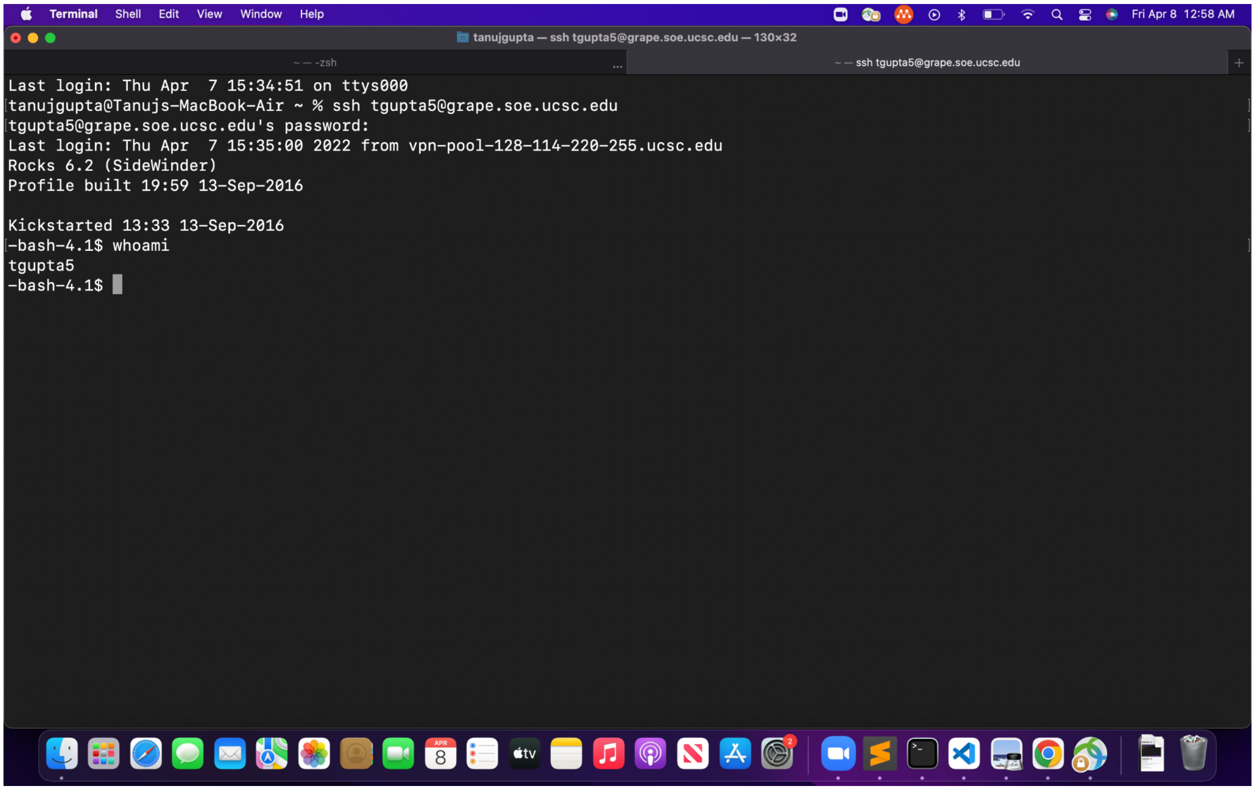
HW 1(i):



HW 1 (ii):

Supercomputer Number- 102

Name- PARAM SIDDHI-AI

Location- Center for Development of Advanced Computing (C-DAC), India

No. nodes, No. processors/node- 41664 Cores

Clock Speed- 2.25GHz

Flops/Processor and Total Flops-
Linpack Performance (Rmax): 4,619 TFlops.
Theoretical Peak (Rpeak): 5,267.14 TFlops.

Total Cores 41,664.
Flops/processor=5,267.14/41,664 TFlops/processor=0.12641945084 TFlops/processor

Memory/Processor and Total Memory
Total Memory -> 84,000 GB
Memory/Processor -> 84,000/41664 = 2.016 GB

Architecture
MIMD- NVIDIA DGX A100

Interconnect
OpenMPI 4.0.4rc3

Use
India's largest supercomputer is focused on promoting inclusive growth using AI. In this endeavor, several strategic collaborations with research institutions, academia, and industry partners have been forged – with the singular goal of developing technology, augmenting R&D, and enhancing ethical standards in the use of AI for social good.

The AI system will strengthen application development of packages in areas such as advanced materials, computational chemistry & astrophysics, and several packages being developed under the mission of platform for drug design and preventive health care system, flood forecasting packages for flood-prone metro cities like Mumbai, Delhi, Chennai, Patna, and Guwahati. It also helped accelerate R&D in other wars against COVID-19 through faster simulations, medical imaging, genome sequencing, and forecasting and is a boon for Indian masses and for start-Ups and MSMEs in particular.

Special Features
-India’s largest and fastest supercomputer.
-It’s the most powerful non-distributed computer system in the world.
-PARAM SIDDHI AI has been developed with a three-layer data security framework, to promote and advance the ethical use of AI and the need for data integrity.

HW 1 (iii):

Food stalls in a food festival are a very good example of real-life HPC architecture.

Multiple nodes
- Multiple stalls at food festivals are multiple nodes. So **multiple stalls = multiple nodes.**

Multiple Cores
- At every stall, there can be multiple stoves in which cooking can be done in parallel. Also, the order is taken separately on a separate window. These multiple stoves/order window are multiple cores in every node(stall). So **multiple stoves/order window = core.**

- Every stall can have a single person or multiple persons working simultaneously. In the case of a single person, a **stall = node with multiple cores and single memory**. In the case of multiple persons, a **stall = node with multiple cores and multiple memory units attached**.

Message passing and communication
- There is also a lot of **message passing and communication** between cores in a stall(node). A person taking orders in a stall(task1) passes messages about order details to the chef cooking(task2).

Caching/Memory/Disk
- Cooking requires raw materials. Getting these materials from the shop is like a **disk seek** which takes a lot of time. Storing these materials at home is like **memory seek**(faster but not the fastest). Storing these materials inside the fridge in the stall is like **caching**(small but fastest).

Prefetching
- Also, if some item is very popular, then we can get all the raw materials for similar items as well. This is analogous to **pre-fetching** and caters to spatial locality.

Message Queues/Asynchronous tasks

- For every stall, there are multiple queues to **match the number of tasks to the throughput of computation**. First is when people wait in queues to order, and second is when they wait in a queue to get their food.