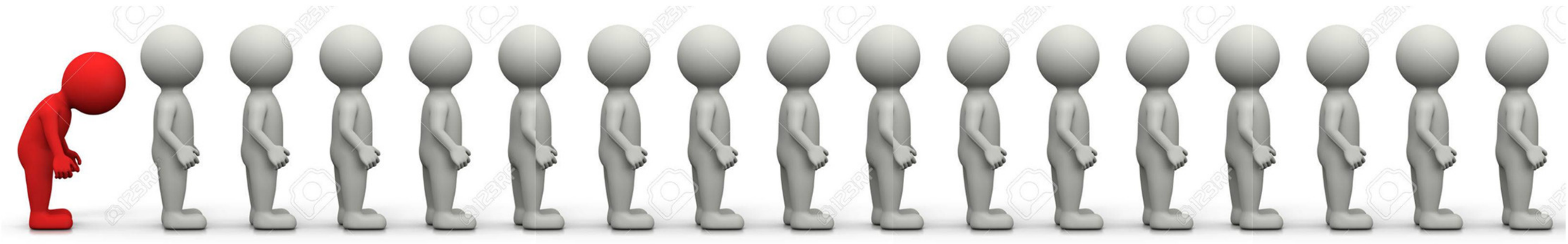


Waiting & Queues
People vs. Computers

Richard Gimarc
rgimarc@featherfall.com

Computer Measurement Group
SOUTHWEST

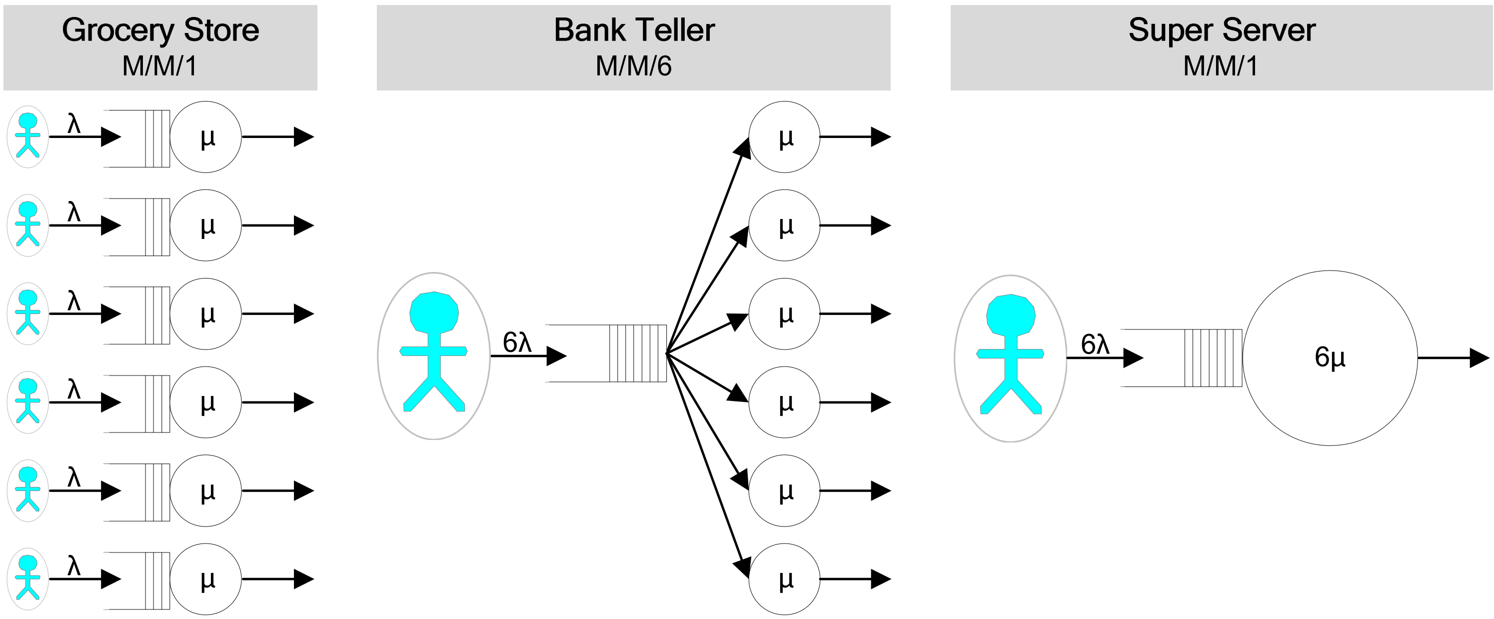
September 19, 2018
Southwest CMG

What are we going to talk about for the next 30 minutes?

“*Waiting & Queues - People vs. Computers*”

- **Examine three “equivalent” queues**
  - Choose the one has the lowest response time
  - Confirm our intuition with queuing theory
  - Get secondary confirmation from a “load test”
- **Hands-on modeling to solve a real queuing problem**
  - Use R to model an application processing trades at “market open”
  - Demonstrate efficiency of this modeling approach

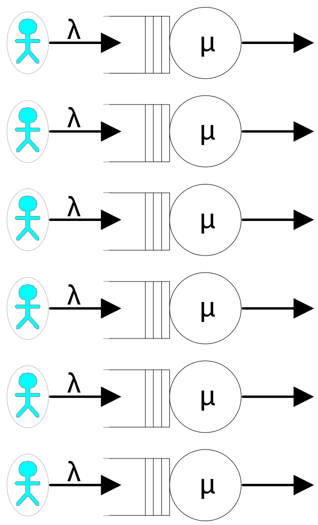
Question: *Which architecture has the lowest response time?*



- Same total customer arrival rate (6λ)
- Service centers have the same total processing power (service rate of 6μ)

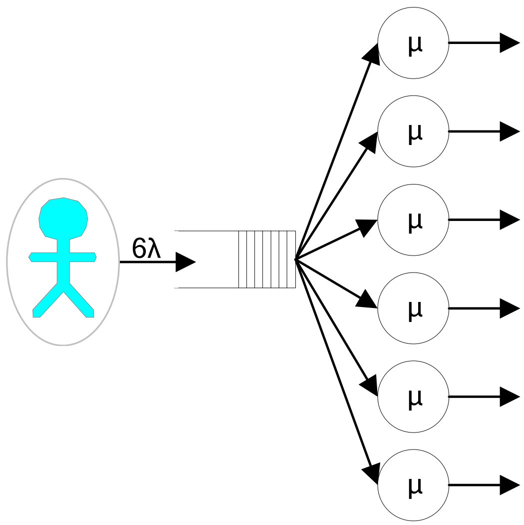
Question: ***Which architecture has the lowest response time?***

Grocery Store
M/M/1



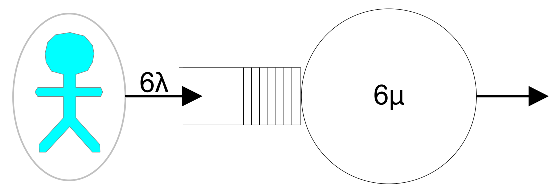
- 6 pairs of queues and checkers
- Random arrivals at each checkout line

Bank Teller
M/M/6



- Single shared queue
- 6 separate tellers
- Arrival rate is 6x
- A free teller is assigned the first customer in the queue

Super Server
M/M/1

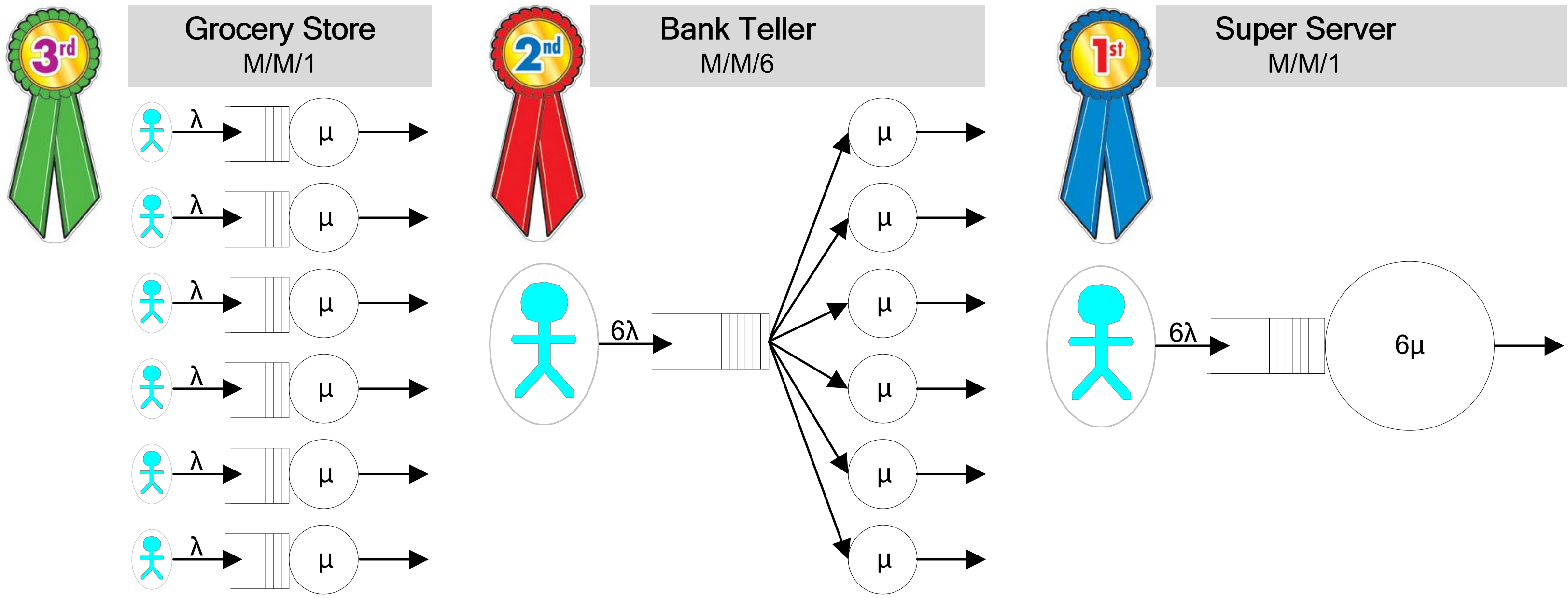


- Single queue & service center
- Arrival rate is 6x
- Service time is 1/6

Question: ***Which architecture has the lowest response time?***

What's the ordering, fastest to slowest in terms of response time?

Answer: ***Which architecture has the lowest response time?***

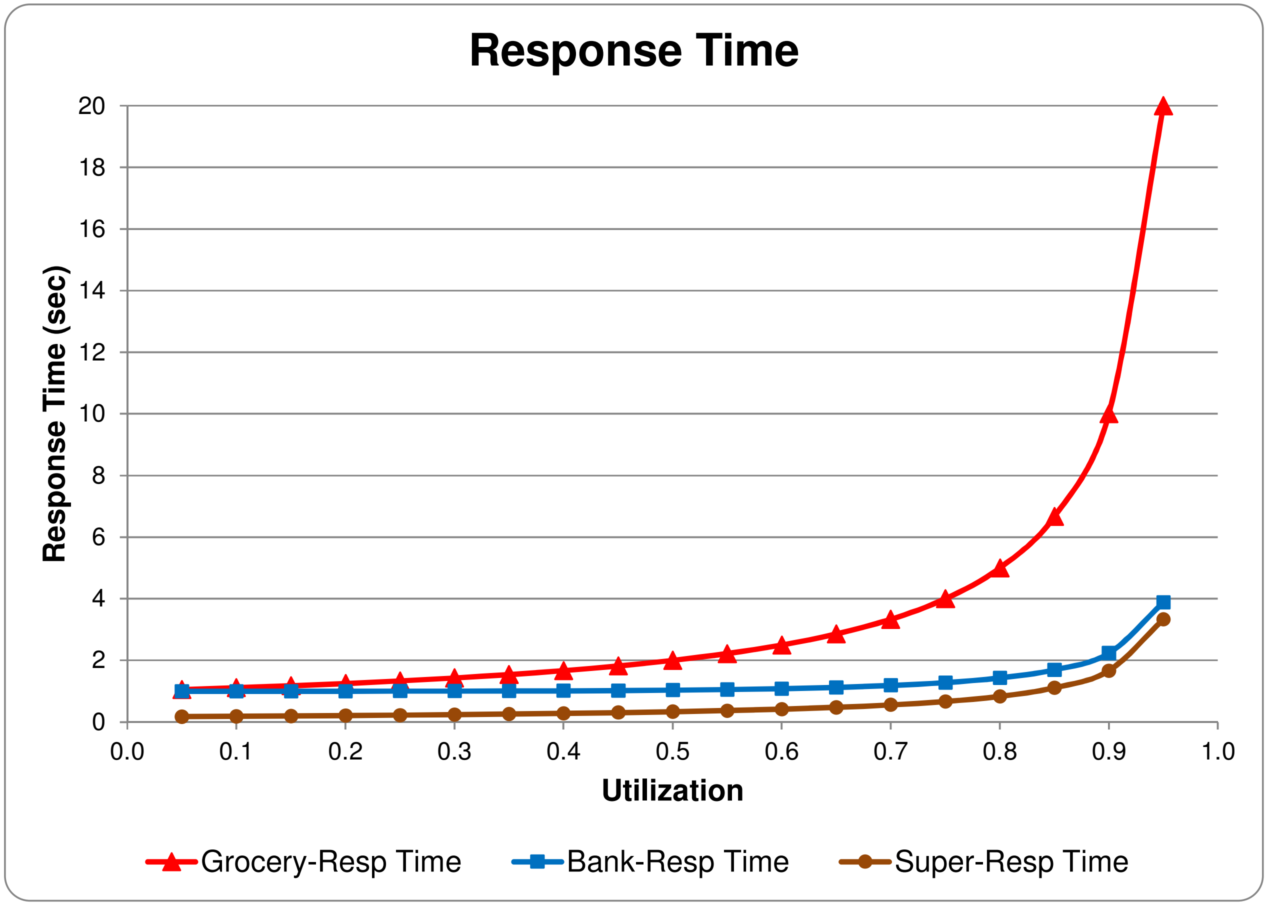


- Customers may be waiting in line when a checker is idle
- Unbalanced queues

- You may have idle servers when there are not enough customers to keep them all busy (unused capacity)

- If customer count > 0, then total processing power is used

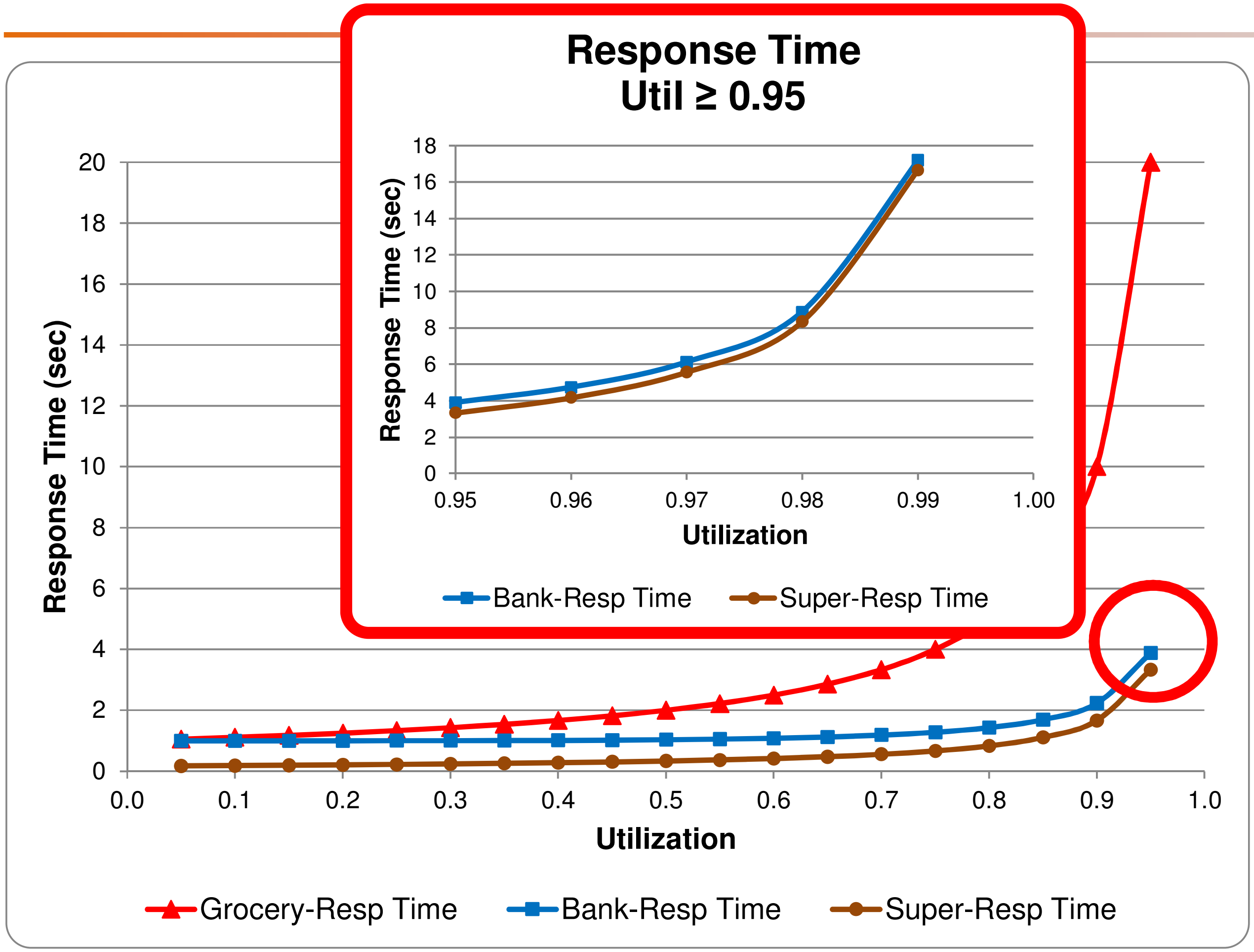
Queuing Theory - Response Time Comparison



Compute total response time using queuing theory

- At low utilization, service time is the dominant component of response time (no queuing)
- Grocery curve seems to begin rapid increase at 50%
- Super & Bank curves begin to rise at 70%
- At high utilization, Bank and Super seem to converge …

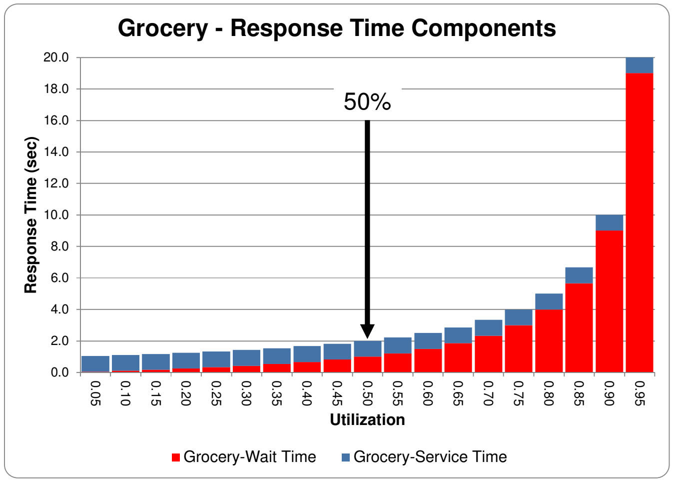
Queuing Theory - Response Time Comparison

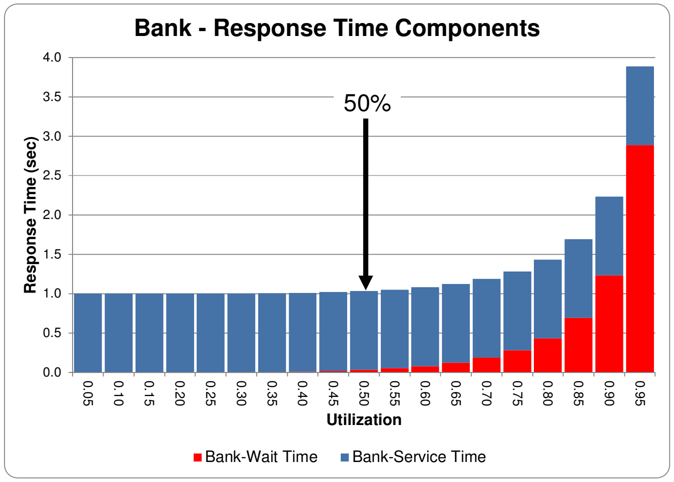


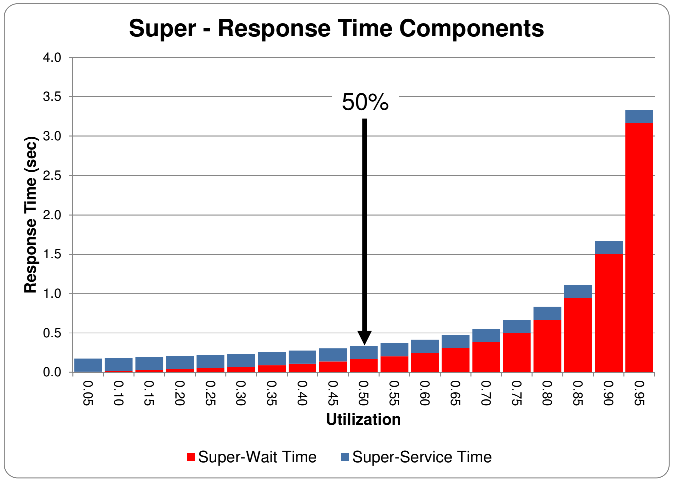
Compute total response time using queuing theory

- At low utilization, service time is the dominant component of response time (no queuing)
- Grocery curve seems to begin rapid increase at 50%
- Super & Bank curves begin to rise at 70%
- At high utilization, Bank and Super seem to converge

Queuing Theory - Wait Time Comparison





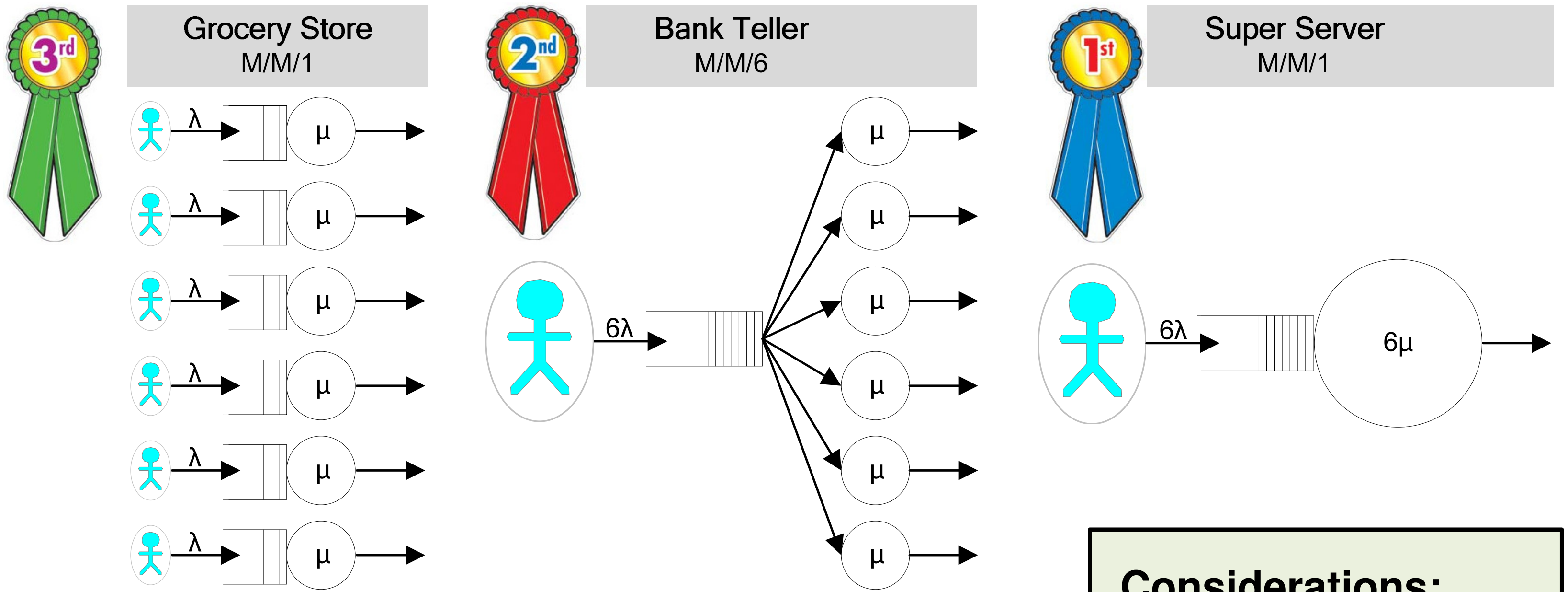


- Blue = Service time
- Red = Wait time

Plot wait and service time from our queuing model

- Note y-axis scale on the Grocery, 20 vs. 4
- Grocery & Super have similar shapes (M/M/1)
- Wait component of Bank is more prominent at high utilization (M/M/6)
- Bank's knee of the curve pushed to the right due to the increased number of servers (6 vs. 1)

Next Question: ***Which architecture would you use for an application?***



Considerations:

- Response time
- Single point of failure
- Cost

"Load Test" - Grocery vs. Bank

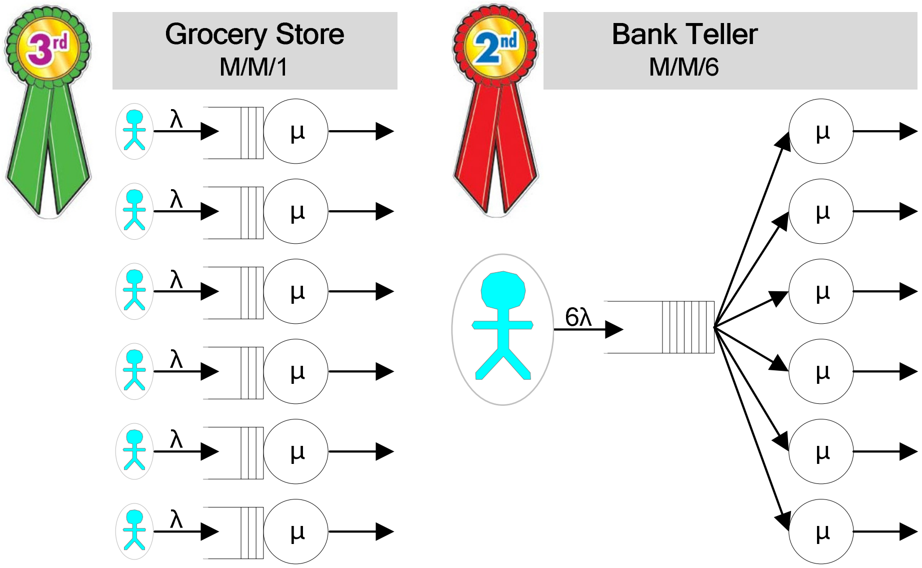
MythBusters – "*Volunteer Special*"

(Episode 242, Feb 6, 2016)

- **Myth**
  - "*In a grocery store, the standard method of letting shoppers choose a checkout counter is not as efficient as a single long "serpentine" line that routes each shopper to the next available checkout.*"
- **But**
  - Queuing theory tells us that the single line gives us a better response time than the individual checker lines

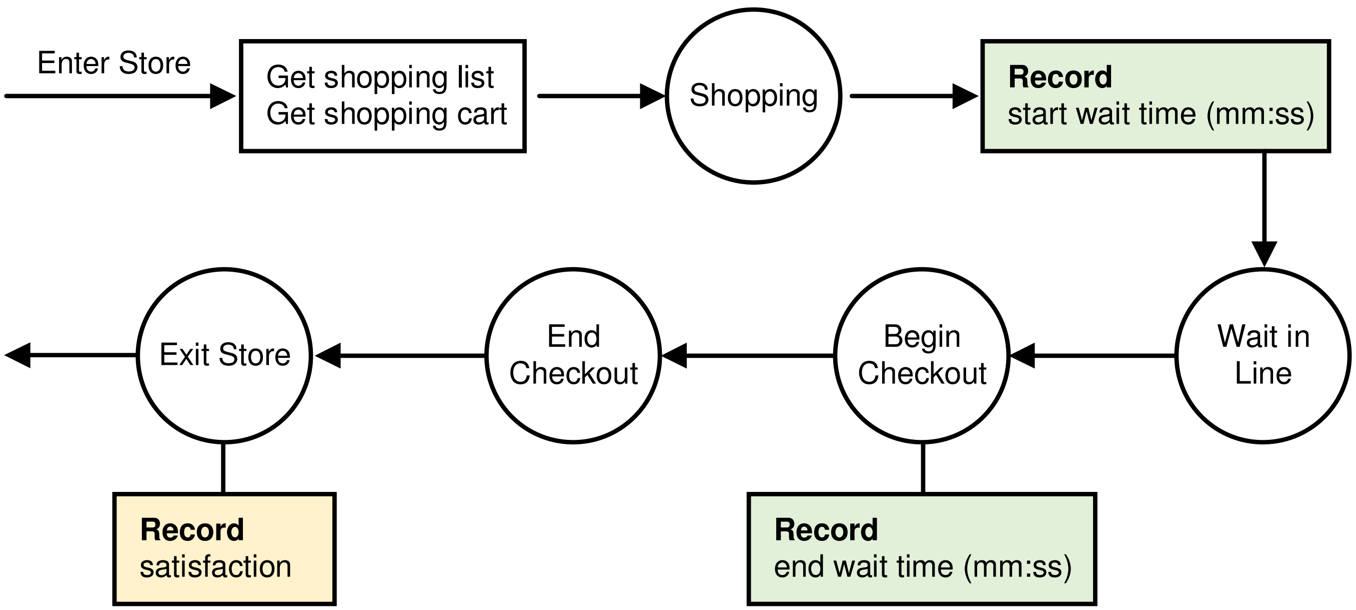
Expect this myth to be confirmed

"Pick a Lane" **"Serpentime"**

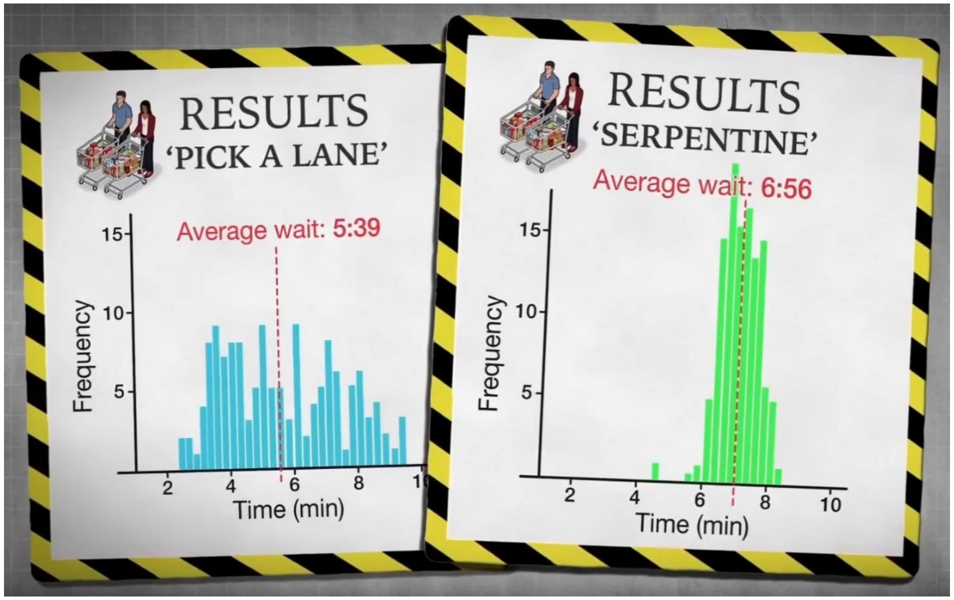


“Load Test” - Grocery vs. Bank

- **Setup**
  - 90 customers & 5 experienced cashiers
  - 5,000 food items spread across 750 feet of shelving
  - 5 minute warmup & 30 minute steady state
- **Metrics**
  - Wait time (mm:ss)
  - Satisfaction: 1-5 (low-high)



"Load Test" - Grocery vs. Bank



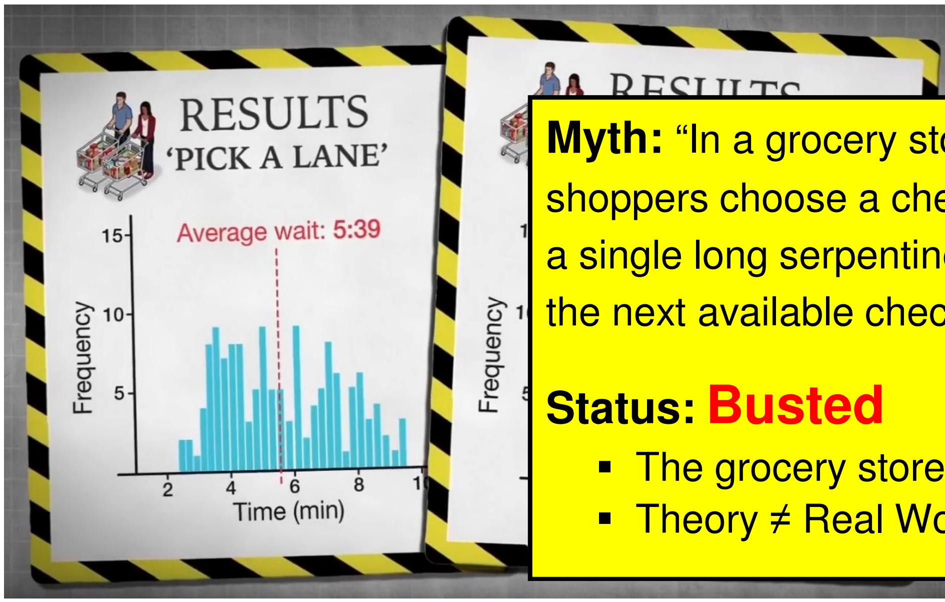
Wait Time

- Grocery store is >1 minute faster (5:39 vs. 6:56)
- Grocery has wider range of wait time (2:10 – 9:21)

Satisfaction

- Bank has higher satisfaction (3.8 vs. 3.45)
- Bank has more "5" (high) scores

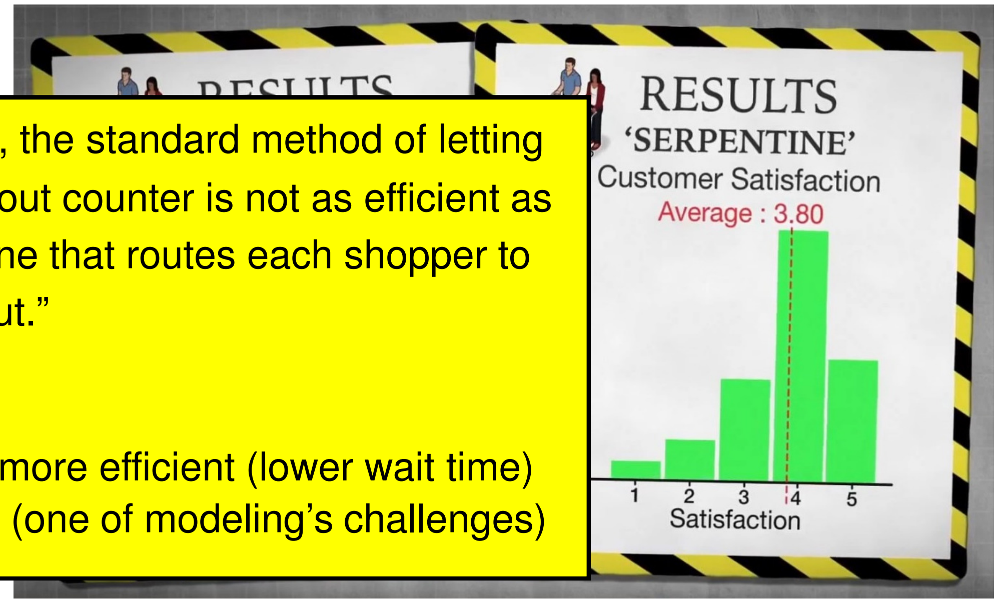
“Load Test” - Grocery vs. Bank



Myth: “In a grocery store, the standard method of letting shoppers choose a checkout counter is not as efficient as a single long serpentine line that routes each shopper to the next available checkout.”

Status: Busted

- The grocery store is more efficient (lower wait time)
- Theory ≠ Real World (one of modeling’s challenges)



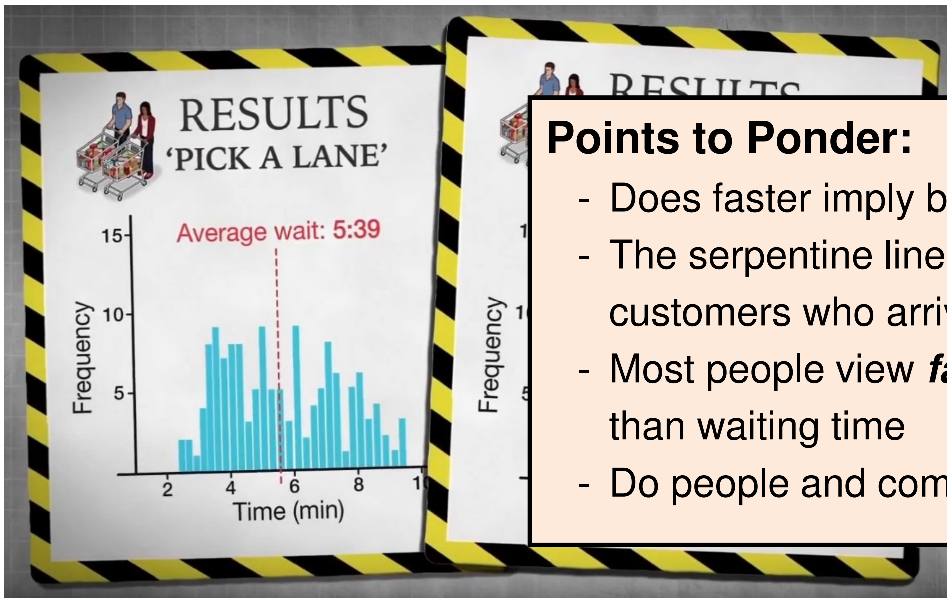
Wait Time

- Grocery store is +1 minute faster (5:39 vs. 6:56)
- Grocery has wider range of wait time (2:10 – 9:21)

Satisfaction

- Bank has higher satisfaction (3.8 vs. 3.45)
- Bank has more “5” (high) scores

“Load Test” - Grocery vs. Bank



Points to Ponder:

- Does faster imply better?
- The serpentine line is ***socially fairer***, because customers who arrive first are always served first
- Most people view ***fairness*** as more important than waiting time
- Do people and computers have the same criteria?

Wait Time

- Grocery store is +1 minute faster (5:39 vs. 6:56)
- Grocery has wider range of wait time (2:10 – 9:21)

Satisfaction

- Bank has higher satisfaction (3.8 vs. 3.45)
- Bank has more “5” (high) scores

Bonus "Load Test" - Zombie Special (redux)

MythBusters – "*Volunteer Special*"
(Episode 242, Feb 6, 2016)

- **Myth**
  - *"An axe is a more effective weapon against a horde of zombies than a gun. A revisit of the 'Axe vs. Gun' myth from 2013."*
- **Add a chainsaw to our set of weapons**
  - Axe (confirmed as best in initial test)
  - Gun
  - Chainsaw (new weapon)

Investigate the effectiveness of a chainsaw against zombies

What's the new ranking?

Axe

Gun

Chainsaw

Bonus “Load Test” - Zombie Special (redux)

MythBusters – “*Volunteer Special*”

(Episode 242, Feb 6, 2016)

- **Myth**
  - *“An axe is a more effective weapon against a horde of zombies than a gun. A revisit of the ‘Axe vs. Gun’ myth from 2013.”*
- **Add a chainsaw to our set of weapons**
  - Axe (confirmed in initial test)
  - Gun
  - Chainsaw (new weapon)

Yes! A chainsaw is better than an axe | gun

Test Results:

1) **Chainsaw**
2) Axe
3) Gun

Hands-on Modeling - Market Open Response Time Analysis in R

Problem:

- Response time problem at market open (90^{th} > 8 min)
- Goal: 90^{th} percentile response time ≤ 1 minute
- How do we fix this performance problem?

Modeling Tool: queuecomputer

- Package for R
- Utilizes "queue departure computation"
- "… v*astly more computationally efficient than existing approaches to DES …*"

Reference:

- "*Computationally Efficient Simulation of Queues: The R Package queuecomputer*"
- Authors: Anthony Ebert, Paul Wu, Kerrie Mengersen, Fabrizio Ruggeri
- Queensland University of Technology
- 2017
- https://arxiv.org/abs/1703.02151

queuecomputer - Example (queue departure computation)


```
# Setup model
SIM_arrival_times <- c(5, 10, 12, 20)
SIM_service_times <- c(3,  5,  2,  4)

# Evaluate the model
SIM_results_raw <- queue_step(arrivals=SIM_arrival_times,
                              service=SIM_service_times,
                              servers=1)
```

Model Input:
- Per customer
  + Arrival time
  + Service time

Model Output:
- Response & wait times
- Etc.

		cust_1	cust_2	cust_3	cust_4	Average
INPUT	SIM_arrival_times	5	10	12	20	
INPUT	SIM_service_times	3	5	2	4	
	(compute departure times)					
OUTPUT	$ departures	8				
OUTPUT	$ system_time	3				
OUTPUT	$ waiting	0				

queuecomputer - Example (queue departure computation)


```
# Setup model
SIM_arrival_times <- c(5, 10, 12, 20)
SIM_service_times <- c(3,  5,  2,  4)

# Evaluate the model
SIM_results_raw <- queue_step(arrivals=SIM_arrival_times,
                              service=SIM_service_times,
                              servers=1)
```

Model Input:
- Per customer
  - Arrival time
  - Service time

Model Output:
- Response & wait times
- Etc.

		cust_1	cust_2	cust_3	cust_4	Average
INPUT	SIM_arrival_times	5	10	12	20	
INPUT	SIM_service_times	3	5	2	4	
		(compute departure times)				
OUTPUT	$ departures	8	15			
OUTPUT	$ system_time	3	5			
OUTPUT	$ waiting	0	0			

queuecomputer - Example (queue departure computation)


```
# Setup model
SIM_arrival_times <- c(5, 10, 12, 20)
SIM_service_times <- c(3,  5,  2,  4)

# Evaluate the model
SIM_results_raw <- queue_step(arrivals=SIM_arrival_times,
                              service=SIM_service_times,
                              servers=1)
```

Model Input:
- Per customer
  - + Arrival time
  - + Service time

Model Output:
- Response & wait times
- Etc.

		cust_1	cust_2	cust_3	cust_4	Average
INPUT	SIM_arrival_times	5	10	12	20	
INPUT	SIM_service_times	3	5	2	4	
	(compute departure times)					
OUTPUT	$ departures	8	15	17		
OUTPUT	$ system_time	3	5	5		
OUTPUT	$ waiting	0	0	3		

queuecomputer - Example (queue departure computation)

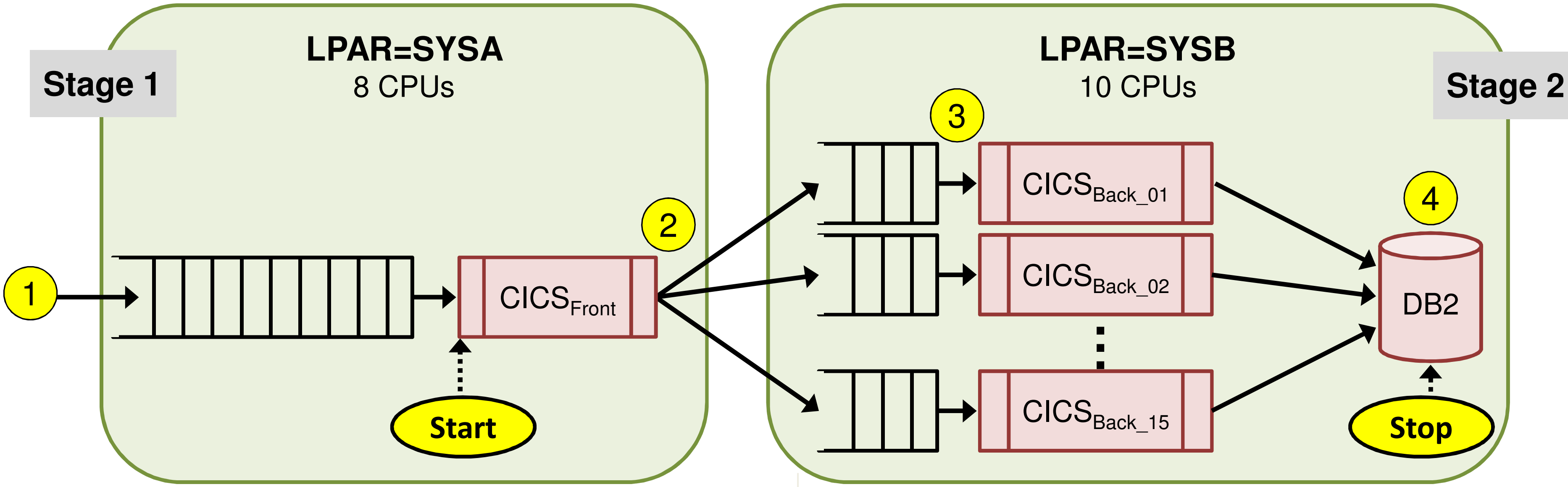

```
# Setup model
SIM_arrival_times <- c(5, 10, 12, 20)
SIM_service_times <- c(3,  5,  2,  4)

# Evaluate the model
SIM_results_raw <- queue_step(arrivals=SIM_arrival_times,
                              service=SIM_service_times,
                              servers=1)
```

Model Input:
- Per customer
  - Arrival time
  - Service time

Model Output:
- Response & wait times
- Etc.

		cust_1	cust_2	cust_3	cust_4	Average
INPUT	SIM_arrival_times	5	10	12	20	
INPUT	SIM_service_times	3	5	2	4	
		(compute departure times)				
OUTPUT	$ departures	8	15	17	24	
OUTPUT	$ system_time	3	5	5	4	4.25
OUTPUT	$ waiting	0	0	3	0	0.75

Application Architecture & Workflow

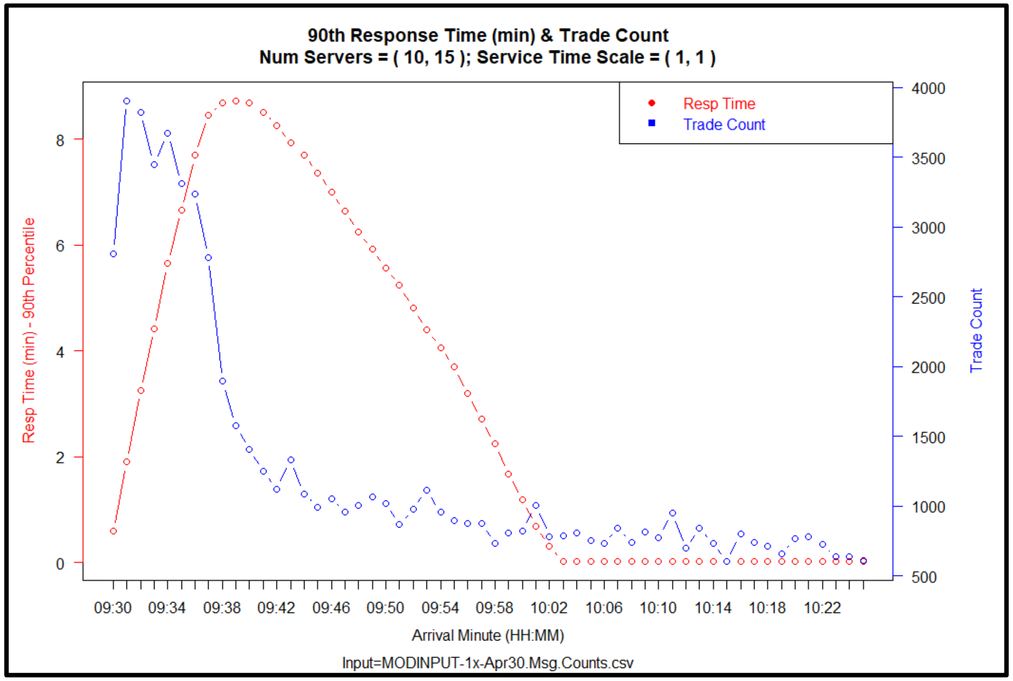


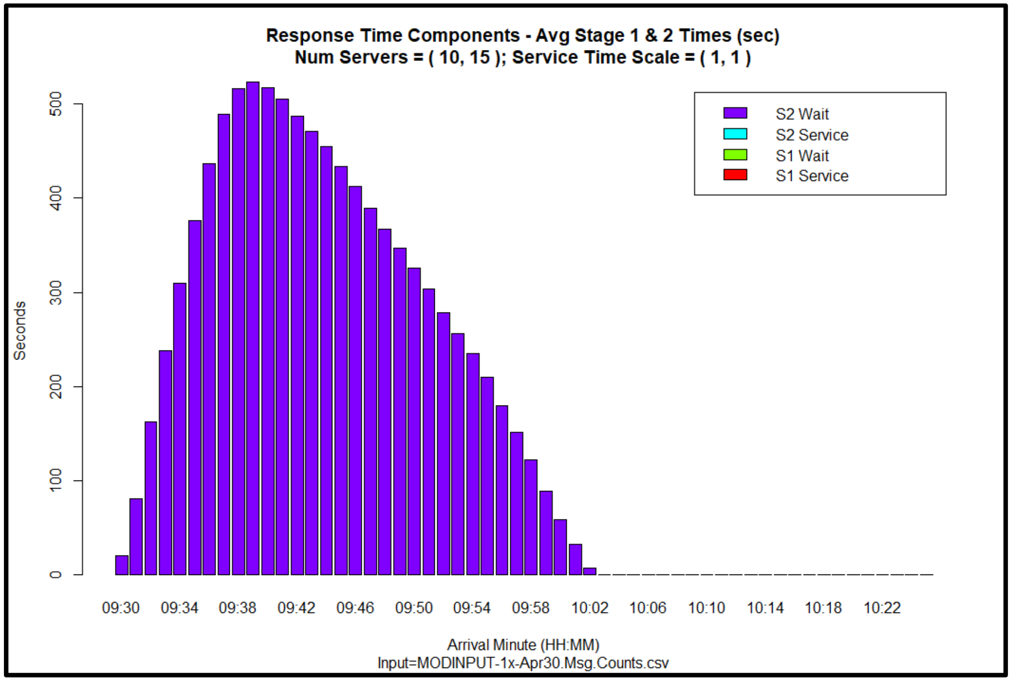
1. Trades enter the single FIFO queue serviced by CICSFront.
2. CICSFront processes a Trade and forwards it to one of CICSBack queues (round-robin)
3. CICSBack pulls a Trade message, does some processing and then posts it to the backend DB2 database
4. After posting to DB2 the trade can be executed (Buy or Sell)

Trade statistics recorded per arrival minute

- Arrival minutes: [9:30,9:31), [9:31,9:32), etc.
- Number of trade entering per minute
- Response times reported per arrival minute

Time for some modeling …





Left: 90th percentile response time in minutes

Right: Number of trades arriving per minute

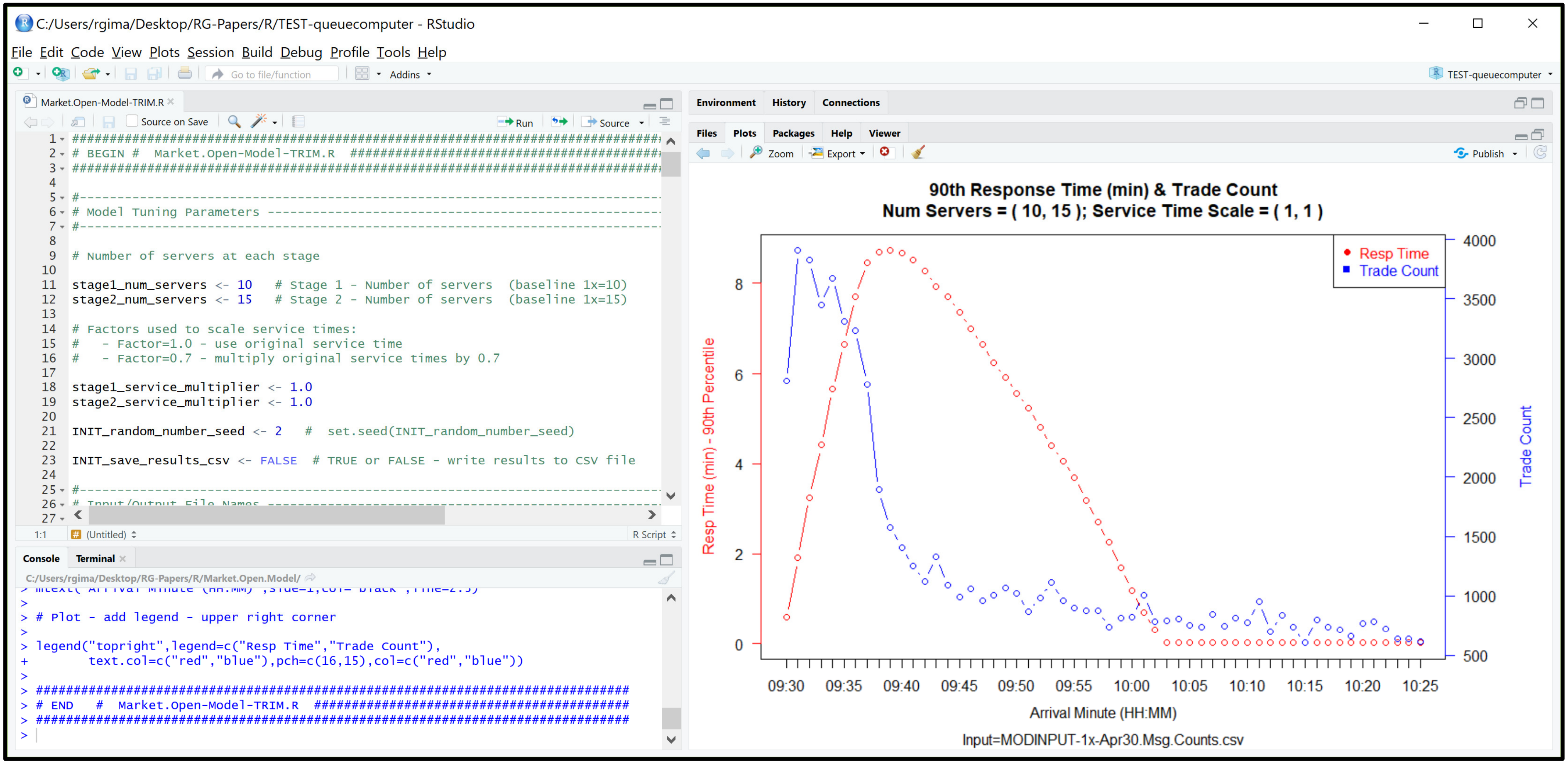
Model tuning knobs:

- Number of servers per stage
- CPU speed

Stacked bar chart of seconds spent in:

- Stage 1 & 2 – wait & service

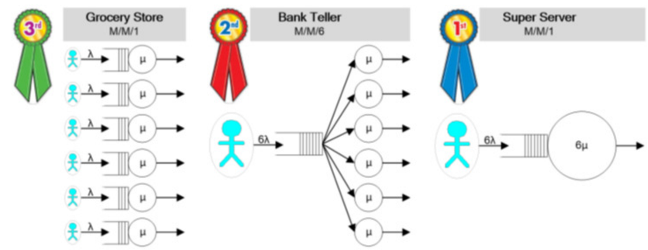
RStudio – Open Source IDE for R



What have we looked at for the past 30 minutes?

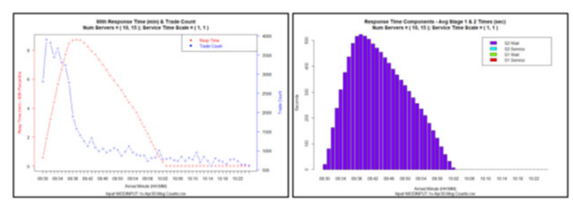
"*Waiting & Queues - People vs. Computers*"

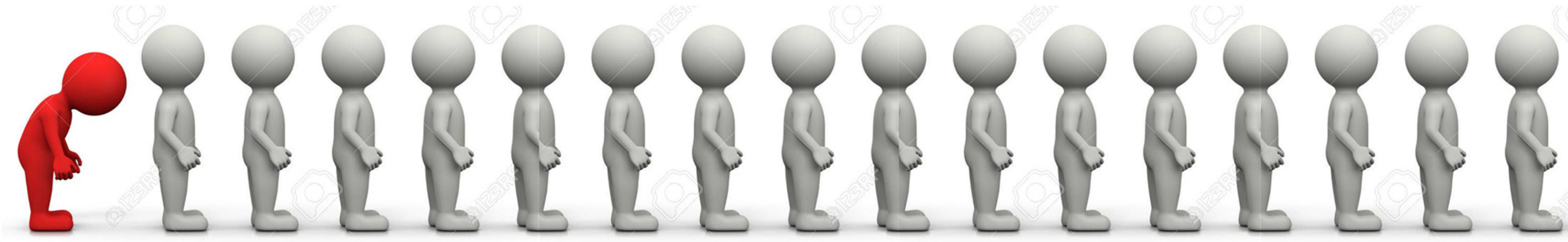
- **Examined three "equivalent" queues**
  - Intuition & queuing theory helped us rank the queues
  - Theory does not always match reality (MythBusters)
  - People & computers view waiting differently



- **Defending yourself against a horde of zombies**
  - Relative efficiency ranking: (1) chainsaw, (2) axe, (3) gun
  - And yes, there are web sites dedicated to this subject

- **Hands-on modeling to solve a real queuing problem**
  - Used R to model a "market open" application
  - Demonstrated the efficiency (speed) of "queuecomputer"
  - Modeling helps to confirm our intuition and provide directional information





Waiting & Queues
People vs. Computers

Richard Gimarc
rgimarc@featherfall.com

Computer Measurement Group
SOUTHWEST

September 19, 2018
Southwest CMG