Choose the Right Hardware

Proposal

By :- Aditya Srivastava

Scenario 1: Manufacturing

Client Requirements and Potential Hardware Solution

Look through the scenario and find any relevant client requirements. Then, suggest a potential hardware type and explain how this hardware would satisfy each of the requirements.

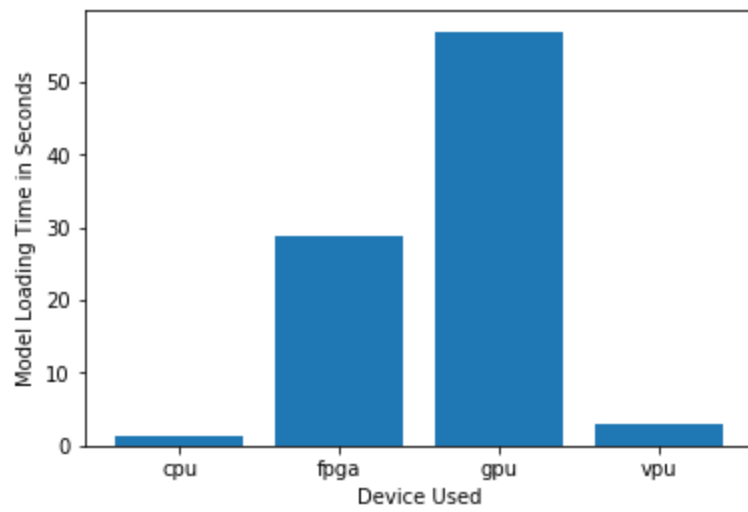| Which hardware might be most appropriate for this scenario?
(CPU / IGPU / VPU / FPGA) |
| --- |
| *FGPA* |

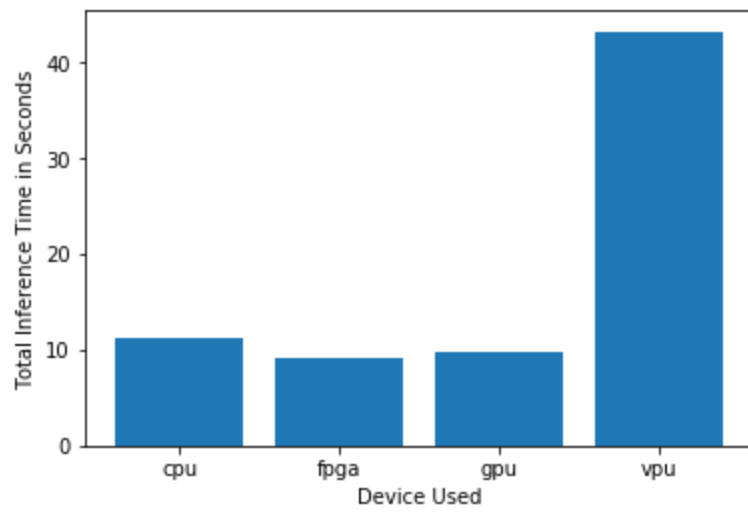| Requirement Observed (Include at least two.)	How does the chosen hardware meet this requirement?
Good Durability (5-10 years)	*FGPAs last that long, we can trust quality and durability there*
30-35 FPS with minimum 5 images per second processing	FGPAs outperform other devices and will be able to meet these requirements
No cost restrictions	FGPAs even if expensive, won't be a problem in this scenario

Queue Monitoring Requirements

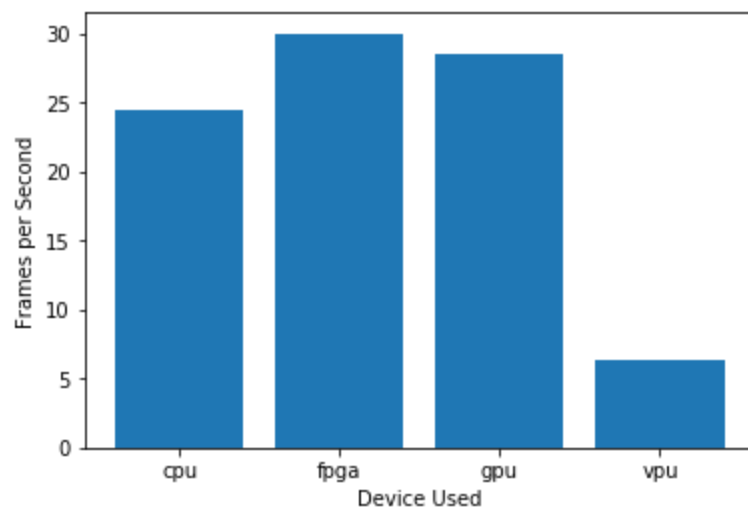| Maximum number of people in the queue	*5*
Model precision chosen (FP32, FP16, or Int8)	*FP16*

Test Results

After you've tested your application on all four hardware types (CPU, IGPU, VPU, and FPGA), copy the matplotlib output showing the comparison into the spaces below. You should have three graphs (for model load time, inference time, and FPS).

Model Load Time

Inference Time

FPS

Final Hardware Recommendation

Now synthesize your points from above and provide a brief write-up describing why the chosen hardware is the best choice for this scenario. Be sure to discuss the client's requirements, the test results, and how these relate to one another (e.g., perhaps one of the devices performed better than the rest, but does not meet one of the client's requirements).

Write-up: Final Hardware Recommendation
*As seen in the graphs above, FPGA shows the best performance. It **meets the** client's 30-35 FPS **requirement**. Even if it is a little slow in loading the model, but **least inference time** and **highest FPS** cover for the loading time. So **FPGA** is still my hardware recommendation.*

Scenario 2: Retail

Client Requirements and Potential Hardware Solution

Look through the scenario and find any relevant client requirements. Then, suggest a potential hardware type and explain how this hardware would satisfy each of the requirements.

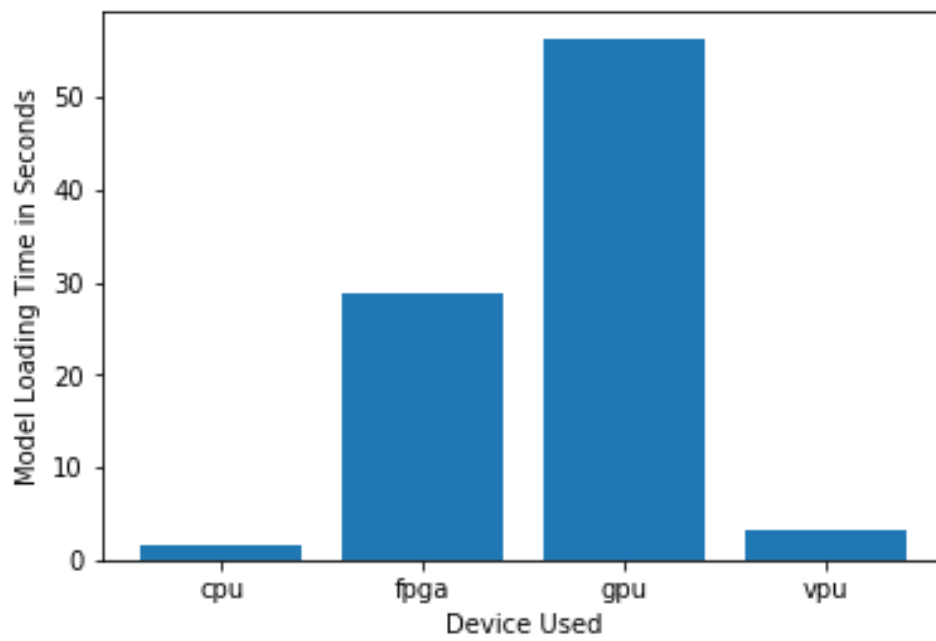| Which hardware might be most appropriate for this scenario? (CPU / IGPU / VPU / FPGA) |
|---|
| *CPU* |

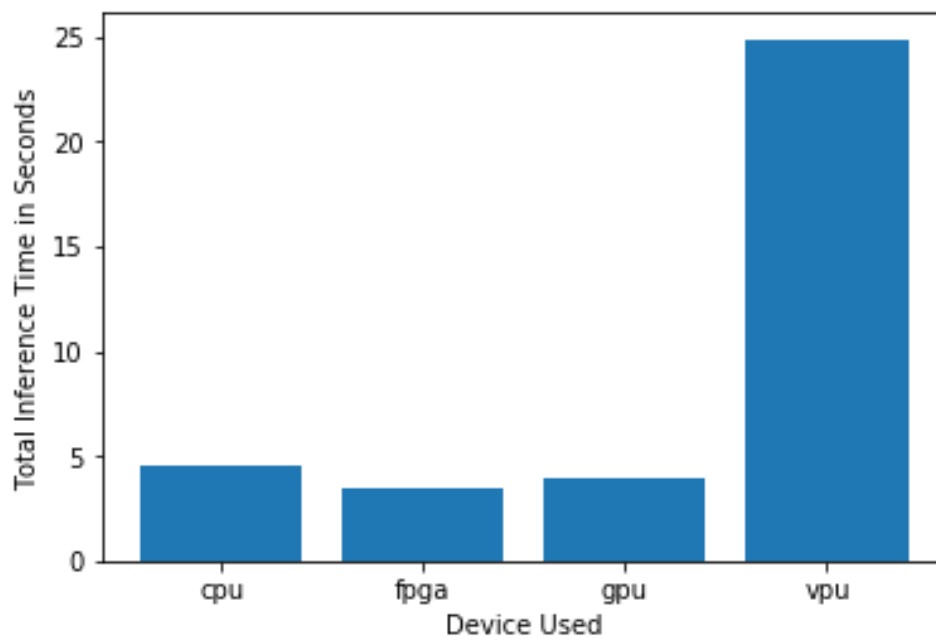| Requirement Observed (Include at least two.)	How does the chosen hardware meet this requirement?
Less money, can't invest in Hardware	Already i7 core processors available
Need during rush hours, good FPS and processing needed	*Since computers are already used, runnin*

Queue Monitoring Requirements

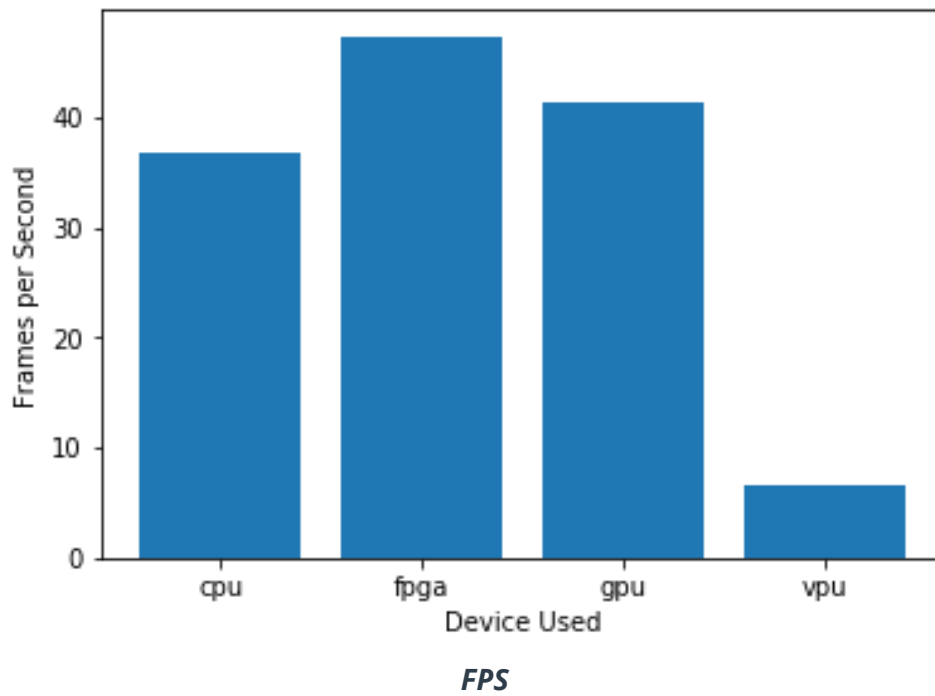| Maximum number of people in the queue	*5*
Model precision chosen (FP32, FP16, or Int8)	*FP16*

Test Results

After you've tested your application on all four hardware types (CPU, IGPU, VPU, and FPGA), copy the matplotlib output showing the comparison into the spaces below. You should have three graphs (for model load time, inference time, and FPS).

Model Load Time

Inference Time

FPS

Final Hardware Recommendation

Now synthesize your points from above and provide a brief write-up describing why the chosen hardware is the best choice for this scenario. Be sure to discuss the client's requirements, the test results, and how these relate to one another (e.g., perhaps one of the devices performed better than the rest, but does not meet one of the client's requirements).

Write-up: Final Hardware Recommendation
*As you can see, the **CPU** does give a **good performance** and even **model load time is least**, even the inference time is not bad. No extra hardware needed so this makes a good case. I would also like to mention that **IGPUs** are a good option too. Though there will be some **initial hardware expenses** and **model load time** is **highest** but that is only a **one time thing**, after that IGPUs perform really well and even their FPS is quite high, also they have **configurable power consumption** so electric bill could be saved. So it's a tough choice between CPUs and IGPUs if a **little initial investment is affordable, I'd recommend IGPUs** as they can save money on electric bills which is a monthly affair in most places and compared to an initial investment, in the long run IGPUs will be profitable. If the **client is not ready to spend** even a little on hardware then CPUs would work too.*

Scenario 3: Transportation

Client Requirements and Potential Hardware Solution

Look through the scenario and find any relevant client requirements. Then, suggest a potential hardware type and explain how this hardware would satisfy each of the requirements.

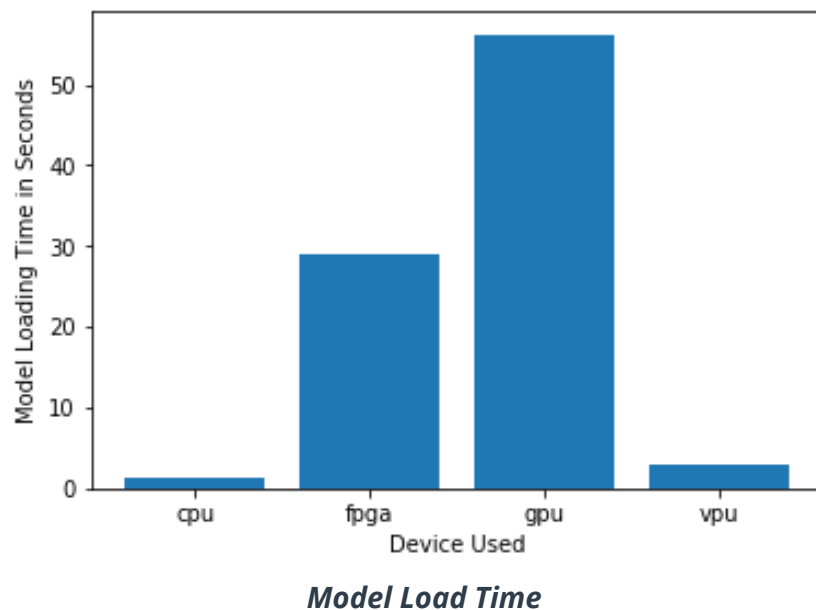| Which hardware might be most appropriate for this scenario? (CPU / IGPU / VPU / FPGA) |
|---|
| VPU |

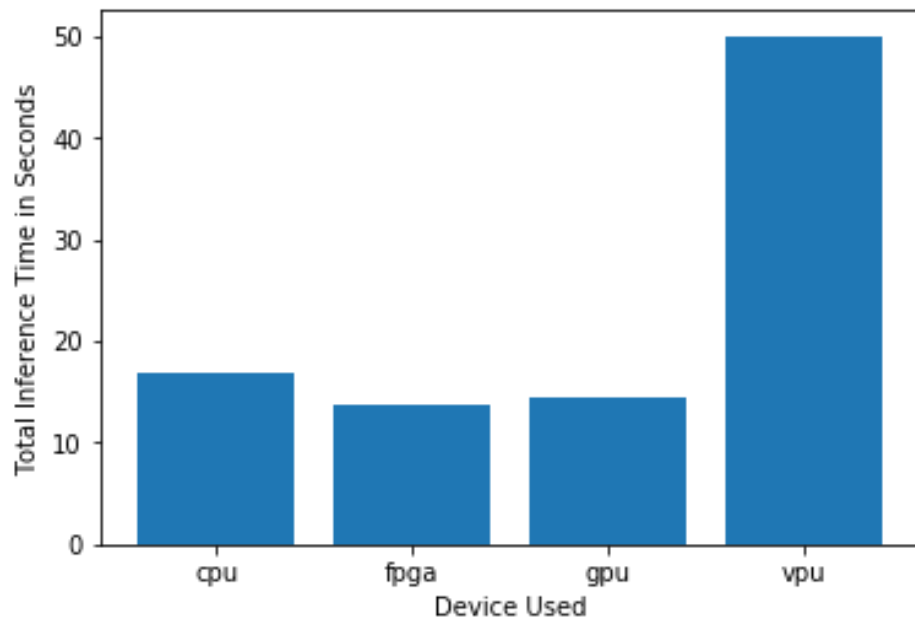| Requirement Observed (Include at least two.)	How does the chosen hardware meet this requirement?
Maximum of $300 per machine (low budget)	VPUs would fit in the price range.
Save both on hardware and future power requirements.	Power Consumption of a VPU is not high as typically it is a small USB attachment.

Queue Monitoring Requirements

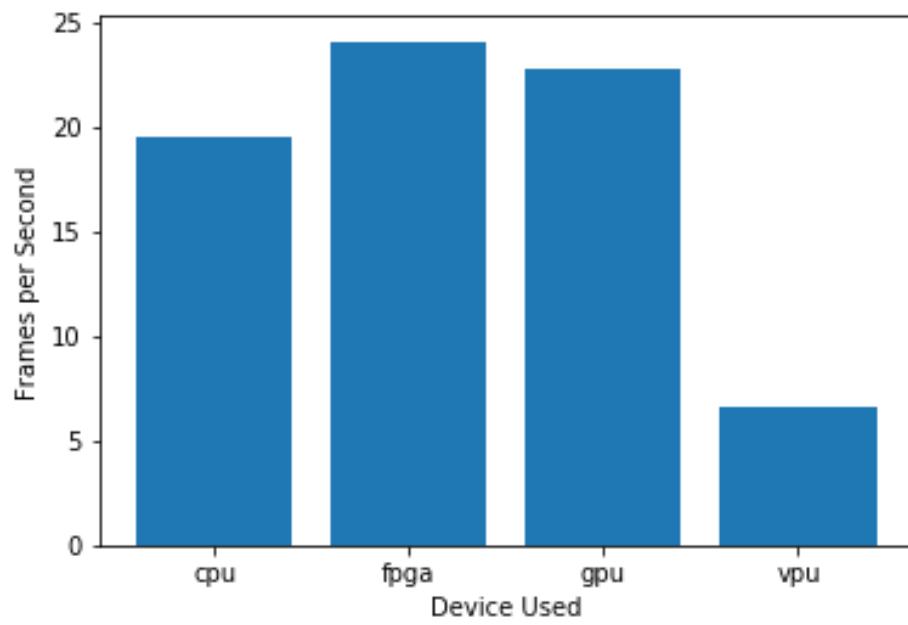| Maximum number of people in the queue	15
Model precision chosen (FP32, FP16, or Int8)	FP16

Test Results

After you've tested your application on all four hardware types (CPU, IGPU, VPU, and FPGA), copy the matplotlib output showing the comparison into the spaces below. You should have three graphs (for model load time, inference time, and FPS).

Model Load Time

Inference Time

FPS

Final Hardware Recommendation

Now synthesize your points from above and provide a brief write-up describing why the chosen hardware is the best choice for this scenario. Be sure to discuss the client's requirements, the test results, and how these relate to one another (e.g., perhaps one of the devices performed better than the rest, but does not meet one of the client's requirements).

Write-up: Final Hardware Recommendation
*The inference time of VPU is high and fps are comparatively lesser than other devices but It will still meet the requirement of the client to handle 7 to 15 people in a queue without spending money on the new hardware for each computer. A simple plugin **VPU** is **the best choice** for this scenario*