

The concept of a bottleneck

Class handout

[As class starts, Radu passes out a handout with the following case and asks the students to read it out loud as a group, each taking turns to read. Radu gives students some time to ponder and discuss among themselves in smaller groups.]

Case: Paula and George's Pizza

Paula Reyes has run a small takeout pizza shop near the university for three years. Her shop does one thing: it sells uncooked pizzas—fresh dough with fresh toppings—for customers to take home and bake themselves. The menu is short, the operation is simple, and Paula has built a loyal following among students and faculty who appreciate the quality and the price.

Making an uncooked pizza involves two steps, done in sequence. First, the dough must be made and tossed into shape, a task that requires both hands, some skill, and cannot be rushed. Second, the pizza is topped with sauce, cheese, and fresh ingredients. The two steps must happen in order: you cannot top a pizza before the dough is ready.

For the first two years, Paula did both steps herself. She timed herself carefully and found that, working at her fastest sustainable pace, she took 5 minutes to make and toss the dough and 5 minutes to prepare toppings and top the pizza. It takes 10 minutes in total to make per pizza.

Last year, Paula brought in help. Her uncle George, a retired restaurant worker with years of experience in pizza kitchens, joined the operation part-time. After a few weeks of working together and timing themselves carefully, Paula and George established the following:

Step	Paula's time	George's time
Make dough	5 min	4 min
Top pizza	5 min	3 min
Both steps	10 min	7 min

Since the shop is busy and demand consistently exceeds what they can produce during peak hours, Paula wants to organize the two of them in a way that maximizes how many pizzas they can turn out per hour. She is considering two arrangements:

1. Paula handles the first step, passes the dough to George, who does the topping. They work in sequence on successive orders.
2. George handles the first step, Paula does the topping.

Question for consideration:

Which arrangement is better for Paula and George?

Class discussion

RADU: [After students finish reading.] So, which arrangement is better?

HAOYU: Neither. George should do everything. He is faster at both steps. Seven minutes versus Paula's ten. You are not going to get more pizzas per hour by using Paula. She should do social media marketing or something with her time.

RADU: That is a bold claim. Let's slow down on that for a moment. Why do you say seven minutes versus ten?

HAOYU: George does the dough in four minutes and the topping in three minutes. That is seven minutes total. Paula does each step in five minutes, so ten minutes total. Faster at both steps means more pizzas per hour, no?

RADU: Ming, does that reasoning convince you?

MING: On the surface, yes. George is faster. So George should do more.

RADU: Camila, you look less convinced.

CAMILA: I am not convinced at all. If George does everything alone, he makes one pizza every seven minutes, which is about 8.6 pizzas per hour. But we have two people. Surely two people can do more than one.

HAOYU: Not if the second person is slower at everything.

CAMILA: Slower at each step, yes. But she adds a second pair of hands. While George is topping one pizza, Paula could already be making the dough for the next one. They do not have to wait for each other to finish before starting the next order.

HAOYU: [Pauses.] Oh, I see, of course ...what was I thinking? [Trails off].

RADU: Farah, I know you want to compute something to make this discussion more precise.

FARAH: [Already writing.] Yes. Let me take the two arrangements from the case. In arrangement one, Paula makes the dough and George does the topping.

RADU: And how do you think about the capacity of that arrangement?

FARAH: Each of them has their own capacity. Paula takes 5 minutes per dough, so her capacity is $60/5 = 12$ pizzas per hour. George tops in 3 minutes, so his capacity is $60/3 = 20$ pizzas per hour.

[Radu writes on the board as Farah talks.]

$$\begin{aligned} C_{\text{Paula}} &= \frac{1 \text{ pizza}}{5 \text{ minutes}} \cdot \frac{60 \text{ minute}}{\text{hour}} = 12 \text{ pizzas per hour} \\ C_{\text{George}} &= \frac{1 \text{ pizza}}{3 \text{ minutes}} \cdot \frac{60 \text{ minute}}{\text{hour}} = 20 \text{ pizzas per hour} \end{aligned}$$

RADU: Okay, and so what is the capacity of the two of them working together?

FARAH: In a serial system, the capacity is the *minimum* of the two.

[The other students stay silent for a moment and look a little puzzled.]

MING: Wait ...why the minimum of C_{Paula} and C_{George} ? Why not the average? Or why not add them somehow? The minimum!?

FARAH: Because the steps have to happen in order. If George can top 20 pizzas per hour but Paula supplies dough at 12 per hour, George will be waiting for dough some of the time.

MING: But what if George goes ahead and tops something else while he waits? Can't there be some extra dough hanging around sometimes?

FARAH: No, there is no extra dough. The only thing George can do is top pizzas, and the only pizzas he can top are the ones Paula has already made.

RADU: Exactly, we say these two resources are working in *series*. Each step depends on the output of the step before it.

MING: [*Slowly.*] So George is stuck waiting for Paula, even though he is faster.

RADU: Exactly. We say that George is *starved*: he is able to do more work, but he sometimes has nothing to work on. And what does that tell us about the overall capacity of the first arrangement?

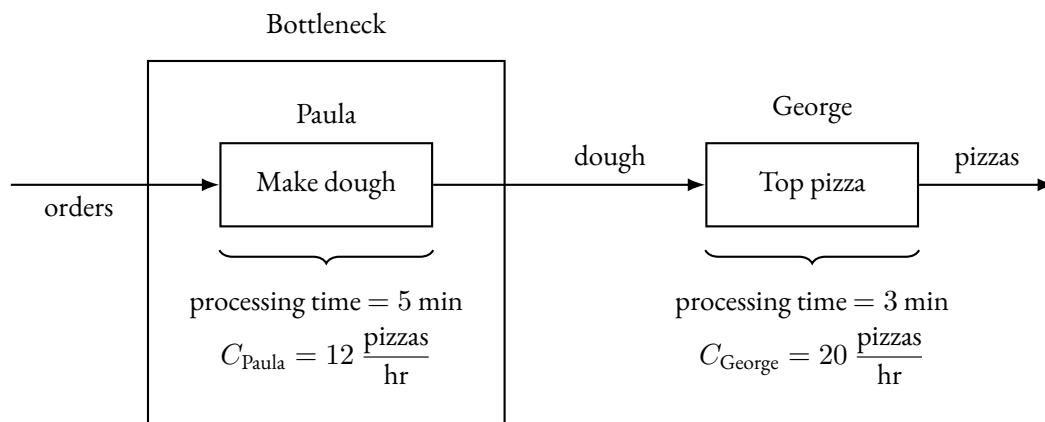
MING: The capacity is limited by Paula. Not George.

FARAH: Which is 12 pizzas per hour.

[Radu writes on the board as Farah answers.]

$$\begin{aligned} C_{\text{arrangement 1}} &= \min\{C_{\text{Paula}}, C_{\text{George}}\} \\ &= \min\{12 \text{ pizzas per hour}, 20 \text{ pizzas per hour}\} \\ &= 12 \text{ pizzas per hour} \end{aligned}$$

RADU: We call Paula the *bottleneck*. She is the resource whose individual capacity determines the capacity of the system. Let's look at the picture here. [*Draws on the board.*]



MING: Why use the word “bottleneck”?

RADU: You don't like the name?

MING: It's not that. Well ...[*Pauses.*] So a bottleneck is the part of a bottle that slims down and limits the flow. [*Gestures an upside-down bottle opening toward her mouth, miming a tapping on the back of the bottle.*] I see, the bottleneck limits the flow.

RADU: Exactly, that is how I think about it. [*Pauses.*] Earlier, you said George should do everything because he is faster at both steps. You are okay with this logic we have developed here.

HAOYU: [*Sheepishly.*] I am actually a little embarrassed about what I said earlier, given this is another restaurant example and I blundered it. Of course George alone is slower than the two of them working together.

RADU: No need to be embarrassed, Haoyu. Sometimes we have real-life experience, and transferring that experience to an admittedly artificial in-class conversation is not always immediate. I think I speak for everyone when I say we would trust your instinct in a real restaurant ahead of any of us.

[All other students nod and look encouragingly at Haoyu.]

BINESI: This all reminds me of washing and drying dishes with a partner. One washes while the other dries, and each plate passes from the first to the second. However fast the dryer is, the plates only arrive as fast as they are washed. So the quick dryer just stands there with the towel between plates, wasting time. This is why simply comparing individual speeds, as Haoyu did at the start, leads you astray. It is the right instinct for a single person doing a job alone. It is the wrong instinct for two people handing work down a line.

RADU: Amazing analogy. It seems you speak from personal experience?

BINESI: Yes, from *too much* personal experience. My older sister was the washer, and I was the dryer. I have many memories pestering her to go faster. I may call her up after class and call her a bottleneck! I have a new taunt in my arsenal against her.

[Everyone laughs.]

RADU: Pestering one's older sister is a great societal tradition. *[Regroups.]* So, what are our thoughts about arrangement two?

FARAH: This is the other way around, but not much changes. George makes dough and Paula tops. George's capacity at the dough step is 15 per hour. Paula's capacity at the topping step is 12 per hour, same as before. The system's capacity is the minimum of these, 12 pizzas per hour. Paula is still the bottleneck.

[Radu writes while Farah talks.]

$$\begin{aligned}
 C_{\text{Paula}} &= \frac{1 \text{ pizza}}{5 \text{ minutes}} \cdot \frac{60 \text{ minutes}}{\text{hour}} = 12 \text{ pizzas per hour} \\
 C_{\text{George}} &= \frac{1 \text{ pizza}}{4 \text{ minutes}} \cdot \frac{60 \text{ minutes}}{\text{hour}} = 15 \text{ pizzas per hour} \\
 \Rightarrow \\
 C_{\text{arrangement 2}} &= \min\{C_{\text{Paula}}, C_{\text{George}}\} \\
 &= \min\{12 \text{ pizzas per hour}, 15 \text{ pizzas per hour}\} \\
 &= 12 \text{ pizzas per hour}
 \end{aligned}$$

[A brief silence.]

MING: Wait a second, I call shenanigans. Both arrangements give the same capacity!? That cannot be right. We put George on his slower task, shouldn't that reduce the capacity?

FARAH: The capacity is 12 in both cases. I checked it twice.

RADU: Let's think about why capacity is unchanged. Who is the bottleneck in both arrangements?

MING: Paula. Both times.

RADU: And what is Paula's processing time?

MING: Five minutes. At either step.

RADU: Right. Her processing time is five minutes regardless of which step she is assigned to. It does not matter that George is faster at one step than the other, because George is not the bottleneck.

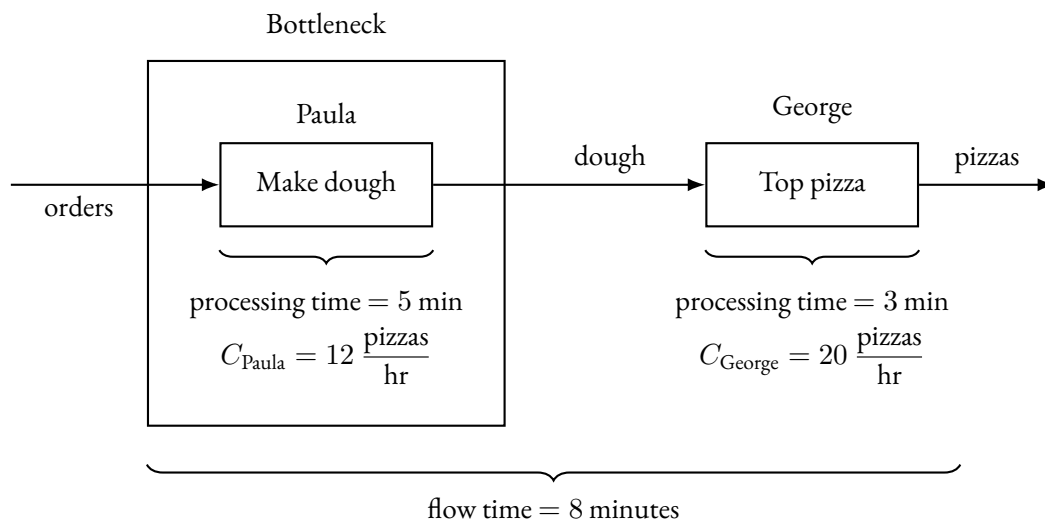
HAOYU: So, both arrangements are equally good? Then does it even matter which one we pick? And am I the only bothered by the fact that George, the fastest resource, is waiting around in both arrangements?

RADU: Let's return to your bother later. As for your first question, about whether there is any difference between these two arrangements. What do people think? Is there a difference here?

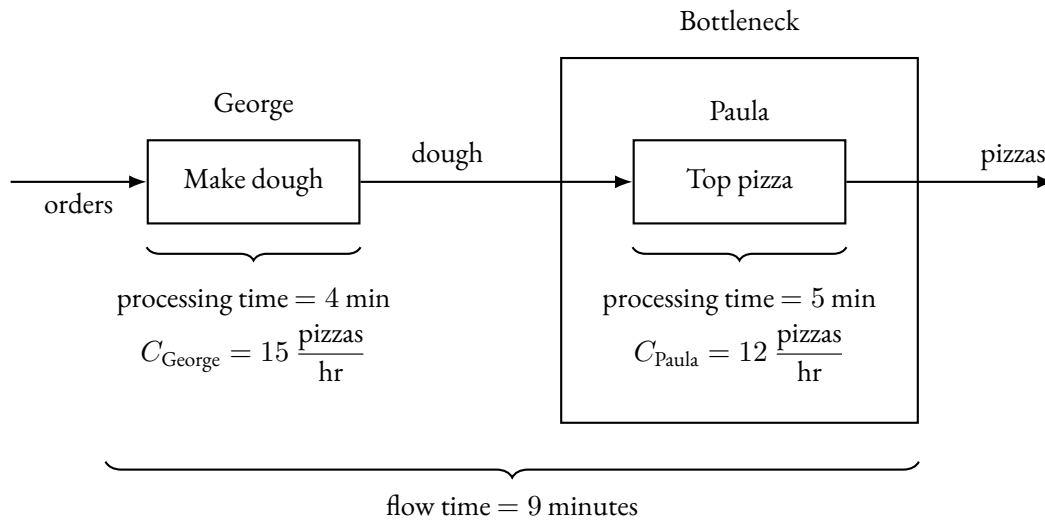
FARAH: It matters for the flow time. In arrangement one, the total processing time per pizza is $5 + 3 = 8$ minutes. In arrangement two, it is $4 + 5 = 9$ minutes. A customer who walks in gets their pizza a minute sooner under arrangement one, even though the shop's overall output rate is the same.

[Radu draws while Farah speaks.]

Arrangement 1



Arrangement 2



MING: But wait. If the flow time is different: 8 minutes in one arrangement and 9 minutes in the other, then in the first case, they can make one pizza every 8 minutes, no? The second case is slower at one pizza every 9 minutes. What is wrong with this logic?

RADU: Excellent thought ... [*Looks puzzled. Caps and uncaps a marker several times.*] ... Farah, can you address this?

FARAH: I think your calculation is wrong. How do you get 1 pizza every 8 minutes for arrangement one?

MING: Well ... the flow time is 8 minutes, and so the throughput is one pizza every eight minutes!

FARAH: [*Pauses.*] I think you may be misusing Little's Law here.

MING: I didn't know I was even using Little's Law.

FARAH: Little's Law says $I = RT$. For that to give $R = 1/T$, which is what you calculated, you need $I = 1$. One pizza in the system at a time. But in a serial system with two workers, there is more than one pizza in the system at once. Paula has one pizza in her hands, and George may have another. The inventory is not 1.

RADU: This is a good point. So what is I here? I am not sure myself.

FARAH: Let me work it out for arrangement one. Paula is the bottleneck. She is always working, so she always has one pizza. George tops in 3 minutes but only receives a new dough every 5 minutes. So he has a pizza in his hands for 3 minutes out of every 5. His average contribution to inventory is $3/5 = 0.6$ pizzas. The total average inventory is $1 + 0.6 = 1.6$ pizzas.

RADU: And does Little's Law check out?

FARAH: Yes: $I = RT$ gives $1.6 = 12 \times (8/60)$. [*Radu writes on board while Farah is talking.*]

$$\underbrace{1.6}_I = \underbrace{12}_R \times \underbrace{\frac{8}{60}}_T$$

The capacity is 12 in both arrangements, but the inventory differs. Arrangement two has $I = 1.8$ because George is occupied $4/5$ of the time rather than $3/5$. The longer flow time is matched by more inventory, leaving throughput unchanged.

MING: So Little's Law is not violated. The inventory adjusts to compensate. [Pauses.] I withdraw the shenanigans.

RADU: Exactly. When the flow time goes up, something else must change to keep Little's Law balanced. Here it is the inventory. This is actually a very useful thing to see in a picture. Let me draw what is happening in each arrangement step by step. [Turns to the board.] The picture I am going to draw is called a *Gantt chart*.

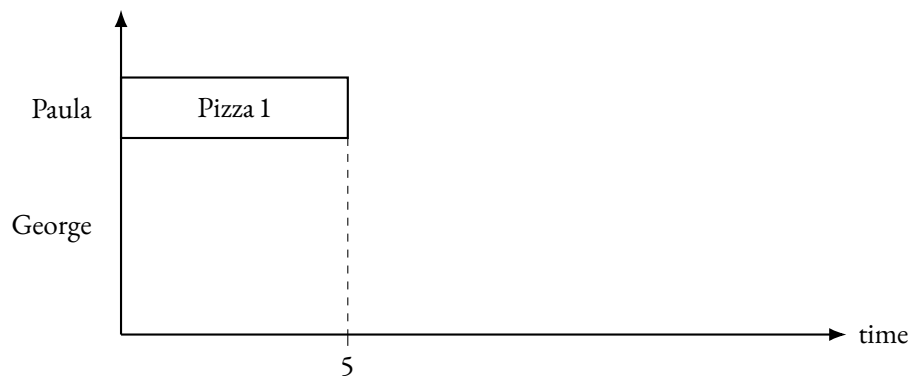
The first thing I draw is the axes. Time runs along the bottom, in minutes. There are two rows along the horizontal representing our two workers—Paula on top, George below. [Labels the rows.] Each row shows how that worker spends their time. A filled block means they are working on a pizza. A gap means they are idle, waiting for something.



Does everyone see the basic structure?

[Students nod.]

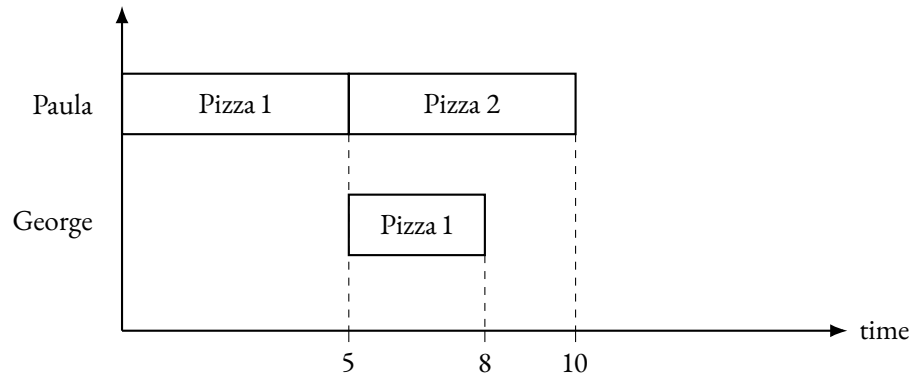
Good. Now I add the first pizza. [Draws.] Paula starts at time zero and makes the dough. That takes 5 minutes, so I draw a block from 0 to 5 on Paula's row. At time 5, Paula is done. She passes the dough to George. George begins topping. He takes 3 minutes, so his block runs from time 5 to time 8.



Any questions so far?

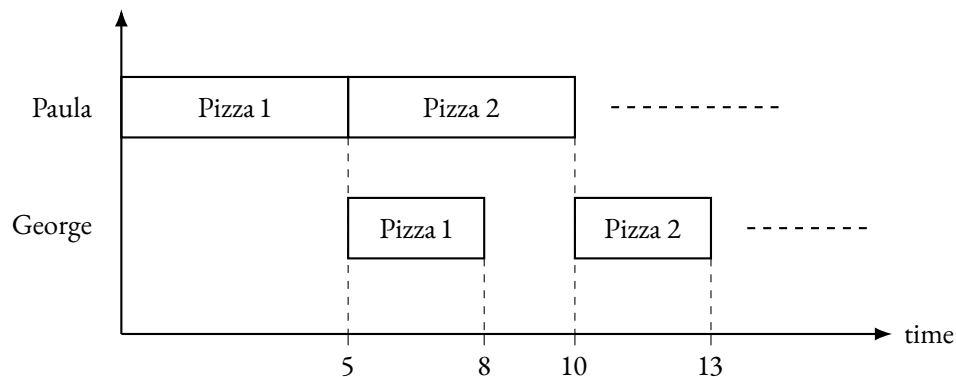
BINESI: I see the picture. But after time 5, what is Paula doing while George is topping?

RADU: That's the key question. There is no need for Paula to wait for George to finish. As soon as she hands the dough to George at time 5, she starts making the dough for pizza number two. [Draws.] So I add a second block on Paula's row, from time 5 to time 10.



MING: George finishes at time 8 but Paula's next dough is not ready until time 10. So George waits two minutes?

RADU: Precisely. That gap on George's row from time 8 to time 10 is idle time. He has nothing to top because the next dough is not ready. And the pattern repeats. At time 10 Paula passes the second dough, George tops from 10 to 13, then waits again from 13 to 15, and so on. [Completes the chart.]



Now, let me define something precisely. Look at Paula. In every 5-minute cycle, she works for 5 minutes. The fraction of time she spends working is $5/5 = 1$, or 100%. Now look at George. In every 5-minute cycle, he works for 3 minutes and waits for 2 minutes. His fraction is $3/5 = 60\%$. This fraction—the proportion of time a resource is actively working—is what we call *utilization*, defined like this: [Draws.]

$$u = \frac{\text{time working}}{\text{time available}}$$

Paula's utilization is 100%. George's utilization is 60%. Paula is at capacity. George is not.

HAOYU: See, as I suspected, George is being wasted 40% of the time in arrangement one?

BINESI: [*Quickly.*] Or he is being protected 40% of the time. [*Pauses.*] Think back to Beanstalk. Ebrahim and Dae-jung were running above capacity, with no recovery time built in. George in arrangement one has 2 minutes in every 5-minute cycle where he is not making pizzas. He can breathe. He can clean his station. He is not being squeezed the way Ebrahim and Dae-jung were.

RADU: That is an important observation. Idle time in a non-bottleneck resource is often seen as inefficiency. But sometimes there is no way to be more efficient!

HAOYU: In my restaurant, I would give him more to do in those two minutes.

RADU: Which could work, as long as what you give him to do does not interfere with topping pizzas.

HAOYU: Obviously.

RADU: Yes, obviously. Now let's draw arrangement two. George makes dough, Paula tops. The axes are the same. George starts at time zero, making dough. He finishes at time 4. At time 4, the dough is ready. But can Paula start topping immediately?

FARAH: She has nothing else to work on, this is the first pizza! So yes, she can start topping at time 4.

RADU: Correct. But notice something. Paula tops in 5 minutes. So she runs from time 5 to time 10 ...[*Interrupted.*]

FARAH: Wait, shouldn't it be time 4 to time 9, not 5 to 10? She starts when the dough arrives at time 4.

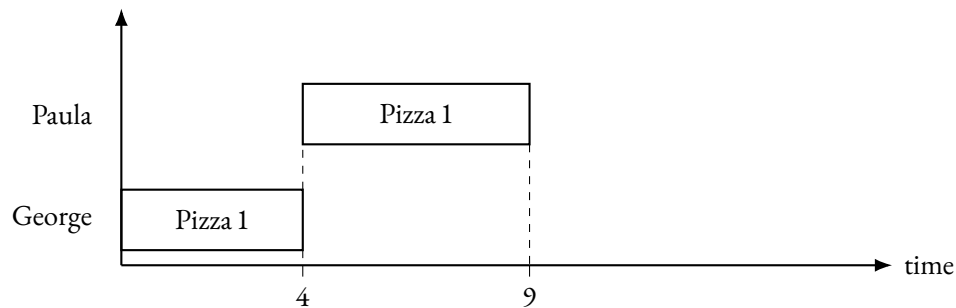
RADU: You are right. Paula tops pizza one from time 4 to time 9. Let me draw that so it is clear to you all, and hopefully to me too.

MING: Should we compute your utilization, Radu? You are starting to make errors.

RADU: One hundred percent utilization, I am sure. I am definitely the bottleneck of this classroom.

[*Everyone laughs.*]

Let me draw the picture now.



So, when does George start on the second dough?

MING: As soon as he finishes the first. Time 4.

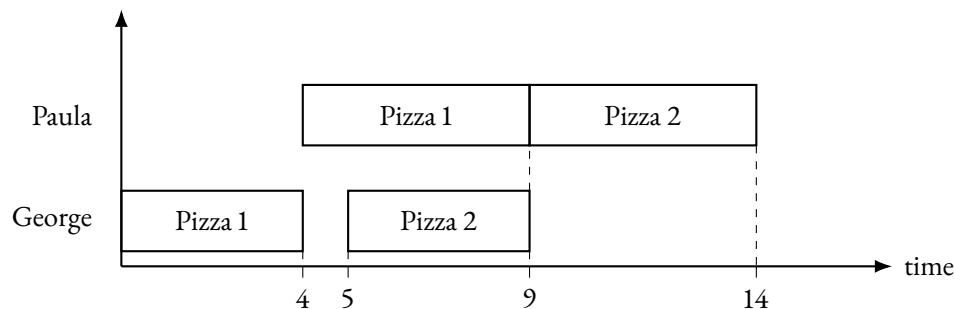
RADU: But Paula is busy topping until time 9. George finishes his second dough at time 8. Can he hand it to Paula?

MING: No. She is not done with the first pizza yet.

RADU: So what should George do?

MING: I see two options now. One where George starts working on the next dough right away at time 4 and then waits a minute from time 8 to 9 before handing it off to start the next one. The other is George waits until time 5 to start working on the next dough and hands it to Paula just as she is ready at time 9. I think I prefer to second option, it keeps the dough fresher.

[Radu draws the second option on the board.]



CAMILA: Wait a second, why does George need to wait for Paula at all? In this arrangement, I don't see what stops him from making more dough.

MING: And put this dough where?

CAMILA: Just on a table or something. Is this a problem?

MING: Yes, I thought about that, and it seems wrong.

RADU: It boils down to thinking about what is more wasteful: George's occasional idleness or pizzas piling up? Notice that if George keeps working ahead of Paula, this pile of rolled out pizza dough gets higher and higher.

CAMILA: I see your point. It's a tradeoff between idleness and growing the inventory.

FARAH: But why is this a tradeoff? There is no way for George *not* to be idle here. Everything gets stuck at Paula regardless, and she is not getting any faster.

RADU: Exactly right. We call this phenomenon *blocking*. George is eventually blocked from producing more as he waits for Paula, either because he has to wait or he runs out of places to put pizzas.

BINESI: If we kept George working harder here, it would just create more, potentially wasted dough.

HAOYU: Dough sitting around, drying out. That's the real crime here. [Haoyu looks personally offended by the thought.]

BINESI: We have found the one thing Haoyu will not optimize. Working people until they bleed, okay. Wasting dough, shudder.

HAOYU: What can I say? I would bleed for good dough.

[Everyone laughs, including Binesi, who gives a thumbs up to Haoyu.]

RADU: So, what is George's utilization now? Let's assume he waits for Paula, there is no room to put down the doughs in the interim.

FARAH: He works 4 minutes out of every 5. That is $4/5 = 80\%$.

RADU: And Paula's?

FARAH: Still 100%. She is still the bottleneck.

RADU: Good. Now compare the two Gantt charts side by side. Same capacity: 12 pizzas per hour. Same bottleneck. Same Paula utilization. But George is at 60% in arrangement one and 80% in arrangement two. Binesi, what do you make of that?

BINESI: Arrangement one is better for George. He has twice as much breathing room—40% idle versus 20%. Same output, less strain.

CAMILA: Though from a pure utilization standpoint, you would say arrangement two is more efficient. George is doing more.

RADU: More efficient?

CAMILA: Um, right. Oh wait ...

RADU: This is why I don't like the word efficient, it has so many meanings in the vernacular. If our goal is to keep people busy, arrangement two is more "efficient" at that. But if our goal is to produce a pizza with wasting a minimum of effort, arrangement two is better.

CAMILA: Yeah, I get it. It is just a force of habit. It is hard for me to say a system that has more idleness is better. Isn't there some famous saying about this, "Idle hands are the devil's workshop"?

HAOYU: Yet another reason to keep all the hands busy!

[Everyone laughs.]

RADU: This is a tension we will keep returning to. A manager who looks only at keeping his workers busy will prefer arrangement two. In this arrangement, George is 80% utilized versus 60%. A manager who thinks about sustainability and what happened at Beanstalk will prefer arrangement one. The numbers do not tell you which value to prioritize. They just make the tradeoff visible.

MING: I see it differently. Arrangement one wins on flow time, on process time, and on George's wellbeing. And loses on nothing.

RADU: That is certainly one way to look at it. When one option dominates another on every dimension, the choice is straightforward. Maybe let's go with arrangement one?

HAOYU: But George has a lot of idle time in this picture. Can we give him something useful to do during those two minutes?

BINESI: Why is your instinct always to give people more work? *[Sits up, seems ready for a fight.]*

HAOYU: Because I have to meet payroll!

BINESI: And because you have to meet payroll, George does not get to breathe!

RADU: Okay, okay, okay, again you two? *[Pauses while people calm down.]* I think it is a reasonable instinct to think about having George do some additional tasks that are not in the pizza-making path—say, managing orders, cleaning, preparing ingredients. That may make him less bored, for example? But I think there are risks with this instinct too. Can we come back to similar thoughts in later classes?

MING: Anything to fill our idle time.

[Everyone laughs as they pack up to slowly file out of the room.]

Email

Dear students, let me use this email to consolidate what we covered in class and push a few of the ideas further.

The capacity of a single resource. When a resource processes one job at a time and takes t minutes of processing time per job, its capacity is:

$$C = \frac{1 \text{ job}}{t \text{ minutes}} = \frac{60}{t} \text{ jobs per hour}$$

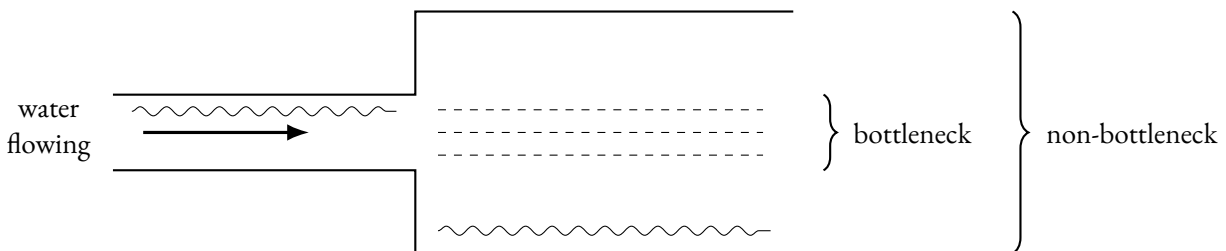
Paula alone, taking 10 minutes per pizza (5 minutes dough, 5 minutes topping), has capacity $\mu_P = 6$ pizzas per hour. George alone, taking 7 minutes, has $\mu_G = 60/7 \approx 8.6$ pizzas per hour. This is where Haoyu's intuition came from, and it is not wrong as far as it goes: George working alone produces more than Paula working alone.

The capacity of a serial system. When two resources work in series, one passes output directly to the next. The capacity of the combined system is:

$$C_{\text{system}} = \min\{C_1, C_2\}.$$

The resource whose individual capacity equals the system capacity is the *bottleneck*. This is the resource that limits flow, in exactly the sense that the narrow part of a bottle limits the rate at which liquid pours out.

More concretely, think of two segments of pipe connected end to end. No matter how wide the second segment is, the flow through the combined pipe cannot exceed the flow through the narrowest segment. Widening a non-bottleneck segment achieves nothing in terms of the amount of flow that can travel through a system. Only widening the bottleneck increases flow.



There can in principle be more than one bottleneck along a path—if two resources happen to have the same individual capacity, both are bottlenecks simultaneously. And in more complex operations with multiple paths, there can be a different bottleneck on each path. We will return to this.

Process time versus flow time, again. We encountered this distinction in an earlier class, and it came up again today when Ming tried to apply Little's Law with $I = 1$.

The *process time* of a pizza is the total time spent working on it: $5 + 3 = 8$ minutes in arrangement one. This is the time a customer waits from placing their order to receiving their pizza, if the system is empty when they arrive.

Ming implicitly made a Little's Law calculation setting $I = 1$, which is a very common inclination. But as Farah showed, the average inventory in arrangement one is $I = 1.6$ pizzas: Paula always has one, and George has one 60% of the time. In arrangement two, the average inventory is $I = 1.8$, reflecting George's higher utilization (80%). Both arrangements have $R = 12$ pizzas per hour; the different flow times are absorbed by different inventory levels, leaving throughput unchanged. Little's Law is not violated—the inventory adjusts.

Utilization in a serial system. The *utilization* of a resource is the fraction of time it spends actively working:

$$u = \frac{\text{time working}}{\text{time available}}.$$

In a serial system running at capacity, the bottleneck has utilization 100%. It is working continuously, and any pause on its part directly reduces system output. Non-bottleneck resources have utilization less than 100%: they finish their step and then wait for the bottleneck to deliver the next unit.

In arrangement one: Paula (bottleneck) is 100% utilized. George works 3 minutes out of every 5-minute cycle, so his utilization is $3/5 = 60\%$. In arrangement two: Paula is still 100% utilized. George works 4 minutes out of every 5-minute cycle, giving a utilization of $4/5 = 80\%$.

The Gantt charts from class make this visible. Paula's row is filled solid in both charts. George's row has regular gaps: 2-minute gaps in arrangement one, and 1-minute gaps in arrangement two.

A common managerial instinct, which Camila voiced in class, is to treat idle time in a non-bottleneck resource as waste to be eliminated. But as Binesi noted, idle time in a non-bottleneck may serve a function: it is the recovery time that makes the pace sustainable, and it is a buffer that can absorb small disruptions without immediately propagating delays through the system. A system in which every resource runs at 100% utilization has no slack and no resilience. This is an important theme we will take up carefully in later classes.

Blocking. Camila asked whether George, in arrangement two, could simply keep making dough continuously rather than waiting for Paula to be ready, accumulating a pile of dough in the interim.

In principle, he could, but only for a while. Working continuously in this way would cause the inventory of untopped pizzas to grow without bound: George produces at a rate of 15 per hour while Paula tops at a rate of 12 per hour. The pile would grow by 3 pizzas per hour indefinitely. This is unsustainable. Eventually there is nowhere to put the dough, and eventually the dough itself degrades.

The phenomenon of a resource being prevented from working because the downstream resource is not ready, or because there is no space to put the output, is called *blocking*. In our case, George is blocked by Paula. He must wait, or the inventory between them grows without limit. In real operations, blocking is managed through buffer design: a small buffer between two serial resources allows the upstream resource to work slightly ahead of the downstream one, absorbing small timing mismatches without causing the upstream resource to idle. But no finite buffer can compensate indefinitely for a systematic capacity imbalance. This will be a major theme of later discussions.

Flexibility: one way around the block. A friend once described to me how her family makes dumplings, and it has stayed with me as the clearest illustration of an idea we have not yet touched. Dumpling-making, like pizza-making, has two steps in sequence: someone rolls the wraps, and someone folds the filling into them. As a child, she worked alongside her mother, and her mother was faster at both jobs.

What her mother did was not to pick a step and stay there. She would roll wrappers steadily, stacking them on a second plate until a small pile had built up, then set down the rolling pin and join her daughter in folding, working the pile back down. Then she would return to the wrappers. My friend remembers this as an ordinary family evening. It is also, as it happens, a rather good piece of operations design.

Apply it to our shop. Suppose George is not assigned to a step at all, but moves between them. He makes dough continuously, letting a small stack accumulate beside him. When the stack has grown to two or three, he sets the dough aside, tops one of the waiting pizzas himself, and then goes back to the dough. Paula never stops topping, because the stack guarantees she is never starved.

Now let's look at the calculations. Consider a repeating 35-minute cycle. George spends 32 minutes making eight doughs and 3 minutes topping one pizza, for 35 minutes in all. Paula spends the full 35 minutes topping seven pizzas. Eight doughs are made and eight pizzas are topped, so the stack ends the cycle exactly where it began and the pattern can repeat indefinitely. The throughput is

$$R = \frac{8 \text{ pizzas}}{35 \text{ minutes}} \cdot \frac{60 \text{ minutes}}{\text{hour}} \approx 13.7 \text{ pizzas per hour,}$$

I do not want to oversell this. Flexibility of this kind is not free. It requires a worker who is genuinely competent at both steps, which is precisely why it was George's role here and not Paula's. It requires somewhere to put the stack. It also asks more of the flexible worker, who now carries the mental overhead of deciding, minute by minute, which job needs him. What the example shows is not that flexibility is always right. It is that the bottleneck we identified so confidently in class was a property of *how we chose to assign the work*, not a fact about Paula. Change the assignment rule and the constraint moves. We will return to this in later classes.

Investing in the non-bottleneck. One more point from the notes I want to make explicit, because it is frequently counterintuitive in practice.

Suppose Paula invests in a technology that lets George top a pizza in 2 minutes instead of 3. His capacity jumps from 20 to 30 pizzas per hour. What is the new system capacity in arrangement one?

$$C_{\text{system}} = \min\{12, 30\} \text{ pizzas per hour} = 12 \text{ pizzas per hour.}$$

Nothing changed. The system capacity is still 12. Paula is still the bottleneck, and improving George's throughput, who was already well above the bottleneck rate, does nothing to relieve it.

What did change is the process time per pizza, which falls from $5 + 3 = 8$ minutes to $5 + 2 = 7$ minutes. So individual customers receive their pizzas one minute sooner. If that matters to the business, the investment may be worthwhile. But if the goal was to increase the number of pizzas produced per hour, the investment was entirely misdirected.

The principle is stark: *to increase the capacity of a serial system, invest in the bottleneck*. Any investment in a non-bottleneck resource that does not also relieve the bottleneck will not increase system capacity. This sounds obvious once stated. It is violated constantly in practice, because the non-bottleneck is often more visible, more measurable, or easier to improve. If you find this hard to believe, you will see exactly what I mean in the classes ahead.

Warmest, Radu.

Extensions

[Posted as homework on the course's website for this class session.]

1. The case offers two arrangements, and the class dutifully analyzed both. But there is a third: Paula and George each make *complete* pizzas, start to finish, working side by side in parallel.
 - (a) Compute the capacity of this parallel arrangement. You should find it beats both serial arrangements. By how much?
 - (b) Something strange has happened: the class spent a whole session choosing between two options that are both dominated by an option no one mentioned. Why do you think nobody—including me (Radu)—proposed it? What does that suggest about how a case's framing shapes the answers you consider?
 - (c) Now defend the serial arrangements anyway. Give at least three practical reasons a real shop might prefer serial to parallel even at lower capacity (think about equipment, skill, consistency, and what happens when one worker is absent). Which reason do you find most convincing, and does it survive if the shop doubles in size?
2. Suppose Paula can invest in exactly one improvement, each of which shaves one minute off one person's time at one step. There are four candidates: Paula's dough ($5 \rightarrow 4$), Paula's topping ($5 \rightarrow 4$), George's dough ($4 \rightarrow 3$), George's topping ($3 \rightarrow 2$).
 - (a) For each of the four improvements, compute the new system capacity under the *best* arrangement available after the improvement. (Careful: the best arrangement may change.)
 - (b) Rank the four investments. You should find that improvements of identical size have wildly different values. State, in one sentence, the principle that explains the ranking.
 - (c) Now suppose the improvement to Paula's dough time costs three times as much as the others. Does the ranking change? What is the maximum premium worth paying for it?
3. Paula is considering hiring her niece Sofia, who (being new) takes 6 minutes at either step. Decide where Sofia should go. Consider at least: Sofia on dough alongside Paula (two dough-makers feeding George), Sofia on topping, and Sofia floating between stations. For your best design, compute the new system capacity, identify the new bottleneck, and state each worker's utilization.
4. The chapter introduced two ways a worker can be forced idle: *starved* (nothing to work on) and *blocked* (nowhere to put finished work). For each of the following, say whether the idle worker is starved, blocked, or something else entirely, and name the bottleneck: (a) a cashier with no customers in line; (b) a surgeon waiting for the operating room to be cleaned; (c) a programmer whose code is finished but cannot deploy until the weekly release; (d) a translator waiting for the author to send the next chapter; (e) you, in any meeting of your choosing. Then state the general rule: from which direction does starvation come, and from which does blocking?

5. Paula's accountant reviews the shop and reports: "George is idle 40% of the time under the current arrangement. Recommend switching workers' roles to raise his utilization to 80%, or reducing his hours and pay by 40%." Write Paula's reply memo. It must (a) explain, with the numbers, why raising George's utilization produces zero additional pizzas; (b) address what would actually happen to output and to George if his hours were cut; and (c) propose what metric the accountant *should* track instead of utilization.
6. Paula decides to start selling *baked* pizzas. Baking takes 8 minutes in an oven that holds one pizza. Now the process is dough, top, and bake.
 - (a) With Paula on dough, George on topping, and the oven baking, find the system capacity and the new bottleneck.
 - (b) An oven that holds two pizzas at once costs twice as much. Compute the capacity with it, and say whether the doubling is worth it.
7. Paula wants to add a second product: a calzone, which uses the same two workers but takes 7 minutes of dough work and 2 minutes of filling work.
 - (a) Suppose the shop sells only calzones. For each of the two serial arrangements, find the capacity and the bottleneck. Notice anything different from the pizza case?
 - (b) Now suppose demand is half pizzas, half calzones, alternating one after the other. With Paula on dough and George on topping/filling, trace a Gantt chart for the sequence pizza–calzone–pizza–calzone and find the throughput. Is it the average of the two pure-product capacities? Explain why or why not.
 - (c) State the general lesson in one or two sentences: what does adding a product to a shared process do to the very idea of "the" bottleneck?

Reflections

[Posted as a reflection exercise on the course's website for students to write about in a journal due at the end of semester.]

1. Haoyu's first instinct was that the faster person should simply do everything. Recall a time you (or a manager above you) gave all the work to the strongest performer. What happened to the output—and what happened to the strongest performer?
2. Paula is the bottleneck in her own shop: slower than George at every step, yet the whole operation's pace is hers. Have you ever been the bottleneck on a team, the one everyone was politely waiting on? How did you find out, and what did it feel like once you knew?
3. George is idle two minutes in every five, and Binesi called that being "protected" rather than "wasted." When you have slack in your day, do you experience it as recovery or as guilt? How did you learn to feel that way about it?

4. The class discovered that raising George's utilization from 60% to 80% produces not one additional pizza. Where in your life are you busier without being more productive? What, honestly, is the busyness for?
5. Being *starved* is waiting for work to arrive; being *blocked* is having finished work with nowhere for it to go. Which of the two do you experience more often, and which one bothers you more? What does that say about what you need from the people upstream and downstream of you?
6. The email warns that the non-bottleneck is often improved simply because it is "more visible, more measurable, or easier to improve." Think of a real improvement effort you have watched, at work, at school, in a family. Was it aimed at the true constraint, or at the thing that was easiest to polish? How could you tell?
7. Suppose someone audited the idle time in your own week and proposed to fill it. What would you defend keeping empty, and on what grounds?