5th Seminar – Performance modelling – Solutions

Dimensional analysis. When solving performance modelling exercises it's recommended to perform dimensional analysis[1] known from physics. Let's look at the kinematics equation $s = v_0t + \frac{a}{2}t^2$
With dimensions: $s[\mathrm{m}] = v_0\left[\frac{\mathrm{m}}{\mathrm{s}}\right]t[\mathrm{s}] + \frac{a}{2}\left[\frac{\mathrm{m}}{\mathrm{s}^2}\right]t^2[\mathrm{s}^2] = v_0t[\mathrm{m}] + \frac{a}{2}t^2[\mathrm{m}]$
The main motivation for dimensional analysis is to check whether the dimensions of certain values match or not. If not, then the used formula is probably incorrect.[2] Dimensional analysis often helps us to choose the correct formula. It's important to note that units like "piece", "request", etc. don't have their own dimensions, thus dimensions $\frac{\mathrm{request}}{\mathrm{s}}$ and $\frac{1}{\mathrm{s}}$ are considered the same.
Basic formulas. Little's law: $N = X \cdot T$, $N\ [1] = X\left[\frac{1}{s}\right] \cdot T\ [s]$
Utilization derived intuitively from Little's law in case of *one exclusive resource instance*:
$U = \frac{X}{X^{\max}} = \frac{T_{\text{busy}}}{T_{\text{measured}}} = N = X \cdot T$
Maximum throughput (when the utilization is 100%) derived from processing time in case of *one exclusive resource instance*: $X = \frac{U}{T} \Rightarrow X^{\max} = \frac{1}{T}$

1 Disk's performance

A disk serves 50 requests per second. Each request takes 0,005 seconds to serve. There is no overlap in the system.

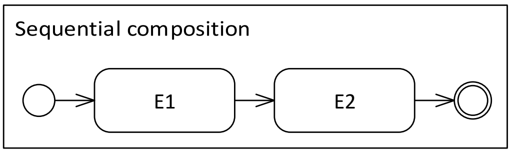
a. What is the utilization of the system?
Solution
The utilization of the resource is $U = X \cdot T$, where X is the average throughput and T is the average response time for a request. $U = 0{,}25$, so the resource utilization is 25%.
The question can also be answered intuitively: the disk has to work 50 request · 0,005 $\frac{\mathrm{s}}{\mathrm{request}}$ second in a second. If the disk works 0,25 s in a second, then the utilization is 25%.

b. What is the maximal workload (arrival rate) that can be served?
Solution
During maximal workload the utilization is $U = 1$ and $X = X^{\max}$, so $X^{\max} = \frac{U}{T}$. The equation for a single resource instance with no overlapping is: $X^{\max} = \frac{1}{T} = \frac{1}{0{,}005\ \mathrm{s}} = 200\ \frac{\mathrm{request}}{\mathrm{s}}$.

2 Inspecting midterm exam evaluation

During the inspection of the exam students have the opportunity to complain about accidental grading mistakes. The final exam result may be modified in case of a justifiable complaint. An exam grader can review 10 exercises in an hour in case of the first exercise (E1) or 20 for the second exercise (E2). Each exercise has its own dedicated exam grader who graded that specific exercise. For each of the following subtasks create a process model describing the given scenario and calculate the maximum throughput (reviewed students per hour) of the inspection process!

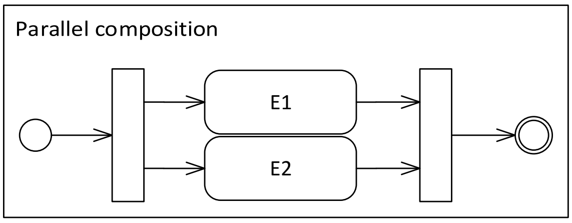
a. Students complain about the first, then the second exercise to the respective graders.
Solution



Sequential composition. The bottleneck (the slowest task) will determine the maximum throughput of the process, because the requests will be queued in front of it (even if the other tasks have a high maximum throughput). Generally: $X^{max} = min(X_1^{max}, X_2^{max})$. Since $X_{E1}^{max} = \frac{10}{h}, X_{E2}^{max} = \frac{20}{h}$, E1 is the bottleneck, so $X^{max} = min(X_{E1}^{max}, X_{E2}^{max}) = min(\frac{10}{h}, \frac{20}{h}) = \frac{10}{h}$.

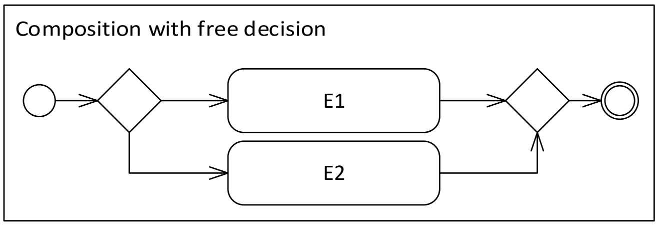
b. The students become resourceful and they give the two exercises to the two graders at the same time, since they were written on different sheets of paper. What are the effects of this parallelization?
Solution

[1] Dimensional analysis (Wikipedia), `https://en.wikipedia.org/wiki/Dimensional_analysis`
[2] Recommended reading: what if? – Droppings, `http://what-if.xkcd.com/11/`



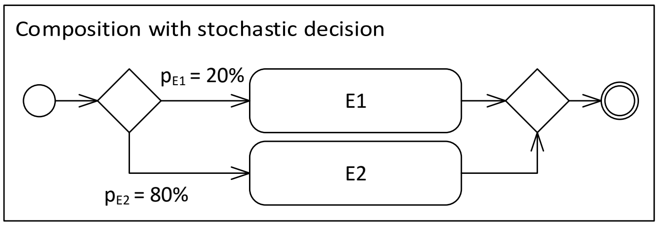
Parallel composition. Since we have to wait for both tasks to end (synchronization) before finishing the process, the bottleneck (the slowest task) will determine the maximum throughput. Generally: $X^{max} = min(X_1^{max}, X_2^{max})$. So $X^{max} = min(X_{E1}^{max}, X_{E2}^{max}) = min(\frac{10}{h}, \frac{20}{h}) = \frac{10}{h}$. What do we gain by this parallelization? The maximum throughput didn't change, but the response time for requests decreased. (It will be the maximum of the response times of the tasks, and not the sum of them, as it would be in the case of a sequence.)

c. Due to the long queue the students complain only about one of the exercises based on which one of the graders is free at the moment.
Solution



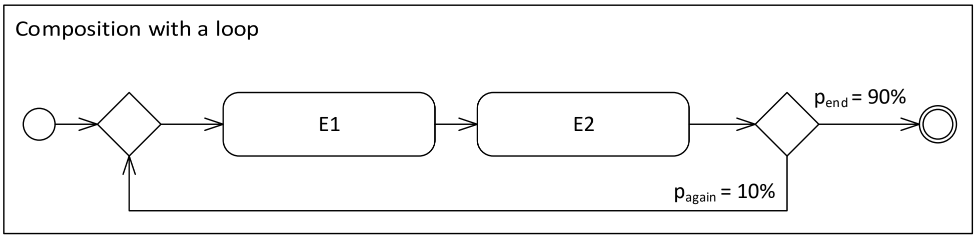
Composition with free decision. The students choose the currently free grader (analogously to K open cash registers in a department store). Generally: $X^{max} = X_1^{max} + X_2^{max}$. So $X^{max} = X_{E1}^{max} + X_{E2}^{max} = \frac{10}{h} + \frac{20}{h} = \frac{30}{h}$

d. Word got around that the grader of the second exercise is less strict, thus 80% of students form a queue in front of the second grader. The remaining 20% of students goes to the grader of the first exercise.
Solution



Composition with stochastic decision. Analogy: user behaviour on a website: 20% chance to buy, 80% chance to discard. Generally: $X^{max} = min(\frac{1}{p_1} \times X_1^{max}, \frac{1}{p_2} \times X_2^{max})$, where p_1 and p_2 are the probabilities of choosing the first and second possibility, respectively ($p_1 + p_2 = 1$). We take the inverse of p_1 because the time spent with E1 is $p_1 \times T_1$, and the corresponding maximum throughput is the inverse of this (assuming one resource instance). The same is true for p_2. So $X^{max} = min(\frac{1}{0{,}2} \times X_{E1}^{max}, \frac{1}{0{,}8} \times X_{E2}^{max}) = min(\frac{50}{h}, \frac{25}{h}) = \frac{25}{h}$.

e. 10% of students only need 1 or 2 points to get the better grade after reviewing their exam so they keep repeating the complaining process. Assume that the complaining process is the same as in subtask a.
Solution



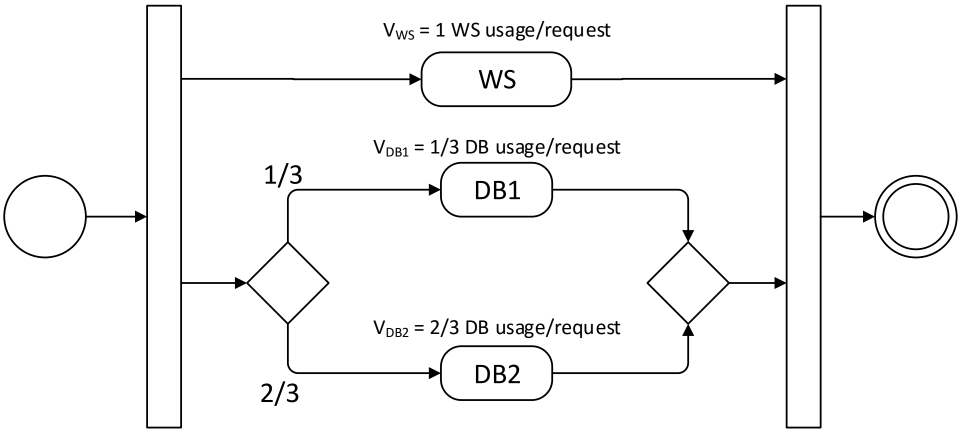
Composition with a loop. Generally: $X^{max} = \frac{1}{\frac{1}{p_{end}}} \times X_1^{max} = p_{end} \times X_1^{max}$, where p_{end} is the probability of exiting the loop and $\frac{1}{p_{end}}$ is the expected number of iterations (see the Probability Theory course).
The value of X_1^{max} is the one we calculated in subtask a. : we used abstraction, so instead of the two tasks we only consider one, that describes the properties of the two tasks together (in this case their maximum throughput with sequential composition). p_{end} is 0,9 in this case. So $X^{max} = \frac{1}{\frac{1}{0{,}9}} \times \frac{10}{h} = 0{,}9 \times \frac{10}{h} = \frac{9}{h}$
This is an approximation compared to real life, because we assumed that the probability of repeating the inspection is independent from the result of the previous try.

Visitation number: the average number of times that a task/subprocess is executed during the execution of the workflow. In case of tasks that are after a decision: the visitation number is the probability of choosing the path that the task is on. In case of tasks that are in a loop: the visitation number is the expected number of iterations. Calculating the maximum throughput based on the visitation number: $X^{max} = \frac{1}{v} \times X_1^{max}$. Calculating the execution time based on the visitation number: $T_{process} = v \times T_{task}$.

3 2-tier architecture

We have a webserver (WS) and a cluster of two database servers (DB1, DB2). We chose between the database servers using a weighted round robin load balancer with ratio 1:2. We use both kinds of resources to serve each request. In the peak period we monitor the system for 30 minutes, during which it serves 9000 requests. The measured busy times are the following: WS – 1350 s CPU time, DB1 – 810 s, DB2 – 1320 s disk IO time.

a. Create a process model based on the above text that models the processing of a request!
Solution


V_{WS} = 1 WS usage/request
WS
V_{DB1} = 1/3 DB usage/request
1/3
DB1
V_{DB2} = 2/3 DB usage/request
DB2
2/3

Since the exercise didn't specify every detail we will assume that serving requests is done in parallel on the different resources. We could use a sequential model instead (the throughput wouldn't change, *but the serving time would*!), but the former one is more general, since the work of the WS could overlap with the work of the DB. In real life the WS is working both before and after the DB call and sometimes even during it. The current model – in lack of detailed specification – aggregates these possibilities and doesn't care about the exact order of execution (abstraction!).

b. What is the current throughput of the servers?
Solution
Reminder: we can use the visitation number for doing transformations between the throughput and maximum throughput of the system and its components (and other things). If we are working with *throughput* then usually we calculate the throughput of the components from the throughput of the system – in this case we have to multiply by the visitation number, since the component has to process a token a number of times equal to its visitation number. If we would like to calculate *maximum throughput*, then we usually calculate the maximum throughput of the system using the (easy to calculate) maximum throughput of the components – in this case we have to divide by the visitation number, since if we have to process a token more than one time, then the total number of tokens the system can serve (before it becomes saturated) decreases.
Let's calculate the current throughput for the system first, and then for the resources! The number of processed requests is $C = 9000$ ("Count"), the time interval of the measure is $T_m =$ 30 min.

- $X_{\text{system}} = \frac{C}{T_m} = \frac{9000 \text{ request}}{30 \text{ min}} = \frac{9000}{1800} \frac{\text{request}}{\text{s}} = 5 \frac{\text{request}}{\text{s}}$
- $X_{\text{WS}} = X_{\text{system}} \cdot v_{\text{WS}} = 5 \frac{\text{request}}{\text{s}} \cdot 1 = 5 \frac{\text{request}}{\text{s}}$
- $X_{\text{DB1}} = X_{\text{system}} \cdot v_{\text{DB1}} = 5 \frac{\text{request}}{\text{s}} \cdot \frac{1}{3} = 1{,}666 \frac{\text{request}}{\text{s}}$
- $X_{\text{DB2}} = X_{\text{system}} \cdot v_{\text{DB2}} = 5 \frac{\text{request}}{\text{s}} \cdot \frac{2}{3} = 3{,}333 \frac{\text{request}}{\text{s}}$

c. How much time does it take for the servers to serve one request?
Solution
For the individual resources (B is the measured "Busy time", and the individual servers processed $C \cdot v_i$ requests):

- $T_{\text{WS}} = \frac{B_{\text{WS}}}{C \cdot v_{\text{WS}}} = \frac{1350 \text{ s}}{9000 \text{ request}} = 0{,}15 \frac{\text{s}}{\text{request}}$
- $T_{\text{DB1}} = \frac{B_{\text{DB1}}}{C \cdot v_{\text{DB1}}} = \frac{810 \text{ s}}{3000 \text{ request}} = 0{,}27 \frac{\text{s}}{\text{request}}$
- $T_{\text{DB2}} = \frac{B_{\text{DB2}}}{C \cdot v_{\text{DB2}}} = \frac{1320 \text{ s}}{6000 \text{ request}} = 0{,}22 \frac{\text{s}}{\text{request}}$

d. What is the maximum throughput of the system?

Solution

The maximum throughput of the system is the biggest throughput where none of the components receive more requests than their maximum throughput. Based on this the following holds for the DB1 path for example:

$$X_{\text{system}} \cdot v_{\text{DB1}} \leq X_{\text{DB1}}^{\text{max}} \Rightarrow X_{\text{system}} \leq \frac{1}{v_{\text{DB1}}} \cdot X_{\text{DB1}}^{\text{max}}.$$

The same is true for DB2 and WS:

$$X_{\text{system}} \leq \frac{1}{v_{\text{DB2}}} \cdot X_{\text{DB2}}^{\text{max}}$$

$$X_{\text{system}} \leq \frac{1}{v_{\text{WS}}} \cdot X_{\text{WS}}^{\text{max}} = X_{\text{WS}}^{\text{max}}. \tag{1}$$

Since the paths to DB1 and DB2 are after a *stochastic decision* (in the long run it's like we divide every work in a 1:2 ratio and forward them to the different paths, so it's kind of between fork-join and free decision [3]), only the minimum of the two calculated number of requests can arrive at the decision point in order to avoid work overload:

$$X_{\text{system}} \leq \min\left(\frac{1}{v_{\text{DB1}}} \cdot X_{\text{DB1}}^{\text{max}}, \frac{1}{v_{\text{DB2}}} \cdot X_{\text{DB2}}^{\text{max}}\right). \tag{2}$$

The fork node forwards the "whole" request on both paths, so the maximum throughput will be the minimum of the number of requests calculated for the WS and for the decision submodel. Based on this and equations 1 and 2 the equation for the system's maximum throughput is:

$$X_{\text{system}}^{\text{max}} = \min\left(X_{\text{WS}}^{max}, \frac{1}{v_{\text{DB1}}} X_{\text{DB1}}^{max}, \frac{1}{v_{\text{DB2}}} X_{\text{DB2}}^{max}\right)$$

So in order to answer the question we need to calculate the maximum throughputs of the components:

- $X_{\text{WS}}^{\text{max}} = \frac{1}{T_{\text{WS}}} = \frac{1}{0{,}15 \ \frac{\text{s}}{\text{request}}} = 6{,}666 \ \frac{\text{request}}{\text{s}}$
- $X_{\text{DB1}}^{\text{max}} = \frac{1}{T_{\text{DB1}}} = \frac{1}{0{,}27 \ \frac{\text{s}}{\text{request}}} = 3{,}704 \ \frac{\text{request}}{\text{s}}$
- $X_{\text{DB2}}^{\text{max}} = \frac{1}{T_{\text{DB2}}} = \frac{1}{0{,}22 \ \frac{\text{s}}{\text{request}}} = 4{,}545 \ \frac{\text{request}}{\text{s}}$

From these the maximum throughput of the system is:
$X_{\text{system}}^{\text{max}} = \min\left(6{,}666 \ \frac{\text{request}}{\text{s}}, 3 \cdot 3{,}704 \ \frac{\text{request}}{\text{s}}, \frac{3}{2} \cdot 4{,}545 \ \frac{\text{request}}{\text{s}}\right) =$
$\min\left(6{,}666 \ \frac{\text{request}}{\text{s}}, 11{,}112 \ \frac{\text{request}}{\text{s}}, 6{,}818 \ \frac{\text{request}}{\text{s}}\right) = X_{\text{WS}}^{\text{max}} = 6{,}666 \ \frac{\text{request}}{\text{s}}.$
It's interesting to observe that the minimum value corresponds to the WS, however the value of DB2 ($6{,}818 \ \frac{\text{car}}{\text{s}}$) is also close. Thus currently the bottleneck is the webserver, although we can only increase the performance of the system in a limited way by adding/upgrading webservers, since DB2 will become the next bottleneck really quick.

e. Why did we use different types of busy times for the two types of resources?

Solution

Because the DB server and WS are both a smaller system in their own, and they have different bottlenecks (HDD for the DB server, and CPU for the WS). In other systems that perform different work there could be other types of bottlenecks, like network link and RAM bandwidth. Notice that this is an abstraction and its goal is to simplify calculations by removing not (or less) relevant information. We use the fact that the relevance of the removed data is much less than the relevance of the kept data (for example: the memory or HDD of the webserver would be a bottleneck much later than its CPU, but we will never reach that point if the system thrashes because of the CPU).

f. What kind of simplifications does the model make?

Solution

We made multiple simplifications, for example:

[3] In case of a free decision we can forward the requests on either paths, so even if one of the paths is saturated, we still can choose the other path (unlike with stochastic decision). So in case of a free decision the maximum throughputs are added together.

- We assumed linear scalability, while a real system's scalability is usually worse (it can even start thrashing).
- We ignored some of the resources of the real system (see the previous task).
- We assumed that we have a perfect load balancing by statically dividing the workload. This is usually not true: the calculated averages are valid in the long run, but in the short run a request with a larger than average execution time for example could saturate the system.

4 Island's traffic network ($*$ previous exam exercise)

The habitants of an island cross the lake around the island every day when they go to work. They can go north on a bridge or south by car ferry. The (in each direction) single-lane bridge is 200 meters long, the speed limit is 60 $\frac{km}{h}$, and the safety distance (tailgating) is 30 meters from tail light to tail light. There are four ferryboats. Each boat takes the island-land-island trip in 15 minutes, and thus together they take at most 800 cars to the land per hour.

a. What is the maximum throughput of the bridge (northwards)?

Solution

Little's law uses throughput and not maximum throughput – but in the special case when the system is saturated they are the same:

- $N = X \cdot T \rightarrow X = \frac{N}{T}$;
- Maximum number of cars on the bridge based on the safety distance and the length of the bridge: $N = \frac{200\ \text{m}}{30\ \text{m/car}} = \frac{20}{3}$ car;
- The time it takes for a car to drive through the bridge based on the speed limit and the length of the bridge: $T = \frac{200\ \text{m}}{60\ \text{km/h}} = \frac{0{,}2\ \text{km}}{60\ \text{km/h}} = \frac{0{,}2}{60}$ h; so
- $X = \frac{20/3}{0{,}2/60} = 2000\ \frac{\text{car}}{\text{h}} = X^{max}$.

b. How many cars can a ferryboat take?

Solution

Using Little's law similarly to the previous task:

- $N = X \cdot T$
- $X = 800\ \frac{\text{car}}{\text{h}}$;
- $T = 15\ \text{min} = 0,25\ \text{h}$;

so $N = 200$, which means that 200 cars are transported at the same time. Since we have 4 ferries: a single ferry can take 50 cars.

c. What is the combined maximum throughput of the two routes leaving the island in the morning rush hours?

Solution

The combined maximum throughput is the summation of the two maximum throughputs. 2000 cars can cross the bridge in one direction in an hour, so the bridge's maximum throughput is 2000 $\frac{\text{car}}{\text{h}}$. The ferries can transport 800 cars in an hour, so the combined maximum throughput is 2800 $\frac{\text{car}}{\text{h}}$ in one direction. If we look at this as a process modelling problem, then it is equivalent to a free choice decision composition.

d. If the highway on the land gets closed due to an accident, and the traffic is diverted through the island (over the bridge, then by ferry), what is the maximum throughput of the diverted path?

Solution

The maximum throughput of the diverted path (using sequential composition):
$X^{max} = \min(X^{max}_{\text{bridge}}, X^{max}_{\text{ferry}}) = 800\ \frac{\text{car}}{\text{h}}$. So the ferry part of the diverted path is the bottleneck.

e. One morning 900 cars left the island by ferry between 7:00 and 8:30. What was the throughput and the utilization of the ferries in this period?

Solution

Throughput: $X = \frac{C}{T} = \frac{900\ \text{car}}{1{,}5\ \text{h}} = 600\ \frac{\text{car}}{\text{h}}$.

Utilization: $U = \frac{X}{X^{\max}} = \frac{600\ \frac{\text{car}}{\text{h}}}{800\ \frac{\text{car}}{\text{h}}} = 0,75 \rightarrow 75\%$.

f. In the above scenario how many cars on average were waiting on the coast at the same time, if the cars arrived to the port in a well-timed manner, on average half minutes before getting in?

Solution

Queueing time for the ferries (using Little's law): $N = X \cdot T = 900\ \frac{\text{car}}{h} \cdot 0,5\ \text{min} = 900\ \frac{\text{car}}{h} \cdot 0,5\ \text{min} \cdot \left[\frac{1\ \text{h}}{60\ \text{min}}\right] = 7{,}5\ \text{car}$.

5 Knowledge base ($*$)

Our company's public professional knowledge base offers articles that may reference each other. On average the server takes 60 ms serving an article request. After reading an article, the reader only leaves the page 30% of the time; most of the cases, they click on a reference to another article.

a. How much server time is needed on average to satisfy a reader's total thirst for knowledge?

   Solution

   Querying an article takes 60 ms, a user checks $v = \frac{1}{0{,}3}$ articles on average[4], where v is the visitation number. From this: $T = 60\ \frac{\text{ms}}{\text{article}} \cdot \frac{1}{0{,}3}\ \frac{\text{article}}{\text{user}} = 200\ \frac{\text{ms}}{\text{user}}$

b. Assume that the requests can't be parallelized. How many unique users can the server serve per hour?

   Solution

   The number of users that the server can serve reaches the maximum when the utilization is a 100%, that is $U = 1$. So $U = X \cdot T \to X = \frac{U}{T} = \frac{1}{0{,}2\ \text{s}} = 5\ \frac{\text{user}}{\text{s}}$. In an hour: $\left[\frac{3600\ \text{s}}{1\ \text{h}}\right] \cdot 5\ \frac{\text{user}}{\text{s}} = 18000\ \frac{\text{user}}{\text{h}}$.

[4] Expected value of geometric distribution (Wikipedia) `https://en.wikipedia.org/wiki/Geometric_distribution`