6th Seminar – Performance Modelling, Visual Data Analysis – Solutions

1 Server's performance

We measured the following performance metrics on a server:

Time of measure [ms]	500	600	700	800	900
Requests processed in the last 100 ms [request]	11	12	21	18	20
Average serving time in the last 100 ms [ms]	15	20	21	25	27
CPU utilization of the last 100 ms [%]	12	13	16	17	19
HDD I/O utilization of the last 100 ms [%]	55	63	87	61	73

a. Based on the available data, which server resource seems to be the bottleneck and why?
Solution
The HDD's utilization is the greatest. If we increase the workload then the HDD will be the first one that becomes saturated (reaches a 100% utilization). (Here we assume a linear scaling of the resources.)

b. What is the server's throughput at the time of the first measure? What is the average and the median of the throughput based on these 5 measurements?
Solution
We can see from the times of the measures that 100 ms elapsed between two measures. From this:

$$X_1 = \frac{r_1}{\Delta t} = \frac{11 \text{ request}}{100 \text{ ms}} = \frac{11 \text{ request}}{100 \text{ ms}} \left[\frac{1000 \text{ ms}}{1 \text{ s}}\right] = 110 \frac{\text{request}}{\text{s}}.$$

We can calculate the average throughput by either calculating the 4 other throughputs or by using the observation that Δt is always 100 ms (and using the average number of requests):

$$\bar{r} = \frac{\sum_{i=1}^{n} r_i}{n} = \frac{11 + 12 + 21 + 18 + 20}{5} = 16{,}4$$

From this the average throughput is $\bar{X} = \frac{\bar{r}}{\Delta t} = \frac{16{,}4}{0{,}1} = 164 \frac{\text{request}}{\text{s}}$.
Median: When we sort the data (11, 12, 18, 20, 21), it is obvious that 18 is the middle element, thus 18 is the median of the data set. The median of the throughput is $\frac{18}{0{,}1} = 180 \frac{\text{request}}{\text{s}}$.

c. What estimate can we provide for the average number of requests being served at the same time based on these 5 measurements?
Solution
We can calculate it from the number of requests processed in the last 100 ms and from the average serving time. Since the average serving time was calculated based on datasets with different number of elements, we can't just calculate a simple average time for the 5 measures. We have to use the weighted average (the weights will be the number of requests for the appropriate average serving times):

$$\bar{T} = \frac{\sum_{i=1}^{n} r_i t_i}{\sum_{i=1}^{n} r_i} = \frac{11 \cdot 15 + 12 \cdot 20 + 21 \cdot 21 + 18 \cdot 25 + 20 \cdot 27}{11 + 12 + 21 + 18 + 20} = 22{,}39 \text{ ms}$$

The system is in a stable state, so we can use Little's law using the average throughput calculated in subtask b.

$$N = \bar{X} \cdot \bar{T} = 164 \frac{1}{\text{s}} \cdot 22{,}39 \text{ ms} = 164 \frac{1}{\text{s}} \cdot 0{,}02239 \text{ s} = 3{,}67196$$

2 Social website

We operate a social web company. Due to its recent rising popularity, response times have increased greatly. The business goal is to have 1500 simultaneous user requests served with less than 4 seconds of response time in average.

a. What minimal throughput should the service infrastructure be designed for, if delays outside our infrastructure (network traffic latency, HTML rendering on the client side) can be estimated as 1 second?
Solution
The infrastructure must serve 1500 users with an average of 3 seconds response time. Using Little's law: $N = 1500$, $T = 3\ \frac{\text{s}}{\text{request}}$, so $X = \frac{N}{T} = 500\ \frac{\text{request}}{\text{s}}$

b. According to measurements, an average user request in the redesigned web site takes 20 ms CPU time on the web server, and occupies the database server for 12,5 ms. Currently we have 15 web servers to handle the requests, while the database is replicated to 5 machines. Assuming linear scalability, how much additional units of each kind of server should we buy to meet the above goal?
Solution
$T_{\text{CPU}} = 20$ ms $= 0{,}02$ s, $T_{\text{DB}} = 12{,}5$ ms $= 0{,}0125$ s. The CPU and the database must be able to serve at least 500 requests per second in order for the system to do the same (using either sequential or parallel composition). Currently for one instance of the resources: $X^{\max}_{\text{CPU}} = \frac{1}{T_{\text{CPU}}} = 50\ \frac{\text{request}}{\text{s}}$, $X^{\max}_{\text{DB}} = \frac{1}{T_{\text{DB}}} = 80\ \frac{\text{request}}{\text{s}}$. The total maximum throughput of 15 webservers is $750\ \frac{\text{request}}{\text{s}}$, while the for the 5 database server it is only $400\ \frac{\text{request}}{\text{s}}$. So we *need 2 more database servers* in order to reach the desired maximum throughput. (One new database server would increase the maximum throughput to $480\ \frac{\text{request}}{\text{s}}$ only, which is less than the required $500\ \frac{\text{request}}{\text{s}}$.)

c. $(*)$ Calculate the utilization of each kind of server in the extended system. If the goal is to push the average utilization of the servers below 50% even during peak hours, do we need to scale out further?
Solution
The maximum throughput of 15 webservers is $X^{\max}_{\text{web}} = 750\ \frac{\text{request}}{\text{s}}$, the required throughput during peak hours is $X_{\text{web}} = 500\ \frac{\text{request}}{\text{s}}$. So their utilizations are $U_{\text{web}} = \frac{X_{\text{web}}}{X^{\max}_{\text{web}}} = \frac{2}{3}$. With the same method: $U_{\text{DB}} = \frac{X_{\text{DB}}}{X^{\max}_{\text{DB}}} = \frac{500}{560} = 0{,}89$.
If we would like to have a 50% utilization ($U = \frac{X_{\text{web}}}{X^{\max}_{\text{web}}}$), then $50\% = \frac{500\ \frac{\text{request}}{\text{s}}}{X^{\max}_{\text{web}}}$ gives $X^{\max}_{\text{web}} = \frac{500\ \frac{\text{request}}{\text{s}}}{0.5} = 1000\frac{\text{request}}{\text{s}}$. For this we need 20 webservers and 13 database servers.

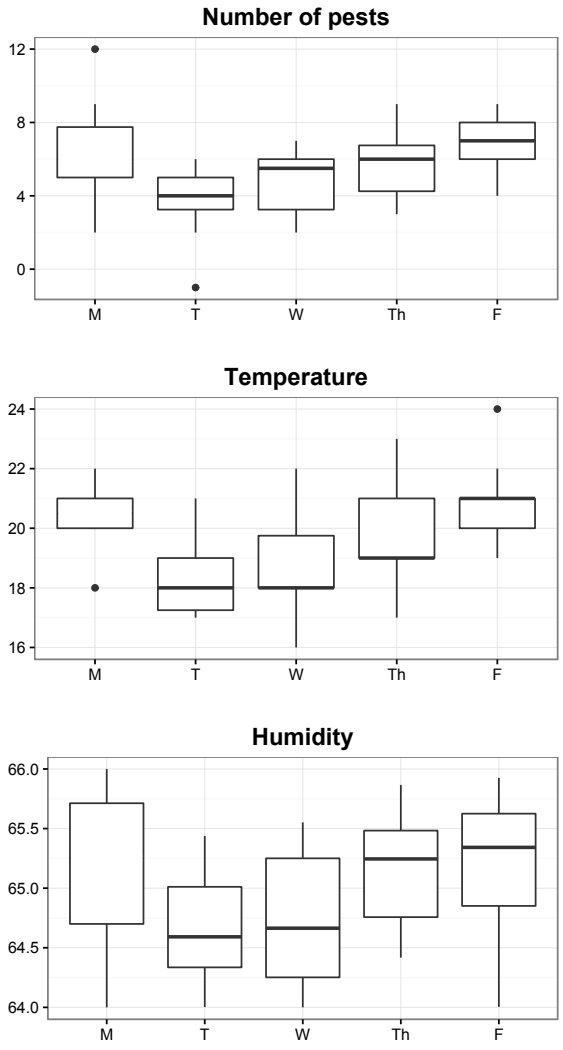
d. Let's consider only 2 webservers and 3 database servers. Create state-based models about the resources in the infrastructure that model the availability of the resources (available or in use). What design decisions do we face? What are the pros and cons of the choices?
Solution
Design choices:
- We model the resources *aggregated by their types* based on how many of a certain resource type is in use. This way we will have a 0–1–2 state chain for the webservers and a 0–1–2–3 state chain for the database servers (with the appropriate state transitions between the states). We can get the complete model of the resource pool by taking the asynchronous product of the two state machines.
  The *advantage* of this solution is that it is really simple. We can also easily model the resource allocation of tasks: if the state machine of the required resource is not in its last state (which means none is available), then the allocation is successful and both the task's state machine (got the resource) and the resource's state machine (one less resource instance is available) change states (synchronization). Releasing the allocated resources is done in the same manner.
  The *disadvantage* of this solution is that it doesn't contain any information about the availability of the individual resource instances (except in the all-free and all-used states). Because of this we can't calculate exact utilization values for the individual resource instances, we can only calculate an average value that describes every resource instance.
- We model *every resource instance individually* with a free/taken state pair (or in even more detail). So we will have as many state machines as resource instances in the system. We can get the complete model of the resource pool by taking the asynchronous product of these state machines.
  The *advantage* of this solution is that for example we can calculate some metrics (like utilization) for the individual resource instances. Or something more interesting: we can

model the failure and repair process of the individual resource instances in order to analyse the effects of resource failures on system level metrics. The properties of the failure and repair events (like the rate of the event) can be different for every instance, so we are able to model a heterogeneous resource pool (for example webservers of multiple brand) or the degradation of resource instances.
The *disadvantage* of this solution is that the consumers of resources see more than one resource instance, so it's harder to model resource allocation. In order to allocate a resource instance we have to find a free one first, and after using it we have to free exactly that same one. This process is even more complicated if we need more than one type/instance of resource to perform a task (possibility of deadlocks and starvation). In this case it is preferable (and a used practice) to introduce a resource management component that hides these complications from the consumers.

3 Sensor network (previous exam exercise) – data analysis

We have an agriculture sensor network that helps us to track the states of our open-field, glasshouse and foil tent areas based on some measured values (temperature, humidity, luminous intensity, wind speed, detected pests, etc.).

Date	Temp. [°C]	Hum. [%]	Pests [piece]
2015. 05. 04. 08:00	18	66,00	3
2015. 05. 04. 09:00	20	65,75	6
2015. 05. 04. 10:00	20	65,75	8
2015. 05. 04. 11:00	20	65,50	9
2015. 05. 04. 12:00	20	65,50	5
2015. 05. 04. 13:00	21	65,00	12
2015. 05. 04. 14:00	21	64,70	5
2015. 05. 04. 15:00	21	64,70	6
2015. 05. 04. 16:00	21	64,60	7
2015. 05. 04. 17:00	22	64,00	2



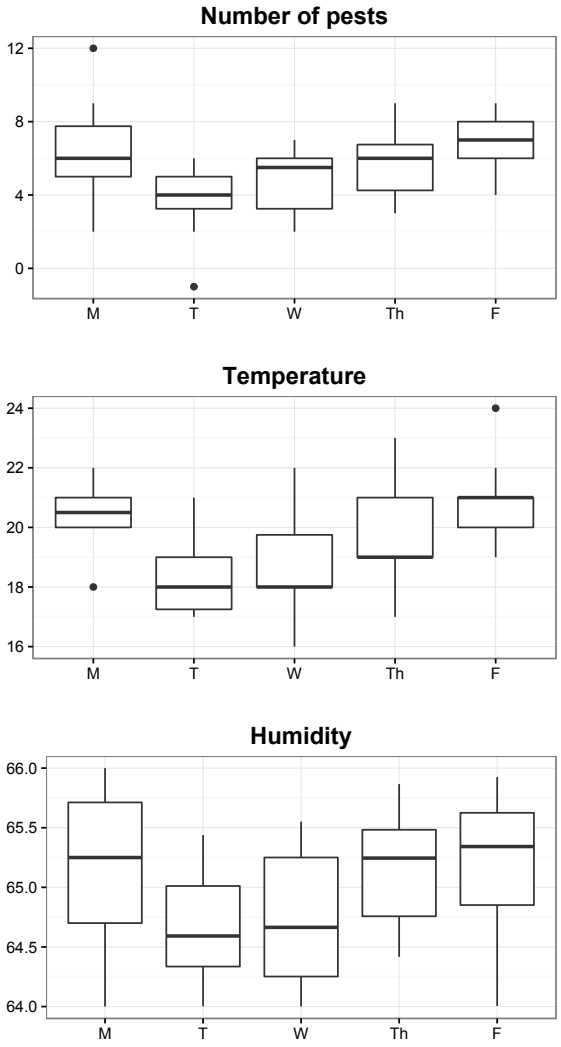
a. Unfortunately the middle values (median) of Monday, May 4th are missing from the figures. Draw them based on the data in the table!
b. Interpret the diagrams: which variable's/variables' first quartile is strictly monotonic in time?
c. (Extra task.) We would like to compare the temperature values and pest numbers of Monday in a parallel coordinates diagram.

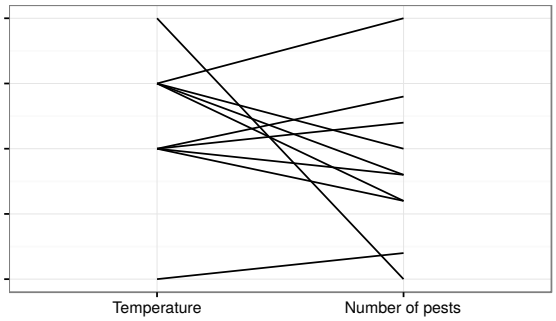
Solution

a. Lets draw the medians on the boxplots. Since we have an even number of values, the median will be the average of the two middle values. The first two column is ordered, so the medians will be: $\frac{20+21}{2} = 20{,}5$ and $\frac{65{,}5+65}{2} = 65{,}25$. After sorting the third column: $2, 3, 5, 5, 6, 6, 7, 8, 9, 12$, the median is $\frac{6+6}{2} = 6$.

b. None of the variables has this property, since none of the bottoms of the "boxes" show a strictly monotonic change.
The main properties of the boxplot diagram are shown on Figure 1. If a value is outside $\pm 1.5 \times \text{IQR}$ then we display it as a dot.
It's interesting to note that using the constant 1,5 is a statistical convention, which is analogous to the $\pm 3\sigma$ principle of the datasets with normal distribution (see Probability Theory course).

c. The values are displayed in the following parallel coordinates diagram. Looking at the parallel coordinates diagram, we see no strong correlation between the two variables.





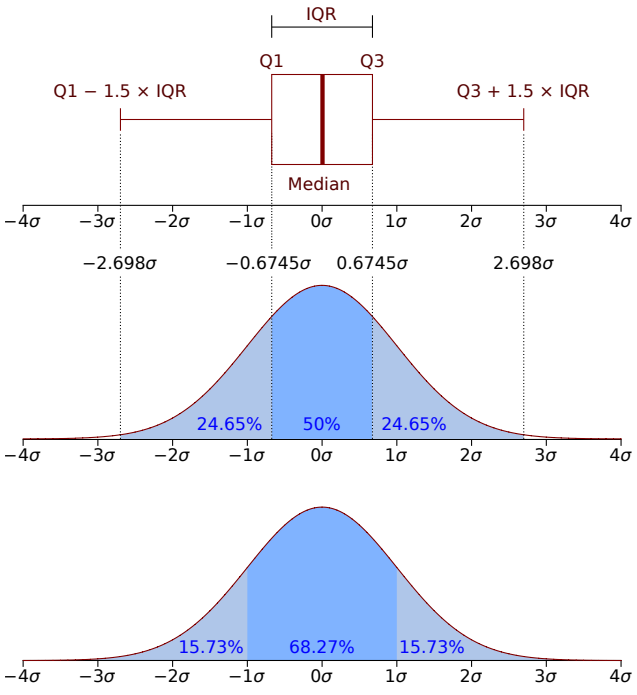


Figure 1: The main properties of the boxplot

4 Sensor network (previous exam exercise) – perf. analysis (∗)

(Performance analysis exercises related to Exercise 3.) The different types of sensors provide data from a 100 meters radius around their location. The sensors forward their timestamped data to the central server through a radio communication-based network. The central server processes the requests then archives them to a storage unit. Our organization installed 4500 sensors and each one sends one measurement data in every minute. The system can successfully handle this load. The radio communication network can forward 100 measurement data every second. The central server's CPU is idle (not doing anything) in 75% of the time. Writing a measurement data to the storage unit takes 8 ms.

a. How many measurement data in a second is the current throughput of the system?
Solution
Little's law: $X = 4500 \cdot \frac{1 \text{ data}}{60 \text{ s}} = 75 \ \frac{\text{data}}{\text{s}}$

b. What is the throughput, maximum throughput and utilization of the radio network, CPU and storage?
Solution
$X_{\text{network}} = X_{\text{CPU}} = X_{\text{storage}} = X = 75 \ \frac{\text{data}}{\text{s}}$, since every visitation number is 1.
$X^{\max}_{\text{network}} = 100 \ \frac{\text{data}}{\text{s}}$ is given in the text $\rightarrow U_{\text{network}} = \frac{X_{\text{network}}}{X^{\max}_{\text{network}}} = 75/100 = 0{,}75 \rightarrow 75\%$
$U_{\text{CPU}} = 1 - 0{,}75 = 0{,}25 = 25\%$ is also given $\rightarrow X^{\max}_{\text{CPU}} = \frac{X_{\text{CPU}}}{U_{\text{CPU}}} = 75/0{,}25 \ \frac{\text{data}}{\text{s}} = 300 \ \frac{\text{data}}{\text{s}}$
$T_{\text{storage}} = 0{,}008$ s and there is no overlap:
$X^{\max}_{\text{storage}} = \frac{1}{T_{\text{storage}}} = 125 \ \frac{\text{data}}{\text{s}} \rightarrow U_{\text{storage}} = \frac{X_{\text{storage}}}{X^{\max}_{\text{storage}}} = 75/125 = 0{,}6 \rightarrow 60\%$

c. How many more sensors can we install (to improve the measurement accuracy) without upgrading our infrastructure? Assume linear scaling!
Solution
Since we use all three resources to process the requests:

$$X^{\max} = \min(X^{\max}_{\text{network}}, X^{\max}_{\text{CPU}}, X^{\max}_{\text{storage}}) = X^{\max}_{\text{network}} = 100 \ \frac{\text{data}}{\text{s}}$$

Since the current throughput is $75 \ \frac{\text{data}}{\text{s}}$, a $4:3$ ratio scaling is possible, which means an additional 1500 sensors.

d. The radio network uses smart encoding, so more than one sensor can forward data at the same time. How many sensors are forwarding data at the same time (overlapping) over the network currently and during maximum load, if a forwarding takes 40 ms?
Solution
We can use Little's law to calculate the number of messages that are transmitted at the same time:

$$T_{\text{network}} = 0{,}040 \text{ s}$$

$$X_{\text{network}} = 75 \ \frac{\text{data}}{\text{s}} \rightarrow N_{\text{network}} = X_{\text{network}} \cdot T_{\text{network}} = 75 \ \frac{\text{data}}{\text{s}} \cdot 0{,}04 \text{ s} = 3 \text{ messages transmitting currently}$$

$$X^{\max}_{\text{network}} = 100 \ \frac{\text{data}}{\text{s}} \rightarrow N_{\text{network}} = 100 \ \frac{\text{data}}{\text{s}} \cdot 0{,}04 \text{ s} = 4 \text{ messages transmitting maximum}$$