**Lab 5**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lab Pair** | **Hostnames** | **Lab Pair** | **Hostnames** |
| Zimu | soctf-pdc-001  soctf-pdc-009 | Chia Ler  Samuel | soctf-pdc-006  soctf-pdc-014 |
| Zheng Wen  Kang Ming | soctf-pdc-002  soctf-pdc-010 | Kevin  Yong Zhi | soctf-pdc-007  soctf-pdc-015 |
| Aiden  Benedict | soctf-pdc-003  soctf-pdc-011 | FFA (node 8 may be down) | soctf-pdc-008  soctf-pdc-016 |
| Phi Long  Bertrand | soctf-pdc-004  soctf-pdc-012 | FFA | soctf-pdc-018  soctf-pdc-019 |
| Teng Le  Guozhen | soctf-pdc-005  soctf-pdc-013 | FFA | soctf-pdc-020  soctf-pdc-021 |

|  |
| --- |
| **Announcements (from Prof/TA):**  If you have not attempted last week’s introductory lab to OpenMPI, please do so instead of this week’s content (which follows up on collective communication operations in OpenMPI). |
| If you have questions: post your question below as follows (so we can join your discussion room):  **[Room X] I need help with <...>?**  [alternatively, you can request for help in Zoom under **Breakout Rooms > Ask for Help**] |
|  |

**Lab 4**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lab Pair** | **Hostnames** | **Lab Pair** | **Hostnames** |
| Zimu | soctf-pdc-001  soctf-pdc-009 | Chia Ler  Samuel | soctf-pdc-006  soctf-pdc-014 |
| Zheng Wen  Kang Ming | soctf-pdc-002  soctf-pdc-010 | Kevin  Yong Zhi | soctf-pdc-007  soctf-pdc-015 |
| Aiden  Benedict | soctf-pdc-003  soctf-pdc-011 | FFA | soctf-pdc-008  soctf-pdc-016 |
| Phi Long  Bertrand | soctf-pdc-004  soctf-pdc-012 | FFA (dual-socket nodes) | soctf-pdc-018  soctf-pdc-019 |
| Teng Le  Guozhen | soctf-pdc-005  soctf-pdc-013 | FFA (node 20 is down) | soctf-pdc-020  soctf-pdc-021 |

|  |
| --- |
| **Announcements (from Prof/TA):**  Please set up your SSH keys for the machine you’re working off if you have not done so already. See Tutorial 1 for instructions.  If you run a program with mpirun and it *appears to stall* in the terminal, check if: (1) you have copied your SSH key to the remote node and (2) you have copied the executable to the remote node. **Otherwise, this indicates you encrypted your SSH key with a password (passphrase)**.  To fix this, run  $ eval $(ssh-agent)  $ ssh <remote node> // type password - will unlock the key for this session  Then try again. |
| If you have questions: post your question below as follows (so we can join your discussion room):  **[Room X] I need help with <...>?**  [alternatively, you can request for help in Zoom under **Breakout Rooms > Ask for Help**] |

**Lab 3**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lab Pair** | **Hostnames** | **Lab Pair** | **Hostnames** |
| Guangyuan  Zimu | **xgpc5** | Chia Ler  Samuel | **xgpf5** |
| Zheng Wen  Kang Ming | **xgpc6** | Kevin  Yong Zhi | **xgpf6** |
| Aiden  Benedict | **xgpc7** | Jin Wen | **xgpf7** |
| Phi Long  Bertrand | **xgpc8** |  | **xgpf8** |
| Teng Le  Guozhen | **xgpc9** |  | **xgpf9** |

|  |
| --- |
| **Announcements (from Prof/TA):**  [3:15PM]  If you are getting “Connected closed”, please proceed to <https://mysoc.nus.edu.sg> (login with your SoC account), under General eServices (dropdown menu) > My SoC account > Login again (LOL) > My SoC resources (left menu) > Enable “SoC compute cluster”  [3:15PM]  For Exercise 10, you should observe that the first print statement appears to always print 32, even though this should not be the case. Try adding a \_\_nanosleep(<time>) statement to delay the thread that prints, and observe what happens. |
| If you have questions: post your question below as follows (so we can join your discussion room): **[Room X] I need help with <...>?**  [alternatively, you can request for help in Zoom under **Breakout Rooms > Ask for Help**] |
|  |
|  |

**Lab 2**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lab Pair** | **Hostnames** | **Lab Pair** | **Hostnames** |
| Guangyuan  Zimu | **soctf-pdc-001**  **soctf-pdc-009** | Teng Le  Guozhen | **soctf-pdc-006**  **soctf-pdc-014** |
| Zheng Wen  Kang Ming | **soctf-pdc-002**  **soctf-pdc-010** | Chia Ler  Samuel | **soctf-pdc-007**  **soctf-pdc-015** |
| Aiden  Benedict | **soctf-pdc-003**  **soctf-pdc-011** | Kevin  Yong Zhi | **soctf-pdc-008**  **soctf-pdc-016** |
| Jin Wen  Zeng Hao | **soctf-pdc-004**  **soctf-pdc-012** | Do not use  >:( | **soctf-pdc-018**  **soctf-pdc-020** |
| Phi Long  Bertrand | **soctf-pdc-005**  **soctf-pdc-013** | Do not use  >:( | **soctf-pdc-019**  **soctf-pdc-021** |

|  |
| --- |
| **Announcements (from Prof/TA):**  [2:15PM]  Please profile the program only on the lab machines and not on your personal laptops/PC.  [2:15PM]  For ex5, modify B to layout its elements in column-major order (if you want to experiment, you can try A, B both in row-major only or column-major only layouts), i.e. your matrix B in your program should be the **transpose of B** instead when represented in column-major order. (You will need to modify the indexes in the multiplication in the thrice-nested for loop.)    [2:15PM]  For ex4, you only need to compute the IPC and MFLOPS for mm-omp.c |
| If you have questions: post your question below as follows (so we can join your discussion room): **[Room X] I need help with <...>?**  [alternatively, you can request for help in Zoom under **Breakout Rooms > Ask for Help**] |
|  |
|  |
|  |
|  |
|  |

**Lab 1**

|  |
| --- |
| **Announcements (from Prof/TA):**  [2:30PM]  What you should do in your breakout rooms - get to know your lab partner, and you can work on the lab exercises together (one of you can share your screen!)  [2:35PM]  To Sunfire: **ssh <your SOC ID>@sunfire.comp.nus.edu**  To lab machines: **ssh <username>@<node name>**  [2:40PM]  If you need the powerpoint slides to refer to the instructions or errata, they’re available in the Telegram group or at: <https://drive.google.com/file/d/1UASiz0d6DkuHwKyM7hr792lnYd88RRb4/view?usp=sharing>  [2:55PM]  If you need to unzip the L1\_code.zip, just run “unzip L1\_code.zip” after transferring your file onto the lab machines (via Sunfire) with scp.  [3:55PM]  Hi everyone, the lab session is almost up - you can leave if you're done with the exercises (I'll leave the rooms up); I have another class at 4pm so I can't stay to help - but you can post questions in the shared doc and I will answer them later |
| If you have questions: post your question below as follows (so we can join your discussion room): **[Room X] I need help with <...>?**  [alternatively, you can request for help in Zoom under **Breakout Rooms > Ask for Help**] |
| [Breakout Room 2] I need help with ... Ex7  1) Output (what should we print)  2i) Is it okay to skip producing if the buffer is full or need to use signals (use signals to check if buffer is empty or full)  2ii) use a global variable to keep track of the size of buffer, if buffer is full, exit producer thread  Ex8  1) How to create shared memory for buffer?  A: For ex7, if you are letting your producers/consumers run indefinitely until manually stopped, you should print out intermediate steps when the consumer consumes an item (to show your program is making progress and not in a deadlock). If you are letting them run for a fixed number of items N, you can let them print at the end after all N items are produced and consumed.  It is OK (in fact necessary otherwise the buffer will be overfull) to skip producing if the buffer is full - it is more ideal however for the producer to block when the buffer is full, to prevent it from repeatedly entering the critical section (and doing no useful work) until the buffer has empty slots again.  For ex8, you need to look up the creation and attaching of shared memory regions - see <https://www.man7.org/linux/man-pages/man2/shmat.2.html> |
| For ex 2:  main thead: creating thread 0  main thead: creating thread 1  main thead: creating thread 2  main thead: creating thread 3  thread #1 incrementing counter. counter = 2  thread #2 incrementing counter. counter = 3  main thead: creating thread 4  thread #0 incrementing counter. counter = 1  thread #3 incrementing counter. counter = 4  thread #4 incrementing counter. counter = 5  main thead: creating thread 5  main thead: creating thread 6  thread #5 incrementing counter. counter = 6  main thead: creating thread 7  thread #7 incrementing counter. counter = 7  Since counter is a shared variable, why is it that we don’t run into a synchronization or race condition issue with the counter value?  A: The program may occasionally print the correct value of the counter even though it is not synchronised - this depends on the scheduling/order in which operations of the threads are reflected in the shared counter. Run the program multiple times - does the final result for the counter change?  Change the number of threads to a larger number. It might take a large number of runs before you see an incorrect final result, since the number of instructions to update the shared counter is pretty small  **Yeah it always sums up correctly.???? I set it to a 100** |
| For ex 4, I seem to still be unable to make pthread\_join wait for all child to complete? I’m doing this:  for (t2 = 0; t1 < SUB\_THREADS; t2++) {  int t = t2;  pthread\_join(sub\_threadid[t], NULL);  }  But for sub and add but it doesn’t seem to wait. What am i doing wrong?  If you want to wait for all children you need to join on the ADD threads as well - the above code section will only wait for all the SUB threads **yes i did the same loop for add threads. Oh! I realized my error**  **Oh that’s good - what was the issue**  **Was using tid and not the thread itself** |
| **How does shared memory (shmget, shmat, etc.) work?**  Use shmget to request the OS to allocate a memory region to be shared - this will return an integer  Oh do you mean like how the OS handles shared memory? And not how to use it? Ah - okay I’ll get back to this after class (sorry!)  Yeah, I want to understand what’s going on  No worries and thanks  Oh, some specific questions are:   * Why does shared memory need to attach and detach? What does that even mean? Since shmget already gets the OS to allocate space and return an ID (a form of handle on the shared memory, I’m guessing), what do attach and detach actually, um, do? * (this one is more of semaphores) In a similar vein, What’s the difference between sem\_unlink and sem\_close? The man pages say “remove a named semaphore” and “close a named semaphore” which isn’t too helpful. And it seems asymmetric too because there’s only one “sem\_open” so if sem\_open pairs with sem\_close then what does sem\_unlink pair with?   [Answered in private] |
|  |