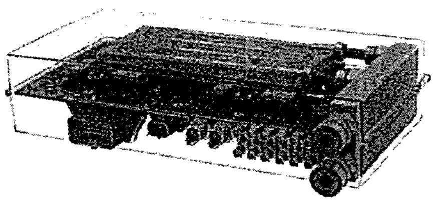
※ 注意：請於試卷內之「非選擇題作答區」依序作答，並應註明作答之大題及小題題號。

PART I: COMPUTER ARCHITECTURE

1. (25 pts) The advance of computer architecture is enabling big data analytics, machine learning and new smart applications. Please read the following article before answering the questions:

Nvidia Announces New Drive PX 2 'Supercomputer in A Lunchbox' For Self-Driving Cars
(Source: techcrunch.com, 2016/01/04)

Following last year's Drive CX, Nvidia just announced an updated version of its computing platform for self-driving cars, the Drive PX 2. Compared to last year's model, this is a much more powerful beast, able to process data from 12 video cameras and other sensors in real time to make educated driving decisions. The company calls it a supercomputer, and it's the size of a lunchbox.



Behind the scene, the Drive PX 2 features 12 different cores representing 8 teraflops of calculation power, 24 deep learning tera operations per second. The company is using a 16nm architecture, and it's a hungry beast as it requires 250W of power. Finally, Nvidia is using a liquid-cooling system. Because the PX 2 will be used in cars, that's not too much of a problem and Nvidia argues that car manufacturers will just plug the device into their existing cooling solutions.

Nvidia co-founder and CEO Jen-Hsun Huang said that the PX 2 was as powerful as 150 MacBook Pros. The company is certainly comparing GPU power with the 13-inch MacBook Pro, which currently sports an Intel Iris Graphics 6100 chip. The PX 2 features two next-gen Tegra processors, as well as a Pascal-based GPU. In total, the system can push up to 8 teraflops and recognize up to 2,800 images per second using the AlexNet neural network-based deep learning algorithm.

見背面

The company also announced Nvidia Drivenet, its own deep neural network. It has the equivalent of 37 million neurons and was trained 120 million objects so far. Like other deep neural networks, it gets better over time.

Companies will be able to leverage this network, but Nvidia also insisted in saying that car makers would want to control their own neural network.

The company is promoting a platform approach, meaning that it wants to work with as many car makers as possible to kickstart their self-driving efforts. Volvo is going to be the first partner shipping Drive PX 2 in about 100 upcoming test cars. Nvidia has also partnered with Audi, Daimler, BMW and Ford to develop and test the PX 2.

As Huang showed during today's keynote, all of this power is needed to enable self-driving cars to know enough about their environment — and interpret it — to safely drive in traffic. In one demo, Nvidia showed a new dashboard that can use this data to show drivers whether there are other cars around them at any given time. Thanks to this, "we soon won't need rear-view mirrors anymore," Huang joked.

As Nvidia clearly noted, though, identifying objects and planning basic paths is only the beginning. Looking ahead, making self-driving cars work in the real world also means that cars can recognize circumstances as well. Not every truck is the same, after all — some are ambulances, for example, and you better make space for those.

Please answer the following questions:

(a) [2 pts] Nvidia is famous for making Graphics Processing Units (GPU's). Can GPU's be used for general purpose computing? How?

(b) [2 pts] Which category does this PX 2 platform belongs: desktop, server, embedded, or supercomputer?

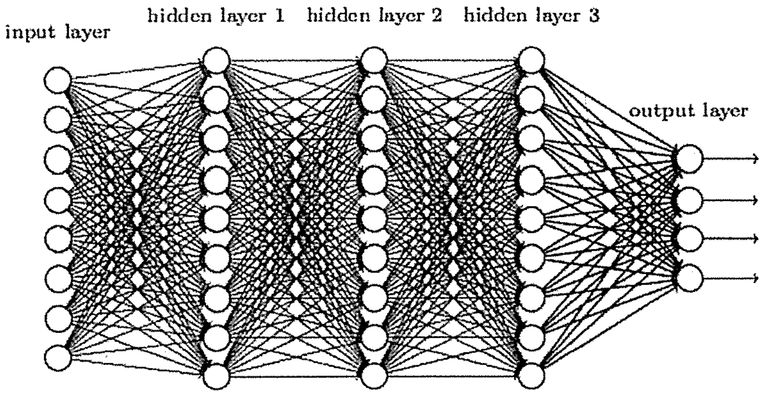
(c) [2 pts] What is the power efficiency of the PX 2 platform, in terms of teraflops/watt?

(d) [2 pts] Nvidia said that the PX 2 was as powerful as 150 MacBook Pros. Why don't we use it to make desktop computers? (In addition to answering yes/now, please give an explanation).

接次頁

題號： 419

The PX 2 is designed to accelerate "deep learning" or "deep neural network" algorithms with GPU's. The following figure is an example of deep neural network, where data are received by the input layer from the left, processed through 3 hidden layers, and finally delivered to the output layer to produce the results. Each circle represents a functional node which receives multiple input data from the previous layer, computes a result with a function, and deliver the result to the nodes in the next layer.



(e) [2 pts] Please explain why GPU may be useful for accelerating such an algorithm?
(f) [2 pts] Can the deep neural network shown above be implemented by a pipelined datapath? Draw a picture to show how the pipelined data path works.
(g) [2 pts] How would pipelining affect the performance, in terms of throughput and latency?
(h) [2 pts] As opposed to Nvidia's GPU approach, do you think it is a good idea to design a chip entirely for deep learning? What would be the advantages and disadvantages?
(i) [2 pts] For a very large deep neural network, we may want to accelerate it with multiple machines by partitioning the input data. Suppose we have two machines connected with Ethernet, and each machine receives half of the input data. For simplicity, assume every layer has N nodes and each node sends its output data (1 floating point number) to every node in the next layer. How many floating point numbers need to be exchanged between the two machines for one set of input data?
(j) [2 pts] Following the previous question, suppose each node takes A

見背面

nanoseconds to complete its function, each machine has Q processor cores, and the Ethernet has 10Gbps duplex bandwidth (i.e. each direction can transfer at 10Gbits/sec). Please estimate the speedup of the execution time with two machines. (Ignore the network delay and the other overhead in this case).

(k) [2 pts] Following the previous question, if we have P machines instead of 2 machines, please estimate the limit of speedup of the execution time, as P approaches infinity.

(l) [3 pts] Based on the above, please discuss how you would design system architectures for small, medium, and large deep neural network for best cost-performance? You may use CPUs, GPU, custom-designed chips, FPGA, clusters, etc.

2. (10 pts) Please answer the following questions. Note that if you don't explain why, you get 0 point.
   (a) [2 pts] Do bus reads take longer than bus writes in Write-Invalidate Bus-Snooping Protocol for write-through cache? Answer YES or NO. Explain why.
   (b) [2 pts] You are using a cache with a block size of eight words, with a single word alternatively written and read by two processors. Assume both processors are accessing different variables. Will this cause false sharing? Answer YES or NO. Explain why.
   (c) [2 pts] Understanding interface is essential in studying Computer Organization. Computer Organization is about hardware/software interface. In comparison, on 12-30-2015, Google announced that next-generation Android will use OpenJDK for Java APIs (Application Programming Interface). Is Application Binary Interface (ABI) higher level than API? Answer YES or NO. Explain why you answer this way.
   (d) [2 pts] When Google decided to use Java Virtual Machine for Android, did Google make abstraction level (or interface level) higher than C? Answer YES or NO, and explain why.
   (e) [2 pts] Note that Android devices mostly use ARM processors. Does Java Virtual Machine help Android port to different architectures? Answer YES or NO, and explain why.

3. (11 pts) Draw a finite state transition diagram for Snooping Protocols for multiprocessors (from CPU angle). Please use 3 states (Invalid, Shared, Exclusive) and 4 CPU request types (CPU local-read request, CPU remote-read request, CPU local-write request, and CPU remote-write request). When you observe a

接次頁

performance problem due to false sharing, how to avoid false sharing?

4. (4 pts) On 2016-01-28, the editorial of Commercial Times (工商時報) expresses the urgency for Taiwan to apply the Blockchain technology. Two popular blockchains are (1) Bitcoin blockchain and (2) Ethereum, which are briefly explained below.
(1) Bitcoin blockchain needs repeated invocation of the following hashing computation, in pseudo code. Note that you don't need to understand the details of each line below.

```
Process the message in successive 512-bit chunks:
break message into 512-bit chunks
for each chunk
    create a 64-entry message schedule array w[0..63] of 32-bit words
    copy chunk into first 16 words w[0..15] of the message schedule array
    for i from 16 to 63
        s0 := (w[i-15] rightrotate 7) xor (w[i-15] rightrotate 18)
              xor (w[i-15] rightshift 3)
        s1 := (w[i-2] rightrotate 17) xor (w[i-2] rightrotate 19)
              xor (w[i-2] rightshift 10)
        w[i] := w[i-16] + s0 + w[i-7] + s1

    Initialize working variables to current hash value:
    a := h0
    b := h1
    c := h2
    d := h3
    e := h4
    f := h5
    g := h6
    h := h7

    Compression function main loop:
    for i from 0 to 63
        S1 := (e rightrotate 6) xor (e rightrotate 11)
              xor (e rightrotate 25)
        ch := (e and f) xor ((not e) and g)
        temp1 := h + S1 + ch + k[i] + w[i]
        S0 := (a rightrotate 2) xor (a rightrotate 13)
```

見背面

```
        xor (a rightrotate 22)
maj := (a and b) xor (a and c) xor (b and c)
temp2 := S0 + maj

h := g
g := f
f := e
e := d + temp1
d := c
c := b
b := a
a := temp1 + temp2
```

(2) Ethereum needs repeated invocation of the hashing computation that is memory hard, making it basically ASIC (Application-Specific Integrated Circuit) resistant. The code is irregular and involves graph building:

```
def produce_dag(params, seed, length):
    P = params["P"]
    picker = init = pow(sha3(seed), params["w"], P)
    o = [init]
    for i in range(1, length):
        x = picker = (picker * init) % P
        for _ in range(params["k"]):
            x ^= o[x % i]
        o.append(pow(x, params["w"], P))
    return o
```

(Details are on github:
https://github.com/ethereum/wiki/blob/master/Dagger-Hashimoto.md
https://github.com/ethereum/go-ethereum/tree/master/pow/dagger)

Please answer the following questions:

(a) [2 pts] Compared with Ethereum, is Bitcoin blockchain's hashing more suitable for GPU (or ASIC)?

(b) [2 pts] Is Ethereum more suitable for CPU than GPU?

接次頁

PART II – OPERATING SYSTEMS

NOTE that in the question, it is intended to provide redundant or miss certain assumption to disguise you. Please make your own assumption if necessary to answer the questions.

Bob is designing his cloud-based storage services, called BobBox and similar to Dropbox, to store the files on his desktop computer, laptop computer, and smart phones to a single cloud storage. Later, he will also open this service for his friends. Hence, the storage services will support multiple users and multiple devices. The service consists of two applications: BoxBoxClient (BBC) on the client devices and BobBoxServer (BBS) on the server. Please answer the following questions.

5. (3 pts) One shortcoming of existing Dropbox service is conflict resolving: many conflicted files are stored when file folder is shared with multiple users and multiple devices. Please describe the possible causes of the conflicts among file synchronization. Hint: you may consider the connectivity between BBS and BBC, and time instances to synchronize the files on BBS and BBC.

6. (16 pts) BBC has several options to transmit the files on client devices to BBS. On BBC, the files are read from local storage devices and send to network handler.
   (a) [8 pts] There are two types of Input/Output (IO) libraries provided by operating systems: <u>Buffered IO</u> and <u>Unbuffered IO</u>. Please describe these two types of the IO libraries (4 points) and compare the number of memory copy operations from storage subsystem to application memory (4 points).
   (b) [8 pts] BBC should upload/download several files to/from the server:
      i. [2 pts] Should the access to local storage system be sequential access or random access? Please explain your answer.
      ii. [6 pts] BBC may use <u>blocking synchronous IO</u> or <u>non-blocking asynchronous IO</u> models to access storage systems. Please describe the above two IO models and choose one to increase application throughput, i.e., avoiding unnecessary waiting and operations.

7. (6 pts) BBS needs to serve concurrent synchronization requests from multiple devices and multiple accounts. Suppose multiple threads are used in BBS to serve concurrent requests. Please describe thread pool model and explain how it may limit the resources used by BBS while serving concurrent requests.

8. (20 pts) Suppose a thread serves only a request. To avoid conflicted files, some file locking mechanism might be applied to block threads not holding the lock.

見背面

After the file is unlocked, all blocked threads start to compete for the lock again.

(a) [10 pts] A smart engineer designs a barrier data type of 2 semaphores and a counter and a **barrier_wait** function to help design the competing mechanism so that all threads calling **barrier_wait** will be blocked and all blocked threads will be unblocked when the number of blocked threads reaches n. Please complete the following codes for barrier and **barrier_wait** in C-like language where mutex(1) stands for the semaphore mutex is initially 1.

```
typedef struct {
semaphore mutex(1), …
…
} barrier;

barrier_wait (barrier *b, int n) {
…
}
```

(b) [10 pts] For the unblocked threads to compete for the lock with the most user satisfaction, especially for good friends, first come first serve might not be the best policy, however, no one would like to wait for a long time. The priority of friends might change dynamically depending on which files are requested. How do you design your scheduling policy? (5 points) How do you implement your policy? (5 points) Hint: describe which parts in the system, application, or else need to be modified. No code is needed.

9. (5 pts) Since the files might be shared by friends, how do you design the access control mechanism to be user-friendly and very efficient, especially the files might be accessed in different devices with different sizes of memory, storage, and screen. Hint: different friends might have different access rights on different files.

試題隨卷繳回