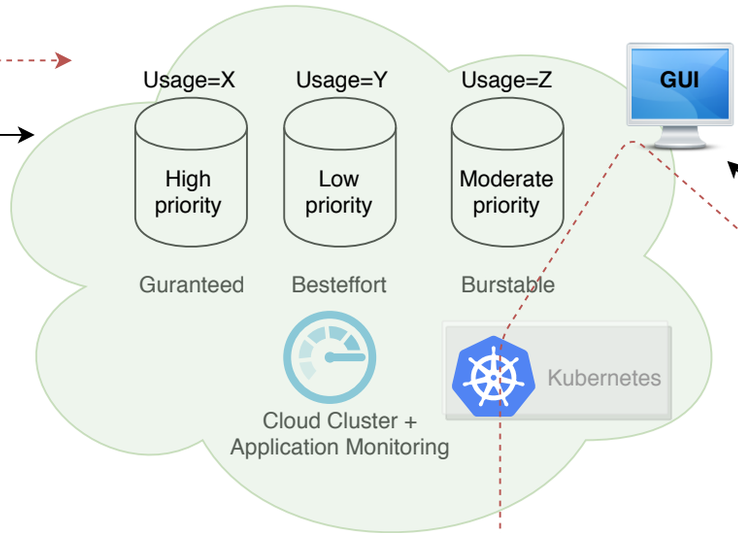
Constraints:

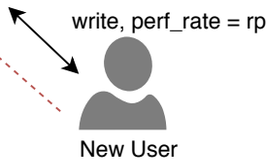
1) Total number of application pods $\leq N_{pod_L}$.
2) Total cluster memory usage $\leq M_{kub_L}$.
3) Total cluster CPU usage $\leq C_{kub_L}$



Output of Regression Model:
- Action selected
- Selection of pod
- Routing of new request

Input of Regression Model:
- Each action probability from ML
- Objective Function
- Constraints





Resource limit

Evolutionary Learning Model (Progressive Learning)

Output:
Probability of each action classification

Input features = $\{N_{pod}(t),\ M_{kub}(t),\ N_{pod_L},\ M_{kub_L},\ U_{type},\ r,\ \#\ of\ PID,\ CPU_{usage_{mean}},\ M_{app_{garbage}},\ errors,\ ....\}$

Scaling = a%
Resource Limit = b%
Priority Defining = c%
Disable = d%
Rollback = e%
Restart = f%
Alert = g%

Explanation:

Consider that ML classification conflicts with constraints. Scaling is not possible. So next action in the probability order is picked.

If Utype = 1, request execute on Guranteed and Burstable.
if $(X_{new} + X + Y + Z)_{CPU} < C_{kub_L}$ || $(X_{new} + X + Y + Z)_M < M_{kub_L}$
Do scaling
else if $(X + Y + Z + Z_{new})_{CPU} < C_{kub_L}$ || $(X + Y + Z + Z_{new})_M < M_{kub_L}$
Do scaling
else
Resource Limit ✓

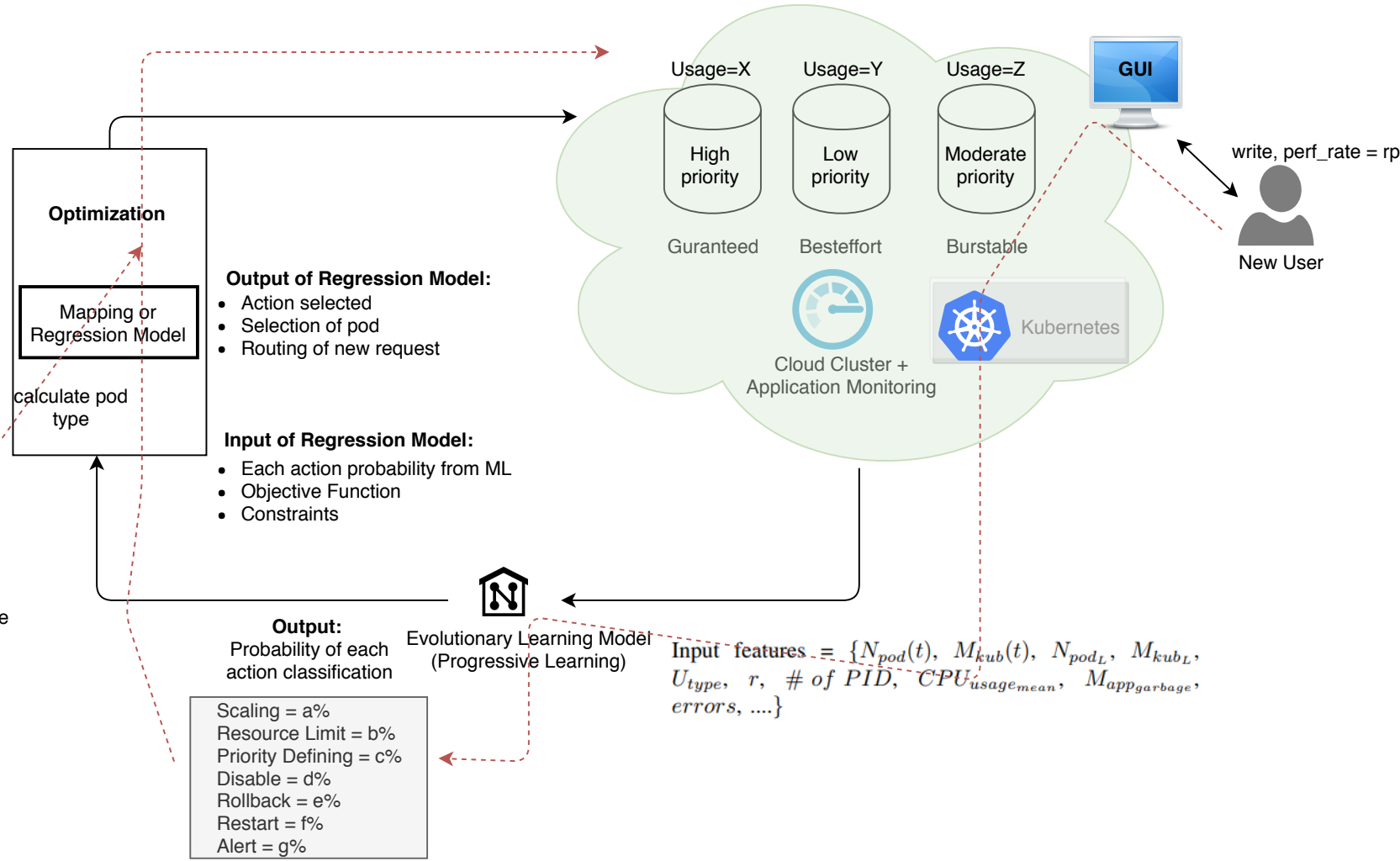
If Utype = 0, request execute on Besteffort and Burstable
if $(X + Y + Y_{new} + Z)_{CPU} < C_{kub_L}$ || $(X + Y + Y_{new} + Z)_M < M_{kub_L}$
Do scaling
else if $(X + Y + Z + Z_{new})_{CPU} < C_{kub_L}$|| $(X + Y + Z + Z_{new})_M < M_{kub_L}$
Do scaling
else
Resource Limit ✓

Objective:

1) Requests of premium users ($U_{type} = 1$) are fulfilled in T_{prem} time.
2) Requests of general users ($U_{type} = 0$) are fulfilled in T_{gen} time. ($T_{prem} < T_{gen}$)

Constraints:

1) Total number of application pods $\leq N_{pod_L}$.
2) Total cluster memory usage $\leq M_{kub_L}$.
3) Total cluster CPU usage $\leq C_{kub_L}$



Explanation:

Consider again that ML classification conflicts with constraints. Even resource limiting is not possible. So next action in the probability order is picked.

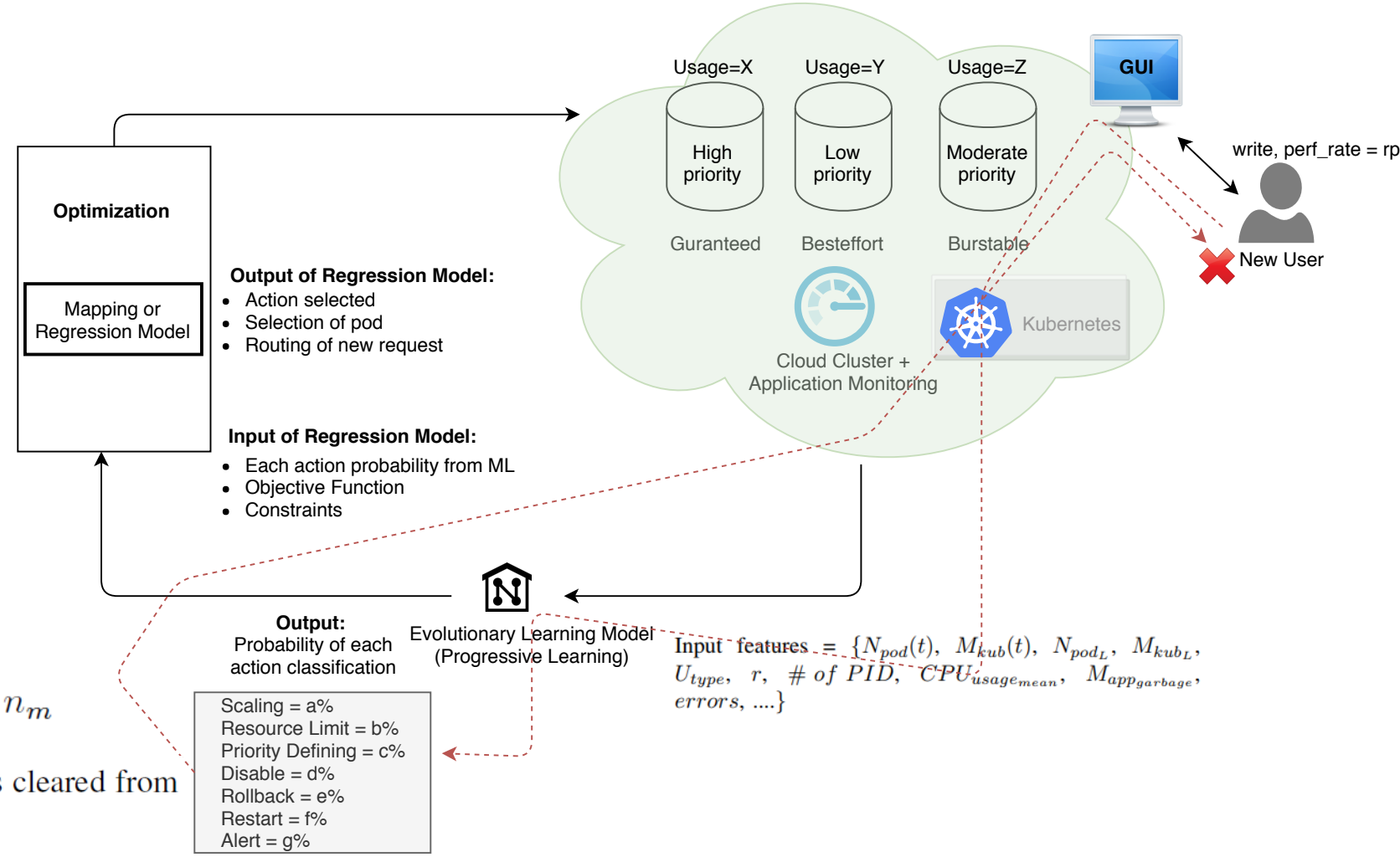
```
If Utype = 1, request execute on Guranteed and Burstable.
   Replace Besteffort with Burstable

While (Utype != 1)
      If Utype = 0, request execute on Besteffort and Burstable
         if len(priority queue == 0)
            Replace Guranteed with Burstable
         else
            Wait until the resource are available
```

Objective:

1) Requests of premium users ($U_{type} = 1$) are fulfilled in T_{prem} time.
2) Requests of general users ($U_{type} = 0$) are fulfilled in T_{gen} time. ($T_{prem} < T_{gen}$)

Constraints:

1) Total number of application pods $\leq N_{pod_L}$.
2) Total cluster memory usage $\leq M_{kub_L}$.
3) Total cluster CPU usage $\leq C_{kub_L}$



Explanation:

If disable action (d) has highest probability or top three actions are not possible then the application doesn't need any change.

Let the threshold for thenumber of violated alert is nm.

$$\lim_{t \to n_m T_{\text{gen}}} U_{\text{gen}}[count(r_p < \bar{r}_p), t] > n_m$$

After $t = n_m T_{\text{gen}}$, the history of user $U_{\text{gen}}$is cleared from the history register.

Objective:

1) Requests of premium users ($U_{type} = 1$) are fulfilled in T_{prem} time.
2) Requests of general users ($U_{type} = 0$) are fulfilled in T_{gen} time. ($T_{prem} < T_{gen}$)

Constraints:

1) Total number of application pods $\leq N_{pod_L}$.
2) Total cluster memory usage $\leq M_{kub_L}$.
3) Total cluster CPU usage $\leq C_{kub_L}$

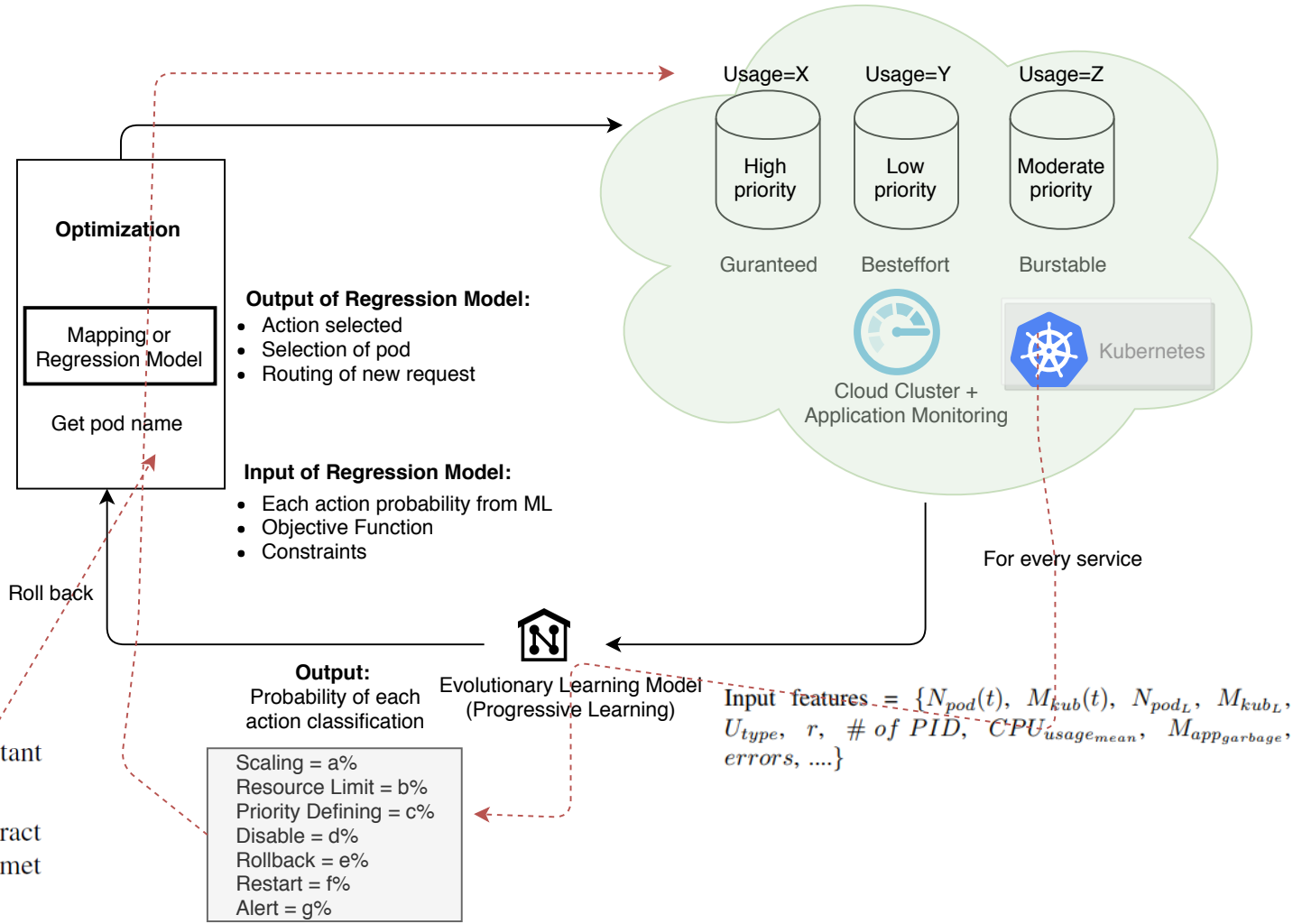
Explanation:

If rollback action (e) has highest probability or top four actions are not possible then a pod is selected to start with base image.

The name of installed packages in a healthy pod is listed at instant $t = 0$ in a variable denoted by P_{k0}.

interval T_r determined by application manager to extract such list. Rollback is actuated if below condition is met for any pod.

$$\lim_{t \to T_r} P_k(t) \neq P_{k0}$$

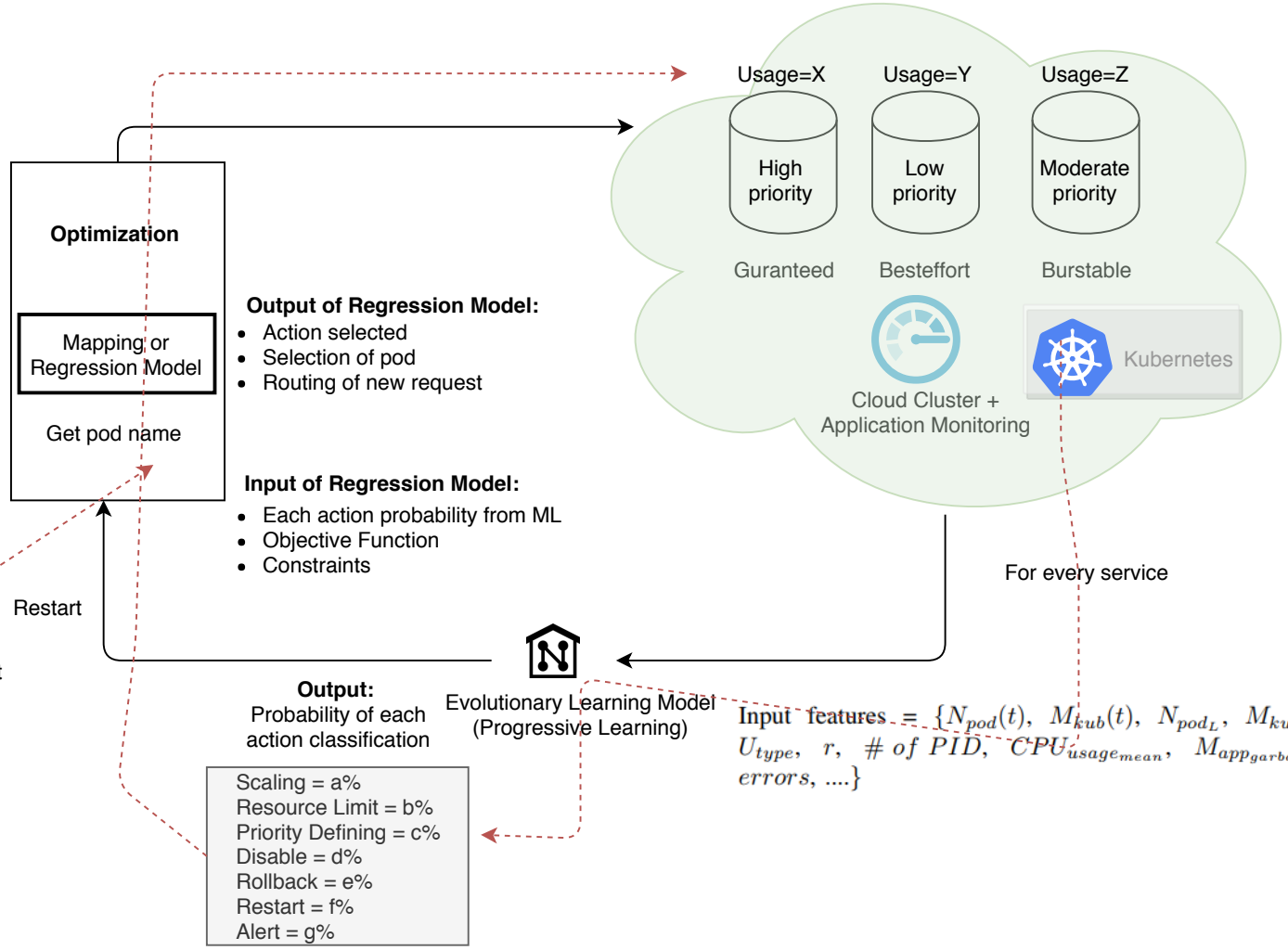


Objective:

1) Requests of premium users ($U_{type} = 1$) are fulfilled in T_{prem} time.
2) Requests of general users ($U_{type} = 0$) are fulfilled in T_{gen} time. ($T_{prem} < T_{gen}$)

Constraints:

1) Total number of application pods $\leq N_{pod_L}$.
2) Total cluster memory usage $\leq M_{kub_L}$.
3) Total cluster CPU usage $\leq C_{kub_L}$



Explanation:

If restart action (f) has highest probability or top five actions are not possible then a pod is selected to restart.

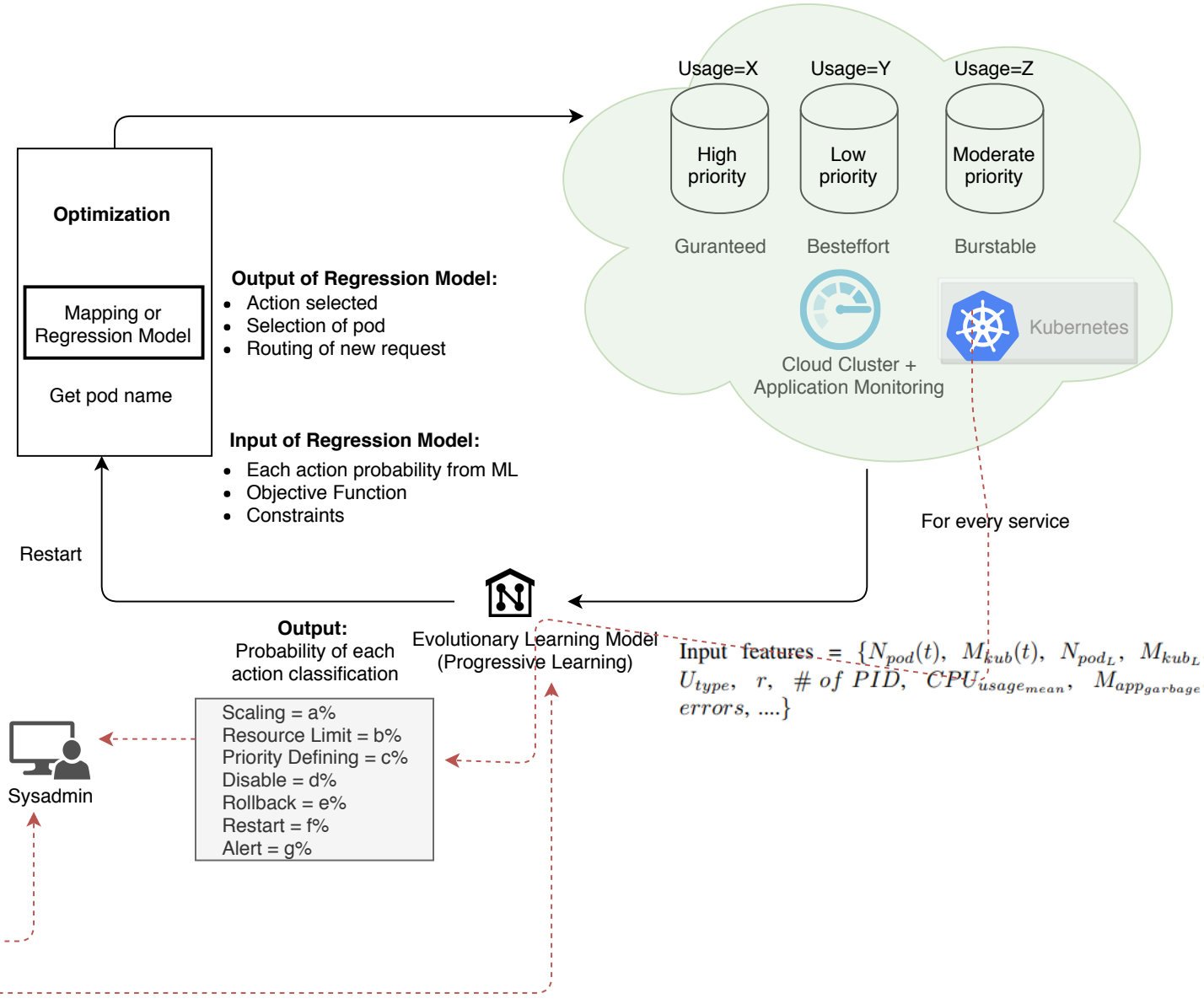
It is activated to refresh the execution of application in a pod. Some of the scenario are application crash, ssh and tcp failures, memory filled with garbage logs, process hang etc. Before restart operation, the pod is removed from the application load balancer.

Objective:

1) Requests of premium users ($U_{type} = 1$) are fulfilled in T_{prem} time.
2) Requests of general users ($U_{type} = 0$) are fulfilled in T_{gen} time. ($T_{prem} < T_{gen}$)

Constraints:

1) Total number of application pods $\leq N_{pod_L}$.
2) Total cluster memory usage $\leq M_{kub_L}$.
3) Total cluster CPU usage $\leq C_{kub_L}$



Explanation:

If alert action (f) has highest probability or top six actions are not possible then an alert is raised with explanation.

unique syscall denoted by $f(S,t)$ is listed by performing various experiments. It represents the application behavior under normal state S_n.

$$S_n = f(S,t)$$

Any new emergence of syscall is considered as a proxy for application behavioral change. In this state S_m, $S_m \not\subset S_n$. An alert is raised for the system administrator with the detailed information regarding emergence of new syscall to perform necessary action. If the system administrator confirms this syscall occurrence as normal then this new syscall is added to the unique syscall list as given below.

$$\tilde{S_n} = S_m \cup f(S,t)$$