WEEK 0 ASSIGNMENT

1. What is the name of the supercomputer developed by C-DAC

A. PARAM
B. CRAY
C. Fugaku
D. Frontier

Answer: PARAM

2. The most common method used for improving the performance of supercomputers is:

A. Parallelism
B. Making changes to sequential code
C. Using multiple storage drives
D. Upgrading the device drivers

Answer: Parallelism

3. Which of the following applications need to make use of supercomputers. Choose the best answer:

A. Weather Forecasting
B. Astrophysics
C. Animation
D. Designing of web pages

Answer: Weather Forecasting, Astrophysics, Animation

4. Which are the major components of a computer

A. CPU, Memory, I/O
B. CPU, Memory, Magnetic Tape
C. Magnetic Tape, Memory, Floppy Disk
D. I/O, Printer, Mouse

Answer: CPU, Memory, I/O

5. Which of the following components interacts with the user.

A. CPU, Basic I/O
B. Operating System (O/S), Basic I/O
C. CPU, Device Driver
D. Operating System (O/S), CPU

Answer: Operating System (O/S), Basic I/O

6. What is the correct order followed to generate executable code from source code.

A. Preprocessor → Compiler → Linker → Loader
B. Compiler → Preprocessor → Linker → Loader
C. Preprocessor → Linker → Compiler → Loader
D. Compiler → Linker → Preprocessor → Loader

Answer: Preprocessor → Compiler → Linker → Loader

7. Which type of architecture is commonly used in supercomputers to achieve higher performance.

A. Single-core architecture
B. Distributed-memory architecture
C. Integrated-memory architecture
D. Microcontroller-based architecture

Answer: Distributed-memory architecture

WEEK 1 ASSIGNMENT

1. What is an Operating System (OS).

A. Hardware component for managing software
B. Program that manages hardware and software resources on a computer
C. Application software for word processing
D. Device for improving network speed

Answer: Program that manages hardware and software resources on a computer

2. Why is an operating system necessary.

A. It provides hardware for computers
B. It simplifies user interaction, manages resources, and ensures smooth and secure operation
C. It enables only one user to use the computer
D. It prevents computers from connecting to the internet

Answer: It simplifies user interaction, manages resources, and ensures smooth and secure operation

3. Linux was initially developed as a result of limitations found in which operating system.

A. Windows
B. MINIX
C. MS-DOS
D. MacOS

Answer: MINIX

4. What is the function of the Memory Management Unit (MMU) in Linux.

A. Scheduling processes to the CPU
B. Translating virtual addresses to physical addresses
C. Handling network communications
D. Interpreting user commands

Answer: Translating virtual addresses to physical addresses

5. Which file type in Linux points to another file or directory.

A. Regular file
B. Directory
C. Symbolic Link
D. Device file

Answer: Symbolic Link

6. Which command is used to check the currently logged-in user.

A. uname
B. whoami
C. ps
D. top

Answer: whoami

7. Which command provides information about the CPU's frequency on a Linux system.

A. free
B. lscpu
C. uname -a
D. ps -e

Answer: lscpu

8. How can you display a list of all running processes in Linux.

A. top
B. ps
C. ps -e
D. kill

Answer: ps -e

9. Which command is used to search for a pattern in a file.

A. awk
B. sed
C. grep
D. uniq

Answer: grep

10. Which command displays all currently running processes in the system with detailed information.

A. ps
B. ps -x
C. ps -ef
D. kill

Answer: ps -ef

11. Which Linux command displays disk space usage.

A. df
B. du
C. vmstat
D. free

Answer: df

12. Which shell is the default on most Linux distributions and is an enhanced version of the Bourne Shell.

A. bash
B. csh
C. zsh
D. ksh

Answer: bash

13. Which Linux file permission structure indicates read, write, and execute permissions for the owner only.

A. rwxr-xr-x
B. rw-r--r--
C. rwx------
D. rw-rw-rw-

Answer: rwx------

14. What is the primary function of the Linux kernel.

A. Managing user interfaces
B. Bridging hardware and software applications
C. Managing internet connectivity
D. Organizing file permissions

Answer: Bridging hardware and software applications

15. What is the role of symbolic links in Linux.

A. To execute commands in a terminal
B. To point to another file or directory
C. To manage memory allocation
D. To handle process synchronization

Answer: To point to another file or directory

WEEK 2 ASSIGNMENT

1. **How do you declare and use a variable in a shell script.**

   A. var=5; echo $var
   B. int var = 5; print(var)
   C. set var 5; display var
   D. variable: 5; output $variable

Answer: var=5; echo $var

2. **How do you check if a file exists using an if condition in a shell script.**

   A. [-d file1]
   B. [-f file1]
   C. [-e file1]
   D. [-x file1]

Answer: [-f file1]

3. **What is the primary function of the Process Scheduler in the Linux kernel.**

   A. To allocate memory to processes
   B. To manage CPU time allocation for processes
   C. To monitor network packets
   D. To synchronize file operations

Answer: To manage CPU time allocation for processes

4. **Which of the following is NOT a key element of the Network Sub-System in Linux.**

   A. Socket API
   B. Inode Table
   C. Network Protocol Layers
   D. Packet Processing

Answer: Inode Table

5. **What is the default signal sent by the kill command in Linux.**

   A. SIGKILL (9)
   B. SIGTERM (15)
   C. SIGSTOP (19)
   D. SIGCONT (18)

Answer: SIGTERM (15)

6. **The performance of supercomputers is typically measured in.**

   A. FLOPS (Floating-Point Operations Per Second)
   B. GHz (Gigahertz)
   C. IOps (Input/Output Operations Per Second)
   D. GBps (Gigabytes per Second)

Answer: FLOPS (Floating-Point Operations Per Second)

7. **What benchmark is used to measure the performance of a supercomputer**

   A. SPECint
   B. Linpack
   C. Geekbench
   D. Cinebench

Answer: Linpack

8. **What is the primary goal of parallel computing.**

   A. To execute tasks sequentially
   B. To increase the complexity of the algorithm
   C. To minimize hardware usage
   D. To reduce computation time by executing tasks simultaneously

Answer: To reduce computation time by executing tasks simultaneously

9. **Which of the following attributes of a parallel algorithm directly impacts its ability to efficiently utilize additional processors as the system size increases.**

   A. Scalability
   B. Concurrency
   C. Data locality
   D. Modularity

Answer: Scalability

10. **Which of the following categories in Flynn's Taxonomy represents a system where one instruction operates on a single data stream.**

    A. MISD
    B. SIMD
    C. MIMD
    D. SISD

Answer: SISD

11. Which memory type in HPC systems acts as a buffer between the CPU and RAM to speed up processing.

A. Flash memory
B. Registers
C. Hard disk
D. Cache memory

Answer: Cache memory

12. Which hardware component is responsible for copying information from main memory to cache memory automatically.

A. DMA controller
B. CPU
C. Memory management unit (MMU)
D. Cache controller

Answer: Cache controller

13. Which of the following best highlights the primary advantage of NUMA over UMA in large-scale multiprocessor systems, especially concerning memory access efficiency and system scalability.

A. NUMA minimizes memory access contention by ensuring each processor has faster access to its local memory and reduces bandwidth bottlenecks across processors
B. NUMA architecture is simpler to implement because it requires fewer memory management techniques and results in less software overhead
C. UMA systems benefit from uniform memory access latency, ensuring predictable performance regardless of the number of processors, which is particularly beneficial for real-time systems
D. UMA architectures scale better with increasing processor count by maintaining uniform memory access, resulting in less complexity in managing memory locality

Answer: NUMA minimizes memory access contention by ensuring each processor has faster access to its local memory and reduces bandwidth bottlenecks across processors

14. Which of the following situations can still lead to race conditions, even when a mutex is used.

A. A thread releases the mutex before completing the critical section
B. The critical section is too small to cause race conditions
C. Multiple mutex locks are used in a consistent order across threads
D. Threads use a recursive mutex that supports multiple locks by the same thread

Answer: A thread releases the mutex before completing the critical section

15. In parallel programming, what happens if a thread fails to reach a barrier in a multi-threaded program

A. The program will terminate all other threads immediately
B. Threads that have reached the barrier will remain blocked indefinitely
C. The barrier will adjust automatically to the number of threads that reach it
D. All threads will proceed regardless of synchronization

Answer: Threads that have reached the barrier will remain blocked indefinitely

WEEK 3 ASSIGNMENT

1. What is a multi-core processor.

A. A single-core processor
B. A processor with multiple execution units
C. A processor that supports multi-threading
D. A processor with more than one core integrated into a single chip

Answer: A processor with more than one core integrated into a single chip

2. What is the primary purpose of using multiple cores in a processor.

A. To increase the processor's clock speed
B. To reduce power consumption
C. To improve parallel processing capabilities
D. To enhance single-threaded performance

Answer: To improve parallel processing capabilities

3. What is a "thread" in computing.

A. A physical component of a processor
B. A single sequence of programmed instructions
C. A cache memory unit
D. A type of input/output operation

Answer: A single sequence of programmed instructions

4. What is the main purpose of cache memory in multi-core architectures.

A. To store data permanently
B. To reduce memory access time
C. To increase the number of cores
D. To manage power consumption

Answer: To reduce memory access time

5. Which of the following is a technique to improve memory access speed in multi-core processors.

A. Increasing clock speed
B. Prefetching
C. Reducing the number of cores
D. Increasing pipeline depth

Answer: Prefetching

6. What is 'NUMA' in the context of multi-core architectures.

A. Non-Uniform Memory Access
B. Network Unified Memory Architecture
C. New Unified Memory Allocation
D. None of these

Answer: Non-Uniform Memory Access

7. What is 'cache thrashing' in multi-core architectures.

A. Frequent invalidation of cache lines causing performance degradation
B. Increasing the size of the cache
C. Reducing memory latency
D. Allocating more memory to a single core

Answer: Frequent invalidation of cache lines causing performance degradation

8. How does hyper-threading improve processor performance.

A. By increasing the clock speed
B. By allowing more efficient use of CPU resources
C. By adding more memory
D. By reducing power consumption

Answer: By allowing more efficient use of CPU resources

9. What is a significant challenge of hyper-threading in terms of memory access.

A. Reduced memory bandwidth
B. Increased memory latency
C. Cache contention between threads
D. Decreased cache size

Answer: Cache contention between threads

10. In hyper-threading, what is 'resource sharing' among threads.

A. Each thread gets a dedicated set of resources
B. Threads share CPU and memory resources
C. Threads do not share any resources
D. Each thread gets more resources than usual

Answer: Threads share CPU and memory resources

11. What is 'cache coherence' in the context of multi-core processors.

A. Caches storing different versions of data
B. Ensuring all caches reflect the most recent data value
C. Increasing cache size
D. Reducing cache access time

Answer: Ensuring all caches reflect the most recent data value

12. What is 'memory latency'.

A. The time it takes for data to move from one core to another
B. The delay between a memory request and the start of data transfer
C. The total capacity of the memory
D. The size of the memory blocks

Answer: The delay between a memory request and the start of data transfer

13. What is the main purpose of virtual memory.

A. To increase the size of the physical memory
B. To provide an abstraction of the physical memory
C. To enhance the CPU speed
D. To manage input/output operations

Answer: To provide an abstraction of the physical memory

14. What is 'paging' in the context of virtual memory.

A. Dividing the CPU time among processes
B. Splitting memory into fixed-size blocks
C. Transferring data from cache to register
D. Organizing disk storage

Answer: Splitting memory into fixed-size blocks

15. In a paging system, what is a 'page fault'.

A. An error in the page table
B. An attempt to access a non-resident page in memory
C. A corrupted disk sector
D. A CPU scheduling error

Answer: An attempt to access a non-resident page in memory

WEEK 4 ASSIGNMENT

1. Which of the following best describes 'speedup' in parallel computing.

A. The ratio of sequential execution time to parallel execution time
B. The increase in power consumption
C. The total number of processors used
D. The reduction in clock speed

Answer: The ratio of sequential execution time to parallel execution time

2. What is 'Amdahl's Law'.

A. A principle for optimizing cache usage
B. A formula to predict the theoretical maximum speedup in parallel computing
C. A method to increase memory bandwidth
D. A technique for improving input/output operations

Answer: A formula to predict the theoretical maximum speedup in parallel computing

3. Which of the following is an example of embarrassingly parallel tasks.

A. Tasks that require frequent synchronization
B. Tasks that can be easily separated into independent parts
C. Tasks that depend on the results of other tasks
D. Tasks that involve continuous communication

Answer: Tasks that can be easily separated into independent parts

4. Which of the following is a challenge in parallel computing.

A. Clock speed management
B. Heat dissipation
C. Load balancing
D. Static memory allocation

Answer: Load balancing

5. What is a 'race condition' in the context of parallel computing.

A. A situation where two or more processes compete for CPU time
B. A condition where the behavior of a system depends on the timing of events
C. A method for optimizing memory usage
D. A technique for increasing processing speed

Answer: A condition where the behavior of a system depends on the timing of events

6. OpenMP is used for.

A. Shared Memory Programming
B. Distributed Memory Programming
C. Both Distributed and Shared Memory Programming
D. None of these

Answer: Shared Memory Programming

7. Which directive is used to parallelize a loop in OpenMP.

A. #pragma omp parallel
B. #pragma omp loop
C. #pragma omp parallel loop
D. #pragma omp parallel for

Answer: #pragma omp parallel for

8. In the OpenMP execution model, what does the term "fork" refer to.

A. The point where the program starts
B. The creation of a single thread
C. The destruction of a team of threads
D. The creation of multiple threads by a single thread

Answer: The creation of multiple threads by a single thread

9. Which of the following best describes the "join" phase in the OpenMP execution model.

A. Threads are created and start executing
B. Threads finish their execution and rejoin the master thread
C. Threads are paused and waiting for synchronization
D. Threads communicate with each other

Answer: Threads finish their execution and rejoin the master thread

10. Which function in OpenMP is used to get the unique ID of a thread.

A. omp_get_num_threads()
B. omp_get_thread_num()
C. omp_num_threads()
D. omp_thread_num()

Answer: omp_get_thread_num()

11. Which function is used to set the number of threads in OpenMP.

A. omp_get_num_threads()
B. omp_get_thread_num()
C. omp_num_threads()
D. omp_set_num_threads()

Answer: omp_set_num_threads()

12. Which function is used to get the total number of threads in OpenMP.

A. omp_get_num_threads()
B. omp_get_thread_num()
C. omp_num_threads()
D. omp_set_num_threads()

Answer: omp_get_num_threads()

13. What is the role of #pragma omp parallel sections in OpenMP.

A. To create a critical section
B. To parallelize individual sections of code
C. To terminate a parallel region
D. To synchronize all threads

Answer: To parallelize individual sections of code

14. Which environment variable is used to set the number of threads in an OpenMP program.

A. OMP_NUM_THREADS
B. OMP_SET_THREADS
C. OMP_THREAD_COUNT
D. OMP_THREADS

Answer: OMP_NUM_THREADS

15. How does OpenMP handle parallel regions in its execution model.

A. By creating multiple parallel regions simultaneously
B. By creating a single parallel region and executing it repeatedly
C. By creating parallel regions dynamically as needed
D. By creating parallel regions only at the beginning of the program

Answer: By creating parallel regions dynamically as needed

WEEK 5 ASSIGNMENT

1. What is the purpose of the single directive in OpenMP.

A. To specify a block of code that should be executed by multiple threads
B. To ensure a block of code is executed by all threads
C. To ensure a block of code is executed by only one thread
D. To synchronize threads at the end of a block of code

Answer: To ensure a block of code is executed by only one thread

2. What is the main function of the master directive in OpenMP.

A. To ensure a block of code is executed by multiple threads
B. To synchronize all threads at a specific point
C. To ensure a block of code is executed only by the master thread
D. To distribute tasks among all threads

Answer: To ensure a block of code is executed only by the master thread

3. What is the purpose of the private clause in OpenMP.

A. To ensure a variable is shared among all threads
B. To create a common (single) copy of a variable for all threads
C. To create a separate copy of a variable for each thread
D. To prevent a variable from being used by any thread

Answer: To create a separate copy of a variable for each thread

4. What happens to the value of a variable specified as private at the end of a parallel region in OpenMP.

A. It retains its last value from the parallel region
B. It is set to zero
C. It is discarded, and the original value is restored
D. It is shared among all threads

Answer: It is discarded, and the original value is restored

5. Can a variable be both private and shared in the same OpenMP parallel region.

A. Yes, it can be both
B. No, a variable cannot be both private and shared
C. Yes, but only when the master clause is used
D. Yes, but only if it is declared with the firstprivate clause

Answer: No, a variable cannot be both private and shared

6. Which clause is used to share a variable among all threads in OpenMP.

A. private
B. shared
C. firstprivate
D. lastprivate

Answer: shared

7. In OpenMP, what is a thread's default data-sharing attribute for variables declared inside a parallel region.

A. private
B. shared
C. firstprivate
D. lastprivate

Answer: shared

8. Which OpenMP clause allows a variable to be initialized and private to each thread.

A. private
B. firstprivate
C. shared
D. reduction

Answer: firstprivate

9. What is the difference between private and firstprivate clauses in OpenMP.

A. private clause initializes variables, firstprivate does not
B. private clause does not initialize variables, firstprivate initializes with original values
C. private clause shares variables, firstprivate does not
D. private clause uses global variables, firstprivate uses local variables

Answer: private clause does not initialize variables, firstprivate initializes with original values

10. In which scenario is the single directive particularly useful in OpenMP.

A. When all threads must execute a piece of code
B. When a block of code needs to be executed exactly once
C. When each thread must execute the code block multiple times
D. When threads need to be synchronized

Answer: When a block of code needs to be executed exactly once

11. What is the main difference between the single and master directives in OpenMP.

A. single allows any one thread to execute the code, while master allows only the master thread to execute the code
B. single synchronizes all threads, while master does not
C. single is used only in nested regions, while master is not
D. single can be combined with nowait, while master cannot

Answer: single allows any one thread to execute the code, while master allows only the master thread to execute the code

12. Which of the following is true about the execution of a single directive block in OpenMP.

A. All threads must reach the single block simultaneously
B. Only one thread will execute the single block, regardless of the number of threads
C. Multiple threads can execute the single block concurrently
D. The single block is ignored if more than one thread reaches it

Answer: Only one thread will execute the single block, regardless of the number of threads

13. What is the primary purpose of the threadprivate directive in OpenMP.

A. To create a shared variable among all threads
B. To create a variable private to each thread across parallel regions
C. To synchronize threads at a specific point
D. To reduce the number of threads

Answer: To create a variable private to each thread across parallel regions

14. In which scenario is the threadprivate directive particularly useful in OpenMP.

A. When a variable needs to be shared among all threads
B. When a variable's value needs to persist across multiple parallel regions
C. When a variable needs to be synchronized
D. When a variable should be used only in the main thread

Answer: When a variable's value needs to persist across multiple parallel regions

15. What is the main difference between private and threadprivate clauses in OpenMP.

A. private variables are local to each thread within a single parallel region, while threadprivate variables persist across parallel regions
B. private variables are shared among all threads, while threadprivate variables are not
C. private variables are global, while threadprivate variables are local
D. private variables are initialized to zero, while threadprivate variables are not

Answer: private variables are local to each thread within a single parallel region, while threadprivate variables persist across parallel regions

WEEK 6 ASSIGNMENT

1. **What is the primary programming model for MPI?**

   A. Shared Memory
   B. Distributed Memory
   C. Hybrid Memory
   D. Virtual Memory

Answer: Distributed Memory

2. **Which MPI function is used to initialize the MPI environment?**

   A. MPI_Start
   B. MPI_Init
   C. MPI_Begin
   D. MPI_Boot

Answer: MPI_Init

3. **What is a communicator in MPI?**

   A. A function that sends data
   B. A data type for storing messages
   C. A group of processes that can communicate with each other
   D. A hardware device

Answer: A group of processes that can communicate with each other

4. **Which function would you use to determine the number of processes in a communicator?**

   A. MPI_Size
   B. MPI_Num
   C. MPI_Comm_size
   D. MPI_Process_count

Answer: MPI_Comm_size

5. **What does MPI_Finalize do?**

   A. Starts the MPI environment
   B. Ends communication between processes
   C. Finalizes the MPI environment
   D. Allocates memory for MPI operations

Answer: Finalizes the MPI environment

6. **What is point-to-point communication in MPI?**

A. Communication between two specific processes
B. Communication between all processes
C. Communication within a single process
D. Communication between a process and a file

Answer: Communication between two specific processes

7. **What does the tag parameter in MPI_Send and MPI_Recv functions represent?**

A. The size of the message
B. The type of data being sent
C. A user-defined identifier to distinguish messages
D. The rank of the sending process

Answer: A user-defined identifier to distinguish messages

8. **What does the count parameter in MPI_Send and MPI_Recv specify?**

A. The size of the buffer
B. The number of elements in the message
C. The rank of the receiving process
D. The tag of the message

Answer: The number of elements in the message

9. **What happens if MPI_Recv is called with MPI_ANY_SOURCE for the source parameter?**

A. It receives a message from any available process
B. It causes an error because the source must be specified
C. It waits for a message from the rank 0 process
D. It receives messages only from processes with an odd rank

Answer: It receives a message from any available process

10. **What is collective communication in MPI?**

A. Communication involving one sender and one receiver
B. Communication involving all processes in a communicator
C. Communication between two specific processes
D. Communication without synchronization

Answer: Communication involving all processes in a communicator

11. **What does the MPI_Bcast function do?**

A. Sends a message from one process to all other processes in a communicator
B. Gathers data from all processes in a communicator
C. Reduces data from all processes in a communicator
D. Synchronizes all processes in a communicator

Answer: Sends a message from one process to all other processes in a communicator

12. **Which function in MPI allows processes to synchronize at a certain point?**

A. MPI_Barrier
B. MPI_Sync
C. MPI_Wait
D. MPI_Pause

Answer: MPI_Barrier

13. **Which MPI function is used to divide data among all processes in a communicator?**

A. MPI_Scatter
B. MPI_Gather
C. MPI_Reduce
D. MPI_Alltoall

Answer: MPI_Scatter

14. **In MPI_Reduce, what is the role of the op parameter?**

A. Specifies the datatype of the elements
B. Specifies the root process
C. Specifies the reduction operation to be performed
D. Specifies the communicator

Answer: Specifies the reduction operation to be performed

15. **If each process sends an array of 4 integers to the root process using MPI_Gather, and there are 4 processes, how many integers will the root process receive in total?**

A. 4
B. 8
C. 20
D. 16

Answer: 16

WEEK 7 ASSIGNMENT

1. Which of the following is the most common interconnect used for GPUs in a server board?
   A. NVLink
   B. PCIe
   C. USB-C
   D. SATA
   Answer: PCIe

2. What is the primary purpose of NVLink in GPU server boards?
   A. Power management
   B. High-speed GPU-to-GPU communication
   C. Memory allocation
   D. Cooling optimization
   Answer: High-speed GPU-to-GPU communication

3. What is the primary advantage of a GPU over a CPU in parallel processing tasks?
   A. Higher clock speed
   B. Better cache memory
   C. Ability to perform thousands of tasks simultaneously
   D. Lower power consumption
   Answer: Ability to perform thousands of tasks simultaneously

4. Which of the following is included in the GPU software stack to allow interaction between the CPU and GPU?
   A. OpenCL Runtime
   B. Vulkan Driver
   C. CUDA Driver
   D. DirectX
   Answer: CUDA Driver

5. Which type of memory is commonly associated with GPU (latest) server boards for high-performance tasks?
   A. DDR4
   B. HBM
   C. GDDR
   D. LPDDR
   Answer: GDDR

6. What does GFLOPS stand for in the context of NVIDIA GPUs?
   A. Giga-floating point operations per second
   B. General floating operations
   C. Graphics floating performance system
   D. GPU floating output processing speed
   Answer: Giga-floating point operations per second

7. What does GPC stand for in Nvidia GPUs?
   A. Graphics Processing Cluster
   B. General Processing Core
   C. GPU Performance Control
   D. Graphics Pipeline Controller
   Answer: Graphics Processing Cluster

8. What does TPC stand for in Nvidia GPUs?
   A. Texture Processing Cluster
   B. Thread Processing Core
   C. Tensor Processing Compute
   D. Task Performance Controller
   Answer: Texture Processing Cluster

9. In the NVIDIA GPU vector pipeline, what does the term "SIMT" stand for?
   A. Single Instruction Multiple Threads
   B. Synchronized Independent Multi-Threading
   C. Sequential Instruction Multi-Tasking
   D. Shared Independent Multi-Threading
   Answer: Single Instruction Multiple Threads

10. Which of the following formulas is used to calculate the theoretical peak performance (FLOPS) of a GPU?
    A. Maximum FLOPS = (SM Count × CUDA Cores per SM × 2 FLOPS per CUDA core × GPU Frequency)
    B. Maximum FLOPS = (SM Count + CUDA Cores per SM) × GPU Frequency
    C. Maximum FLOPS = (CUDA Cores per SM × GPU Frequency) ÷ 2
    D. Maximum FLOPS = (SM Count × GPU Frequency) × 2 FLOPS per CUDA core
    Answer: Maximum FLOPS = SM Count × CUDA Cores per SM × 2 FLOPS per CUDA core × GPU Frequency

11. In CUDA, how is the execution of threads organized?
    A. Single threads running sequentially
    B. Blocks of threads within grids
    C. One large grid with independent threads
    D. Multiple streams with independent execution
    Answer: Blocks of threads within grids

12. In the CUDA programming model, what is the maximum dimensionality of a block and grid?
    A. Block: 1D, Grid: 1D
    B. Block: 2D, Grid: 2D
    C. Block: 3D, Grid: 3D
    D. Block: 4D, Grid: 4D
    Answer: Block: 3D, Grid: 3D

13. Which of the following defines the number of blocks in a grid in CUDA?
    A. blockDim
    B. gridDim
    C. threadIdx
    D. blockIdx
    Answer: gridDim

14. Which of the following is used to access the thread's unique index within its block in CUDA?
    A. gridDim
    B. blockDim
    C. threadIdx
    D. blockIdx
    Answer: threadIdx

15. How is the Global Thread ID (TID) typically calculated in a 1D grid in CUDA?
    A. GlobalTID = threadIdx.x * blockIdx.x
    B. GlobalTID = threadIdx.x + blockIdx.x
    C. GlobalTID = threadIdx.x * blockDim.x + blockIdx.x
    D. GlobalTID = blockIdx.x * threadIdx.x + blockDim.x
    Answer: GlobalTID = threadIdx.x * blockDim.x + blockIdx.x

WEEK 8 ASSIGNMENT

1. What is the primary scheduling policy used by the warp scheduler in NVIDIA GPUs?
   A. Round-Robin
   B. Greedy-then-oldest
   C. First-Come, First-Served
   D. Least Recently Used
   Answer: Greedy-then-oldest

2. In NVIDIA GPUs, how many threads are typically there in a warp?
   A. 16
   B. 32
   C. 64
   D. 128
   Answer: 32

3. Which of the following affects warp scheduling efficiency on NVIDIA GPUs?
   A. Memory access patterns
   B. Instruction dependencies
   C. Control flow divergence
   D. All of the above
   Answer: All of the above

4. What happens if a warp encounters a memory access latency during execution?
   A. The warp is discarded
   B. It stalls until the data is fetched
   C. It switches to another warp
   D. The execution fails
   Answer: It stalls until the data is fetched

5. In Nvidia GPUs, what does the Instruction Dispatcher do?
   A. Fetches data from global memory
   B. Dispatches instructions from the instruction cache to the execution units
   C. Handles thread synchronization
   D. Allocates shared memory dynamically
   Answer: Dispatches instructions from the instruction cache to the execution units

6. Which of the following functions is used to determine the number of devices in the system that CUDA can use?

A. cudaGetDeviceProperties()
B. cudaGetDevice()
C. cudaGetDeviceCount()
D. cudaSetDevice()
Answer: cudaGetDeviceCount()

7. Which of the following CUDA functions provides information about the total amount of memory available on a specific device?
   A. cudaMalloc()
   B. cudaMemGetInfo()
   C. cudaDeviceSynchronize()
   D. cudaMemcpy()
   Answer: cudaMemGetInfo()

8. The kernel code is identified by the _____ qualifier with void return type.
   A. **device**
   B. **global**
   C. **host**
   D. **shared**
   Answer: global

9. Calling a kernel is typically referred to as _____.
   A. Kernel Invocation
   B. Kernel Execution
   C. Kernel Launch
   D. Kernel Compilation
   Answer: Kernel Invocation

10. _____ is callable from the host only.
    A. **global**
    B. **device**
    C. **host**
    D. **constant**
    Answer: host

11. What is the term for when threads in a warp take different execution paths?
    A. Warp fragmentation
    B. Warp serialization
    C. Warp divergence

D. Warp branching
Answer: Warp divergence

12. What could be a performance bottleneck caused by the Instruction Dispatcher in a GPU?
    A. Memory bandwidth limitations
    B. Instruction fetch latency
    C. Cache miss rate
    D. Thread divergence
    Answer: Instruction fetch latency

13. Which programming language predominantly uses row-major order for multidimensional arrays?
    A. Fortran
    B. Python
    C. C/C++
    D. MATLAB
    Answer: C/C++

14. If a 2D array `arr[3][3]` is stored in row-major order, what is the memory layout?
    A. arr[0][0], arr[0][1], arr[0][2], arr[1][0], arr[1][1], arr[1][2], arr[2][0], arr[2][1], arr[2][2]
    B. arr[0][0], arr[1][0], arr[2][0], arr[0][1], arr[1][1], arr[2][1], arr[0][2], arr[1][2], arr[2][2]
    C. arr[0][0], arr[1][1], arr[2][2], arr[0][1], arr[1][2], arr[2][0], arr[0][2], arr[1][0], arr[2][1]
    D. arr[2][2], arr[2][1], arr[2][0], arr[1][2], arr[1][1], arr[1][0], arr[0][2], arr[0][1], arr[0][0]
    Answer: arr[0][0], arr[0][1], arr[0][2], arr[1][0], arr[1][1], arr[1][2], arr[2][0], arr[2][1], arr[2][2]

15. What is the main advantage of using Numba's `@cuda.jit` for GPU programming over traditional CUDA programming in C/C++?
    A. Numba provides higher performance than CUDA C++
    B. Numba allows Python programmers to write GPU code without needing to learn C/C++
    C. Numba eliminates the need for kernel launches
    D. Numba automatically optimizes memory access patterns
    Answer: Numba allows Python programmers to write GPU code without needing to learn C/C++

Week 9 Assignment

1. **What is the purpose of the nvcc --version command in CUDA?**
   A. It compiles the CUDA program
   B. It checks the version of the NVIDIA GPU driver
   C. It displays the version of the CUDA compiler (nvcc)
   D. It checks the GPU hardware status

   Answer: It displays the version of the CUDA compiler (nvcc)

2. **Which of the following commands is used to compile a CUDA program using the NVIDIA compiler?**
   A. gcc
   B. nvcc
   C. cl
   D. Make

   Answer: nvcc

3. **In a SLURM script, how do you specify that your job requires access to a GPU for running a CUDA program?**
   A. SBATCH --gres=gpu:1;
   B. SBATCH --host=2;
   C. SBATCH --nvidia=1;
   D. SBATCH --devices=gpu:1

   Answer: SBATCH --gres=gpu:1

4. **Which command would you use to check if the NVIDIA driver and GPU are correctly installed and functioning on a Linux system?**
   A. nvidia-smi
   B. gpu-status
   C. check-gpu
   D. Nvidia-driver-status

   Answer: nvidia-smi

5. **Which command is used to compile CUDA code using the NVIDIA CUDA Compiler (nvcc)?**
   A. gcc -o myprogram myprogram.cu;
   B. nvcc -o myprogram myprogram.cu;
   C. cl -o myprogram myprogram.cu;
   D. make -o myprogram myprogram.cu

   Answer: nvcc -o myprogram myprogram.cu

6. **Which directive in OpenACC is used to specify parallel regions?**
    A. #pragma acc data
    B. #pragma acc kernels
    C. #pragma acc loop
    D. #pragma acc cache

    Answer: #pragma acc kernels

7. **Which of the following is not a valid OpenACC directive?**
    A. #pragma acc data
    B. #pragma acc parallel
    C. #pragma acc device
    D. #pragma acc routines

    Answer: #pragma acc device

8. **Which clause in OpenACC is used to manage data transfer between the host and device?**
    A. parallel
    B. data
    C. kernels
    D. host_data

    Answer: data

9. **What is the role of the `copyin` clause in OpenACC?**
    A. To copy data from the device to the host
    B. To copy data from the host to the device
    C. To initialize data on the device
    D. To delete data from the device

    Answer: To copy data from the host to the device

10. **What does the `copyout` clause do in OpenACC?**
    A. Copies data from the device to the host
    B. Copies data from the host to the device
    C. Synchronizes data between host and device
    D. Deletes data from the host

    Answer: Copies data from the device to the host

11. **What is the purpose of the `wait` clause in OpenACC?**
    A. To synchronize data transfer
    B. To wait for a parallel region to complete
    C. To initialize data on the device
    D. To delete data from the host

    Answer: To wait for a parallel region to complete

12. **What does the `update` directive do in OpenACC?**

A. Transfers data from host to device or device to host
B. Synchronizes device memory with host memory
C. Initializes device memory
D. Finalizes data transfer

Answer: Transfers data from host to device or device to host

13. Which clause in OpenACC is used to allocate memory on the device?

A. copy
B. malloc
C. allocate
D. create

Answer: create

14. Which directive in OpenACC is used to deallocate memory on the device?

A. #pragma acc exit data
B. #pragma acc delete
C. #pragma acc deallocate
D. #pragma acc free

Answer: #pragma acc delete

15. What does the `independent` clause do in OpenACC?

A. Ensures parallel execution of a loop
B. Indicates that iterations of a loop are independent
C. Synchronizes memory between host and device
D. Allocates memory on the device

Answer: Indicates that iterations of a loop are independent

Week 10 Assignment

1. **Which of the following is a key feature of Nsight Systems?**
    A. Real-time data encryption
    B. Detailed timeline view of CPU and GPU activities
    C. Automated code generation
    D. Cloud-based storage integration
    Answer: Detailed timeline view of CPU and GPU activities

2. **What is the recommended use case for Nsight Systems?**
    A. Optimizing gaming graphics
    B. Identifying performance bottlenecks in applications
    C. Managing server configurations
    D. Encrypting sensitive data
    Answer: Identifying performance bottlenecks in applications

3. **What type of workloads can Nsight Systems analyze?**
    A. CPU workloads only
    B. GPU workloads only
    C. Both CPU and GPU workloads
    D. Network workloads only
    Answer: Both CPU and GPU workloads

4. **What is a code profiler primarily used for?**
    A. Debugging syntax errors
    B. Analyzing the runtime performance of code
    C. Compiling source code
    D. Encrypting sensitive data
    Answer: Analyzing the runtime performance of code

5. **Which of the following is NOT a feature of code profilers?**
    A. Detecting memory leaks
    B. Debugging code directly
    C. Measuring function call frequency
    D. Identifying slow code sections
    Answer: Debugging code directly

6. **Which programming language often uses tools like gprof as a code profiler?**
    A. Python
    B. JavaScript
    C. C/C++
    D. Java
    Answer: C/C++

7. **Which type of profiling does gprof primarily provide?**
    A. Static profiling
    B. Memory profiling
    C. Call graph and flat profiling
    D. Network profiling

    Answer: Call graph and flat profiling

8. **Which compiler flag must be used to enable gprof profiling during compilation?**
    A. -Wall
    B. -pg
    C. -g
    D. -O2

    Answer: -pg

9. **What is the output file generated by a program instrumented with gprof called?**
    A. profile.log
    B. trace.out
    C. gmon.out
    D. performance.data

    Answer: gmon.out

10. **What is Valgrind primarily used for?**
    A. Debugging memory errors and profiling programs
    B. Compiling source code
    C. Encrypting data
    D. Managing system processes

    Answer: Debugging memory errors and profiling programs

11. **Which tool in Valgrind is used to detect race conditions in multi-threaded programs?**
    A. Helgrind
    B. Callgrind
    C. Memcheck
    D. Cachegrind

    Answer: Helgrind

12. **What is a major limitation of Valgrind?**
    A. It only works on Windows
    B. High performance overhead during profiling
    C. Lack of memory leak detection tools
    D. Inability to handle C/C++ programs

    Answer: High performance overhead during profiling

13. **What is the primary purpose of GDB?**
    A. Compiling programs
    B. Debugging and analyzing programs
    C. Encrypting code
    D. Optimizing performance

    Answer: Debugging and analyzing programs

14. **What does the break or b command in GDB do?**
    A. Compiles the program
    B. Sets a breakpoint in the program
    C. Resumes program execution
    D. Displays variable values

    Answer: Sets a breakpoint in the program

15. **How can you display the value of a variable in GDB?**
    A. Using the watch command
    B. Using the print or p command
    C. Using the info command
    D. Using the run command

    Answer: Using the print or p command

Assignment Week 11

1. **What is the primary goal of profiling MPI applications?**
    A. To reduce code size
    B. To identify performance bottlenecks
    C. To minimize compilation time
    D. To debug code execution
    Answer: To identify performance bottlenecks

2. **What can excessive communication overhead in MPI lead to?**
    A. Improved data processing
    B. Reduced latency
    C. Decreased scalability
    D. Faster execution time
    Answer: Decreased scalability

3. **When profiling MPI applications, what does "load imbalance" refer to?**
    A. Unequal memory usage between processes
    B. Unequal distribution of computational tasks
    C. Excessive synchronization barriers
    D. Uneven communication bandwidth
    Answer: Unequal distribution of computational tasks

4. **In MPI profiling, what is latency defined as?**
    A. Time taken to compute numerical operations
    B. Time taken for a message to travel from one process to another
    C. Time taken to allocate memory
    D. Time taken to execute parallel threads
    Answer: Time taken for a message to travel from one process to another

5. **What is the primary purpose of debugging MPI applications?**
    A. To optimize memory usage
    B. To find and fix errors
    C. To improve graphics rendering
    D. To increase bandwidth
    Answer: To find and fix errors

6. **How can memory leaks in MPI applications be detected?**
    A. Using MPI_Finalize
    B. Using memory debugging tools like Valgrind
    C. Using MPI_Comm_free
    D. Using MPI_Gather

    Answer: Using memory debugging tools like Valgrind

7. **How do numerical methods support high-performance computing?**
    A. By reducing memory hardware requirements
    B. By providing efficient algorithms for large-scale computations
    C. By reducing floating-point operations entirely
    D. By simplifying software development

    Answer: By providing efficient algorithms for large-scale computations

8. **What is the primary use of linear algebra libraries?**
    A. Data visualization
    B. Perform mathematical operations on matrices and vectors efficiently
    C. Debugging programs
    D. Web development

    Answer: Perform mathematical operations on matrices and vectors efficiently

9. **What is the primary purpose of BLAS?**
    A. Solve differential equations
    B. Provide basic operations for linear algebra
    C. Perform data compression
    D. Visualize high-dimensional datasets

    Answer: Provide basic operations for linear algebra

10. **What is the role of Thrust in linear algebra?**
    A. It offers parallel algorithms and data structures for linear algebra on GPUs
    B. It provides symbolic computation capabilities for dense matrices
    C. It automates visualization of linear transformations
    D. It is a Python library for large matrices

    Answer: It offers parallel algorithms and data structures for linear algebra on GPUs

11. **What is the primary purpose of SLURM?**
    A. Data visualization
    B. Job scheduling and resource management in HPC clusters
    C. Debugging parallel programs
    D. File system management

    Answer: Job scheduling and resource management in HPC clusters

12. Which SLURM command is used to submit a batch job?

A. squeue
B. sbatch
C. srun
D. scancel

Answer: sbatch

13. What is the role of the squeue command in SLURM?

A. Submit a job
B. Cancel a job
C. Display the status of jobs in the queue
D. Allocate resources for a job

Answer: Display the status of jobs in the queue

14. **What is the main purpose of scientific visualization?**

A. Compress large datasets
B. Represent data graphically for analysis and understanding
C. Perform statistical calculations
D. Debug algorithms

Answer: Represent data graphically for analysis and understanding

15. Which software is widely used for 3D scientific visualization?

A. Microsoft Excel
B. Blender
C. ParaView
D. Notepad++

Answer: ParaView

Assignment Week 12

1. **What is the primary purpose of MPI in parallel programming?**
    A. Shared memory parallelism
    B. Message passing in distributed systems
    C. GPU programming
    D. Directive-based programming

    Answer: Message passing in distributed systems

2. **Which directive is used to initiate parallel regions in OpenMP?**
    A. #pragma acc parallel
    B. MPI_Init
    C. #pragma omp parallel
    D. CUDA Kernel

    Answer: #pragma omp parallel

3. **CUDA is designed for programming on:**
    A. CPUs
    B. GPUs
    C. Distributed systems
    D. Real-time systems

    Answer: GPUs

4. **In OpenMP, what does the schedule clause specify?**
    A. The number of threads in a program
    B. How iterations of a loop are divided among threads
    C. Synchronization of threads
    D. Data sharing policy

    Answer: How iterations of a loop are divided among threads

5. **What does update do in OpenACC?**
    A. Transfers data between host and device
    B. Initializes the OpenACC environment
    C. Declares parallel regions
    D. Synchronizes threads

    Answer: Transfers data between host and device

6. **CUDA threads are organized into:**
    A. Blocks and grids
    B. Processes and communicators
    C. Regions and schedules
    D. Host and device memories

    Answer: Blocks and grids

7. **Which of the following is a GPU-specific memory in CUDA?**
    A. L3 Cache
    B. Shared Memory
    C. DDR4 Memory
    D. RAID Storage

    Answer: Shared Memory

8. **Which function initializes CUDA?**
    A. cudaMalloc
    B. cudaMemcpy
    C. cudaSetDevice
    D. CUDA does not require explicit initialization

    Answer: CUDA does not require explicit initialization

9. **What does the firstprivate clause do in OpenMP?**
    A. Shares variables between threads
    B. Copies the initial value of variables into each thread
    C. Allocates memory on the device
    D. Reduces variables across threads

    Answer: Copies the initial value of variables into each thread

10. **Which of the following is true about global memory in CUDA?**
    A. It is local to each thread
    B. It is shared among all threads and blocks
    C. It is faster than register memory
    D. It is not accessible by the host

    Answer: It is shared among all threads and blocks

11. **Which of the following is true about OpenMP #pragma omp parallel directive?**
    A. It initializes a single-threaded region
    B. It defines a parallel region where multiple threads execute
    C. It allows only two threads to execute simultaneously
    D. It is used only for loop parallelization

    Answer: It defines a parallel region where multiple threads execute

12. Which of the following is true about CUDA memory hierarchy?

A. Global memory is faster than shared memory
B. Shared memory is limited to a block of threads
C. Registers are shared among all threads in a block
D. Constant memory is writable by all threads

Answer: Shared memory is limited to a block of threads

13. Which of the following is true about the OpenMP critical directive?

A. It specifies a private copy of a variable for each thread
B. It creates a region of code that only one thread can execute at a time
C. It specifies the number of threads for parallel execution
D. It initializes the OpenMP environment

Answer: It creates a region of code that only one thread can execute at a time

14. Which of the following is true about CUDA thread synchronization?

A. All threads are automatically synchronized after each instruction
B. Synchronization is achieved using the __syncthreads() function
C. Threads in a block cannot be synchronized
D. Synchronization is not supported in CUDA

Answer: Synchronization is achieved using the __syncthreads() function

15. Which of the following is true about CUDA registers?

A. Registers are shared among all threads in a block
B. Registers are private to each thread and have very low latency
C. Registers are slower than global memory
D. Registers are slower than shared memory

Answer: Registers are private to each thread and have very low latency