IMPERIAL COLLEGE LONDON

TIMED REMOTE ASSESSMENTS 2020-2021

BEng Honours Degree in Computing Part III
BEng Honours Degree in Electronic and Information Engineering Part III
MEng Honours Degree in Electronic and Information Engineering Part III
MEng Honours Degree in Electronic and Information Engineering Part IV
BEng Honours Degree in Mathematics and Computer Science Part III
MEng Honours Degree in Mathematics and Computer Science Part III
MEng Honours Degrees in Computing Part III
MSc Advanced Computing
MSc Computing
MSc in Computing (Specialism)
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant assessments for the Associateship of the City and Guilds of London Institute

PAPER COMP60017=COMP97103=COMP97104

PERFORMANCE ENGINEERING

Thursday 25 March 2021, 10:00
Duration: 140 minutes
Includes 20 minutes for access and submission

Answer ALL TWO questions
Open book assessment

Paper contains 2 questions

1 a Consider a 2^{6-1} fractional factorial design with factors A, B, C, D, E, and F, each having two levels.

i) How many experiments would you need to run on the system to determine the sum-of-squares total (SST) value for this design? Briefly justify your answer.

ii) Observe that, for this design, a confounding is given by $F = ABCDE$. Use this information to find the confoundings for the terms A, EF, and CDF.

iii) Using the confounding structure determined in the previous question, indicate the resolution of this design. Briefly explain your answer.

iv) Consider the rows of the sign table for this design where factors A, B, C, and D are set to their high level. What are the possible values for factors E and F in these rows? Justify your answer.

b An autoregressive model of order 2, for short AR(2), characterizes a time-series as a stochastic difference equation

$$A_t = c + \phi_1 A_{t-1} + \phi_2 A_{t-2} + \varepsilon_t$$

where A_t is a random variable for the time-series value at time t, ε_t indicates a white noise, and c, ϕ_1 and ϕ_2 are scalar model parameters. Assuming that A_t is stationary, determine a relation between the parameters c, ϕ_1 and ϕ_2 and the mean μ of A_t.

c For each of the following statements, indicate whether it is *True* or *False*. Write, for each statement, a brief paragraph to justify your answer.

i) Right after increasing the injection rates, an enterprise benchmark should immediately start monitoring the steady-state performance metrics.

ii) Time-varying workload mixes may lead to the formation of millibottlenecks.

iii) Resource throttling should be used both when a resource is scaled out and when it is scaled down.

iv) An optimal system controller aims at minimizing the spectral radius of the input-output transfer function.

v) Typical user behaviour graphs for a website can be generated automatically from HTTP log files.

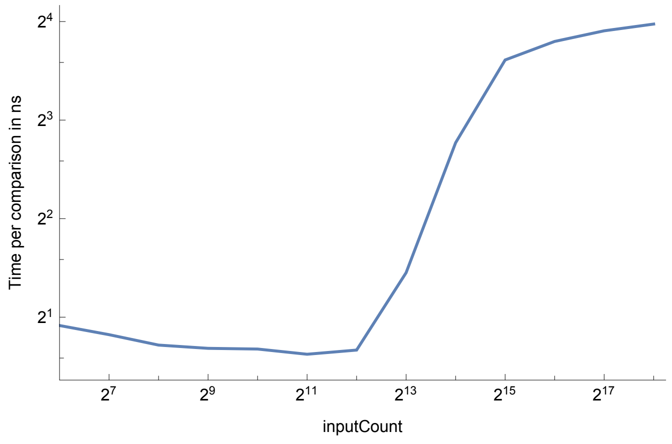
The three parts carry, respectively, 30%, 20%, and 50% of the marks.

2 a Name three exemplary situations in which a system is unbalanced and how you would recognize that fact through a microarchitectural profiling? You can describe your examples either as code snippets or in text.

b The following hot code section:

```
for(auto i = 0U; i < inputCount; i++)
  for(auto j = 0U; j < inputCount; j++)
    if(input[i] == input[j])
      count++;
```

produces the following performance graph varying `inputCount` (assume the values in the input are unique but shuffled)



i) Explain the performance behavior of the experiment.

ii) Can this experiment be modeled using the access pattern algebra? If so, how would you model it? If not, how could you approximate it?

iii) Describe how performance would change if the number of duplicates in the input was increased.

c Memory management in code can be performed in (at least) three different manners:

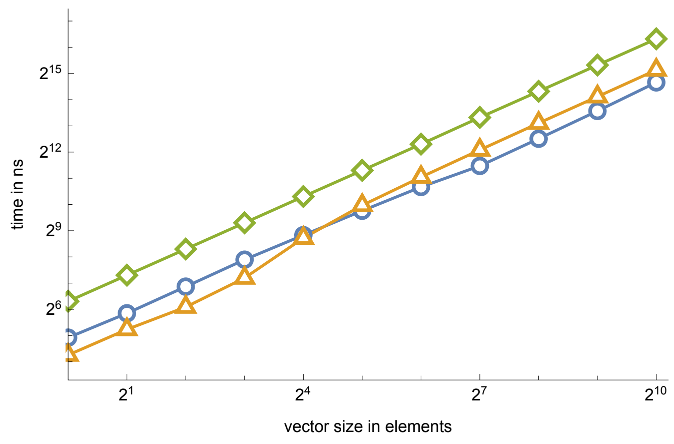
- Manually, i.e., calling **delete** or free when memory is no longer required
- Through Reference Counting, i.e., freeing allocated memory at the time that it is in no further use
- Through Garbage Collection, i.e., occasionally traversing the object graph, identifying and deleting objects that are no longer in use

The following snippets illustrate each of these approaches

```
extern vector<char*> manualVector;
static void Manual(benchmark::State& state) {
  for(auto _ : state) {
    for(auto i = 0U; i < state.range(0); i++)
      a[i] = (new char[2]{});
    for(auto& it : a)
      delete it;
  }
}
extern vector<shared_ptr<char*>> refCountVector; // shared_ptr!!!
static void ReferenceCounting(benchmark::State& state) {
  for(auto _ : state)
    for(auto i = 0U; i < refCountVector.size(); i++)
      a[i] = new char[2]; // shared_ptrs use reference counting
}

extern vector<char*, gc_allocator<char*>> gcVector;
void GarbageCollected(benchmark::State& state) {
  for(auto _ : state)
    for(auto i = 0U; i < gcVector.size(); i++)
      a[i] = (char*)GC_malloc(2); // using the boehm garbage collector
}
```

i) The following chart illustrate the performance of each of these approaches when varying the vector size. Which of the approaches corresponds to which line? Justify your answer.



ii) Do you think the size of the allocation plays a role for the relative performance of the benchmarks? If so, what role?

d Service Level Agreements (SLAs) are common for software but virtually never applied in CPU design. Why do you think that is? Describe challenges and opportunities that a hardware SLAs would have.

The four parts carry, respectively, 25%, 30%, 20%, and 25% of the marks.