**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC VÀ KỸ THUẬT THÔNG TIN**



**BÁO CÁO BÀI TẬP KIẾN TRÚC DỮ LIỆU LỚN**

**NHÓM 7**

**GVHD: ThS. Nguyễn Thị Anh Thư**

**1.**Nguyễn Thế Tiến Đạt **MSSV:** 22520225

**2.**Vũ Quang Huy **MSSV:** 22520587

**3.**Võ Nhất Phương **MSSV:** 22521172

**4.**Lê Nguyễn Thùy Dương **MSSV:** 22520298

**5.**Đoàn Ngọc Thanh Sơn **MSSV:** 21521385

**Tp. Hồ Chí Minh, 03/2024**

**NỘI DUNG THẢO LUẬN**:

**Bài tập:** Tìm hiểu về kiến trúc dữ liệu lớn:

* Massively parallel processing (MPP)
* NoSQL database
* Distributed Computing
* Cloud Computing

**Trả lời:**

***1/ Massively Parallel Processing (MPP)***

**Massively Parallel Processing (MPP)** là một mô hình sử dụng hàng trăm, hàng nghìn bộ xử lý để thực hiện song song các phép tính toán để xử lý lượng dữ liệu khổng lồ.

Mỗi bộ xử lý sẽ sử dụng các hệ điều hành riêng biệt nhau cùng với các thiết bị input, output riêng và không chia sẻ bộ nhớ với nhau.

Các bộ xử lý sẽ thông qua các kết nối “đặc biệt” giữa các dữ liệu để có thể gửi tín hiệu cho nhau.

Những trường hợp sử dụng MPP:

- *Data Warehousing*: các hệ thống MPP phù hợp trong việc xử lý và phân tích các tập dữ liệu lớn

- *Business Intelligence and Analytics*: MPP cho phép các doanh nghiệp thực hiện các phân tích phức tạp và tạo ra thông tin chi tiết từ lượng dữ liệu khổng lồ

- *Data Science and Machine Learning*: hệ thống MPP có thể đẩy nhanh việc đào tạo và đánh giá các mô hình học máy bằng cách phân phối các tính toán trên nhiều bộ xử lý.

- *Real-time Data Processing*: hệ thống MPP có thể xử lý các luồng dữ liệu tốc độ cao và xử lý dữ liệu thời gian thực để có thông tin chi tiết và ra quyết định ngay lập tức.

Một số CSDL phổ biến sử dụng mô hình MPP: **Teradata, Greenplum, Netezza (IBM), Redshift (AWS), Azure SQL DW (Azure), BigQuery (Google Cloud Platform), and Vertica (HP).**

- Các thành phần chính trong MPP

+ Bộ xử lý (hay các nút xử lý)

+ Kết nối với tốc độ cao: ethernet, phân tán cáp quang,…

+ Distributed Lock Manager (DLM): có nhiệm vụ điều phối phân tán tài nguyên giữa các bộ xử lý

…

*- Ưu điểm:*

+ Hiệu suất cao: nhờ lượng lớn bộ xử lý đồng thời sẽ gia tăng đáng kể hiệu suất xử lý dữ liệu

+ Khả năng mở rộng: có thể tăng thêm bộ xử lý hoặc nâng cấp CPU, bộ nhớ, lưu trữ cho từng nút, điều này sẽ dẫn tới gia tăng hiệu suất

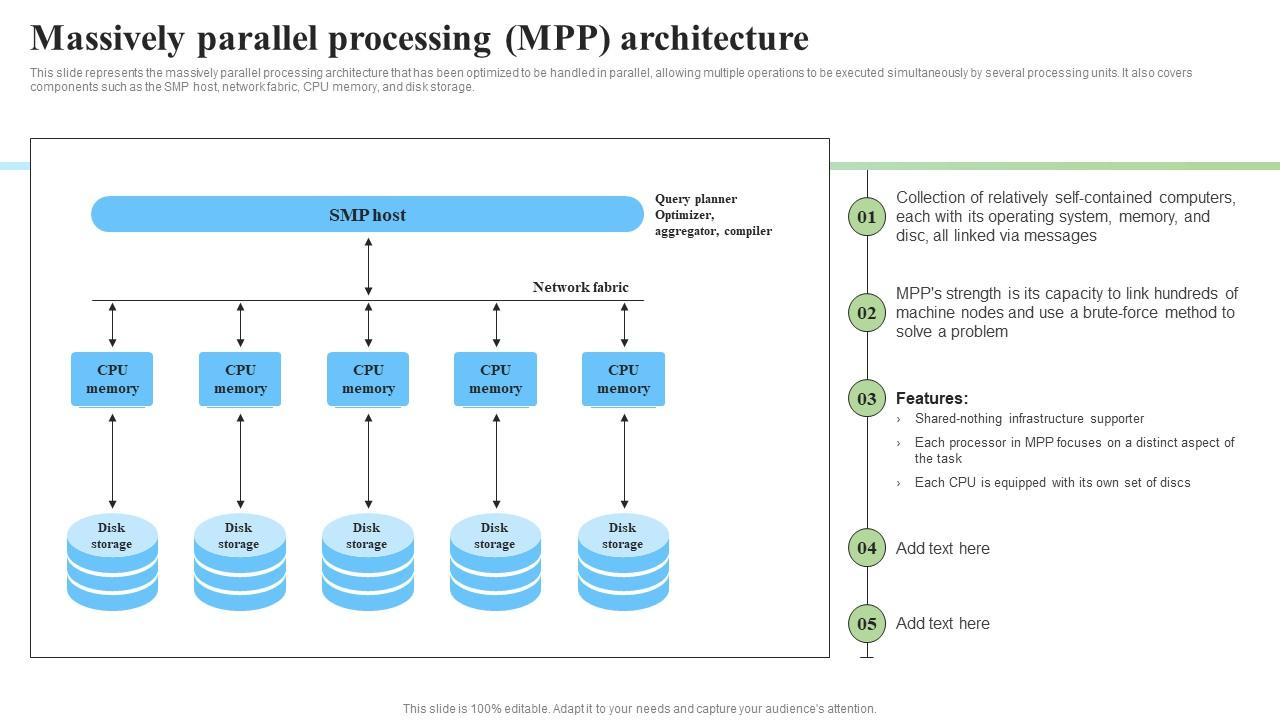
+ Nếu một trong các nút bị hư hại thì dữ liệu vẫn không bị ảnh hưởng

*- Nhược điểm:*

+ Tính phức tạp: do hệ thống đã tối ưu các phần cứng thích hợp nên việc quản lý sẽ có nhiều khó khăn khi do độ phức tạp phần cứng của từng bộ xử lý là khác nhau

+ Dữ liệu phân tán sẽ khiến người quản lý phải có khóa phù hợp để có thể tối ưu được dữ liệu

+ Mô hình này không phù hợp cho lưu trữ dữ liệu đám mây, do phần cứng phải mất nhiều thời gian để có thể thêm vào hoặc tái cấu trúc dữ liệu

****

***2/ NoSQL***

NoSql: cơ sở dữ liệu phi quan hệ, phân tán

4 loại

* Column-oriented databases: cassandra
* Graph databases: Neo4j
* Document Oriented databases: mongodb
* Key-value stores: Redis

Ví dụ về hiệu suất truy vấn nosql so với sql

**Column-oriented databases vs row-oriented databases**

Row-oriented databases: lưu dữ liệu theo hàng

Ví dụ:

| Mã số sinh viên | Tên | Tuổi |
| --- | --- | --- |
| 123 | Đạt | 18 |
| 456 | Sơn | 18 |
| 678 | Phương | 18 |

dữ liệu sẽ được lưu như sau

{123, Đạt, 18}

{456, Sơn, 18}

{678, Phương, 18}

Column-oriented databases

Ví dụ:

| Mã số sinh viên | 123 | 456 | 789 |
| --- | --- | --- | --- |
| Tên | Đạt | Sơn | Phương |
| Tuổi | 18 | 18 | 18 |

Dữ liệu sẽ được lưu như sau

Mã số sinh viên : {123, 456, 789}

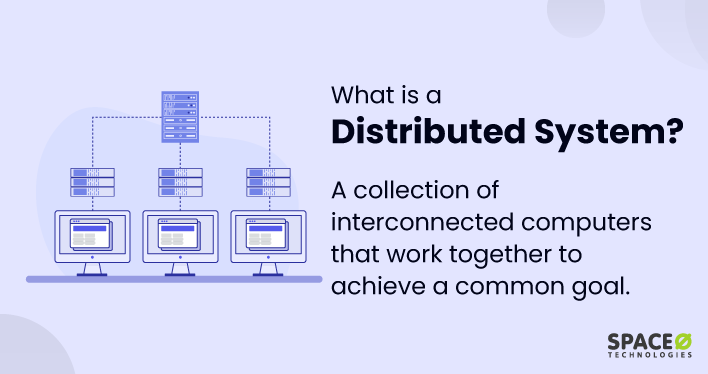
Tên : {Đạt, Sơn, Phương}

Tuổi: {18, 18, 18}

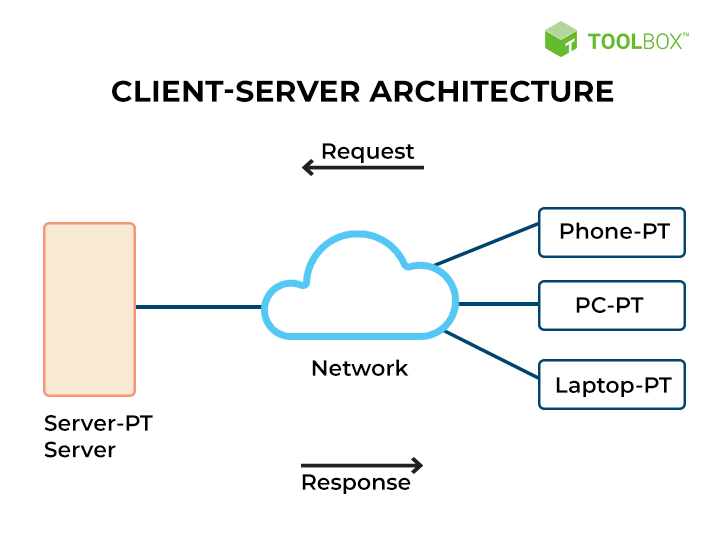
* Ưu điểm:
* NoSQL(Column-oriented databases): truy vấn chỉ cần truy xuất trên một số cột cụ thể, truy vấn tốc độ cao còn SQL truy vấn cần truy xuất toàn bộ dữ liệu hàng
* Nhược điểm:
* Không hiệu quả cho các truy vấn tính toán phức tạp trên dữ liệu cột
* Không hiệu quả cho các truy vấn tính toán phức tạp trên dữ liệu cột
* Truy cập hiệu suất không tối ưu cho các truy vấn cột cụ thể
* Tốn nhiều không gian lưu trữ khi lưu trữ dữ liệu phân tán
* Khả năng nén dữ liệu thấp hơn

***3/ Distributed Computing (Điện toán phân toán)***

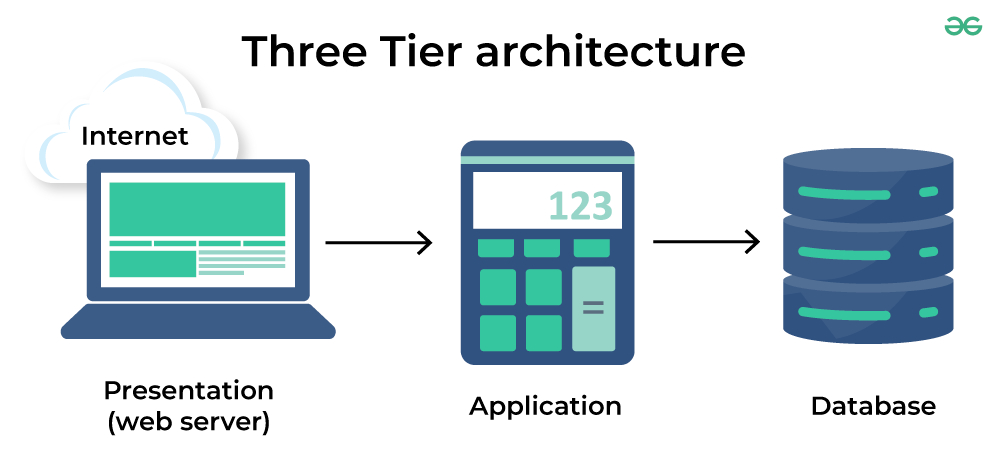
**Điện toán phân toán** là phương pháp giúp nhiều máy tính làm việc cùng nhau để giải quyết một vấn đề chung. Phương pháp này biến mạng máy tính thành một máy tính đơn lẻ mạnh mẽ, có khả năng cung cấp tài nguyên quy mô lớn để giải quyết những thách thức phức tạp.

****

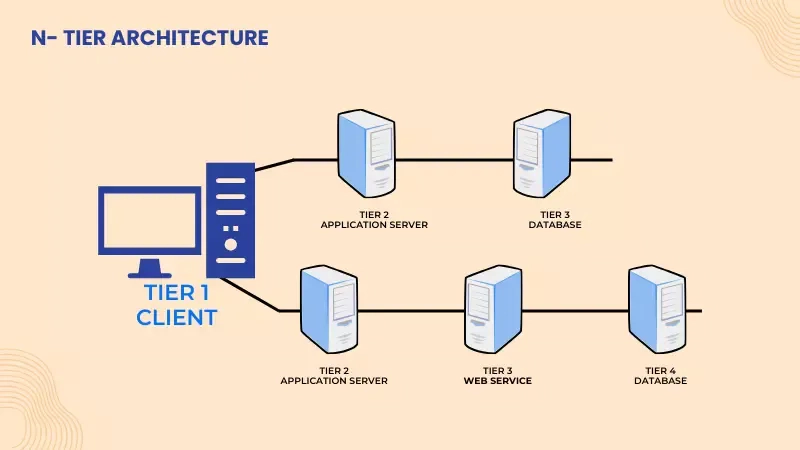
* **Ưu điểm** của Điện toán phân toán
* Khả năng điều chỉnh quy mô: Có thể phát triển theo khối lượng công việc và yêu cầu của bạn, tức là nhiều thiết bị điện toán hơn, vào mạng điện toán phân tán khi cần
* Mức độ sẵn sàng: Không bị sập nếu một trong các máy tính gặp sự cố
* Tính nhất quán: Hệ thống phân tán chia sẻ thông tin và sao chép dữ liệu với nhau, nhưng hệ thống sẽ tự động quản lý tính nhất quán của dữ liệu trên tất cả các máy tính khác nhau.
* Tính minh bạch: hệ thống phân tán chia sẻ thông tin và sao chép dữ liệu với nhau, nhưng hệ thống sẽ tự động quản lý tính nhất quán của dữ liệu trên tất cả các máy tính khác nhau.
* Độ hiệu quả: hiệu suất nhanh hơn với mức sử dụng tài nguyên tối ưu của phần cứng cơ sở.
* Một số trường hợp sử dụng của Điện toán phân toán:
* Chăm sóc sức khỏe và khoa học đời sống:Chăm sóc sức khỏe và khoa học đời sống sử dụng điện toán phân tán để lập mô hình và mô phỏng dữ liệu khoa học đời sống phức tạp. Phân tích hình ảnh, nghiên cứu thuốc y tế và phân tích cấu trúc gen đều trở nên nhanh hơn với hệ thống phân tán. Sau đây là một số ví dụ:
* Tăng tốc độ thiết kế thuốc dựa trên cấu trúc bằng cách hiển thị các mô hình phân tử theo không gian ba chiều.
* Giảm thời gian xử lý dữ liệu bộ gen để có thông tin chuyên sâu sớm về ung thư, xơ nang và bệnh Alzheimer.
* Phát triển các hệ thống thông minh giúp bác sĩ chẩn đoán bệnh nhân bằng cách xử lý một lượng lớn hình ảnh phức tạp như MRI, X-quang và chụp CT.
* Nghiên cứu kỹ thuật: Các kỹ sư có thể mô phỏng các khái niệm vật lý và cơ học phức tạp trên hệ thống phân tán. Họ sử dụng nghiên cứu này để cải tiến thiết kế sản phẩm, xây dựng các cấu trúc phức tạp và thiết kế phương tiện nhanh hơn. Dưới đây là một số ví dụ:
* Nghiên cứu Điện toán động lực học chất lưu nghiên cứu hành vi của chất lỏng và thực hiện các khái niệm đó trong thiết kế máy bay và đua xe.
* Kỹ thuật có máy tính hỗ trợ yêu cầu các công cụ mô phỏng thiên về điện toán để kiểm tra kỹ thuật nhà máy, điện tử và hàng tiêu dùng mới.
* Dịch vụ tài chính: Các công ty Dịch vụ tài chính sử dụng hệ thống phân tán để thực hiện các mô phỏng kinh tế tốc độ cao nhằm đánh giá rủi ro danh mục đầu tư, dự đoán động thái của thị trường và hỗ trợ ra quyết định tài chính. Họ có thể tạo các ứng dụng web sử dụng thế mạnh của hệ thống phân tán để thực hiện những việc sau:
* Cung cấp phí bảo hiểm tùy chỉnh theo cá nhân, chi phí thấp
* Sử dụng cơ sở dữ liệu phân tán để hỗ trợ một cách bảo mật khối lượng rất lớn các giao dịch tài chính.
* Xác thực người dùng và bảo vệ khách hàng khỏi gian lận
* Năng lượng và môi trường: Các công ty năng lượng cần phân tích khối lượng lớn dữ liệu để cải thiện hoạt động đồng thời chuyển đổi sang các giải pháp bền vững và thân thiện với khí hậu. Họ sử dụng hệ thống phân tán để phân tích luồng dữ liệu khối lượng lớn từ một mạng cảm biến lớn và các thiết bị thông minh khác. Đây là một số tác vụ họ có thể làm:
* Truyền và hợp nhất dữ liệu địa chấn cho thiết kế cấu trúc của các nhà máy điện
* Giám sát giếng dầu theo thời gian thực để chủ động quản lý rủi ro
* Các loại kiến trúc điện toán phân tán:
* Kiến trúc khách - chủ: Khách - chủ là phương pháp tổ chức phần mềm phổ biến nhất trên hệ thống phân tán. Các chức năng được chia thành hai hạng mục: máy khách và máy chủ.
* Client (Khách hàng/Máy khách): Máy khách có thông tin và khả năng xử lý hạn chế. Thay vào đó, máy khách đưa ra các yêu cầu tới máy chủ, nơi quản lý hầu hết dữ liệu và các tài nguyên khác. Bạn có thể đưa ra các yêu cầu đối với máy khách và nó sẽ giao tiếp với máy chủ thay cho bạn.
* Máy chủ: Máy chủ đồng bộ hóa và quản lý quyền truy cập vào tài nguyên. Máy chủ phản hồi các yêu cầu của máy khách với dữ liệu hoặc thông tin trạng thái. Thông thường, một máy chủ có thể xử lý các yêu cầu từ nhiều máy.
* Lợi ích và hạn chế: Kiến trúc khách - chủ mang lại lợi ích về bảo mật và dễ dàng quản lý liên tục. Hạn chế của kiến trúc khách-chủ là máy chủ có thể gây ra tắc nghẽn giao tiếp, đặc biệt là khi nhiều máy đồng thời đưa ra yêu cầu.

**

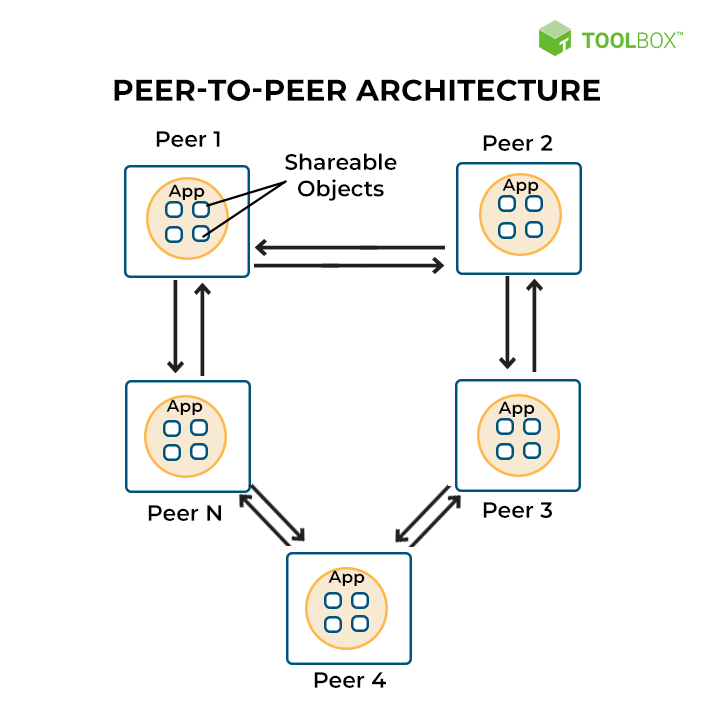
* Kiến trúc ba bậc: Trong hệ thống phân tán ba bậc, máy khách vẫn là bậc đầu tiên mà bạn truy cập. Mặt khác, máy chủ được chia thành hai hạng mục:
* Máy chủ ứng dụng: Máy chủ ứng dụng là bậc trung gian để giao tiếp. Chúng chứa logic ứng dụng hoặc các chức năng cốt lõi mà bạn thiết kế cho hệ thống phân tán.
* Máy chủ cơ sở dữ liệu: Máy chủ cơ sở dữ liệu là bậc ba để lưu trữ và quản lý dữ liệu. Chúng chịu trách nhiệm về truy xuất dữ liệu và tính nhất quán của dữ liệu.
* Bằng cách phân chia trách nhiệm máy chủ, hệ thống phân tán ba bậc giảm tắc nghẽn giao tiếp và cải thiện hiệu suất điện toán phân tán.



* Kiến trúc N bậc: Mô hình N bậc bao gồm một số hệ thống khách-chủ khác nhau giao tiếp với nhau để giải quyết cùng một vấn đề. Hầu hết các hệ thống phân tán hiện đại sử dụng kiến trúc n bậc với các ứng dụng doanh nghiệp khác nhau hoạt động cùng nhau như một hệ thống đằng sau đó.



* Kiến trúc ngang hàng: Hệ thống phân tán ngang hàng phân công trách nhiệm ngang nhau cho tất cả các máy tính nối mạng. Không có sự phân tách giữa máy khách và máy chủ, và bất kỳ máy tính nào cũng có thể thực hiện mọi trách nhiệm. Kiến trúc ngang hàng đã trở nên phổ biến để chia sẻ nội dung, truyền tệp và mạng blockchain.



* Ví dụ thực tế về Điện Toán Phân Tán:

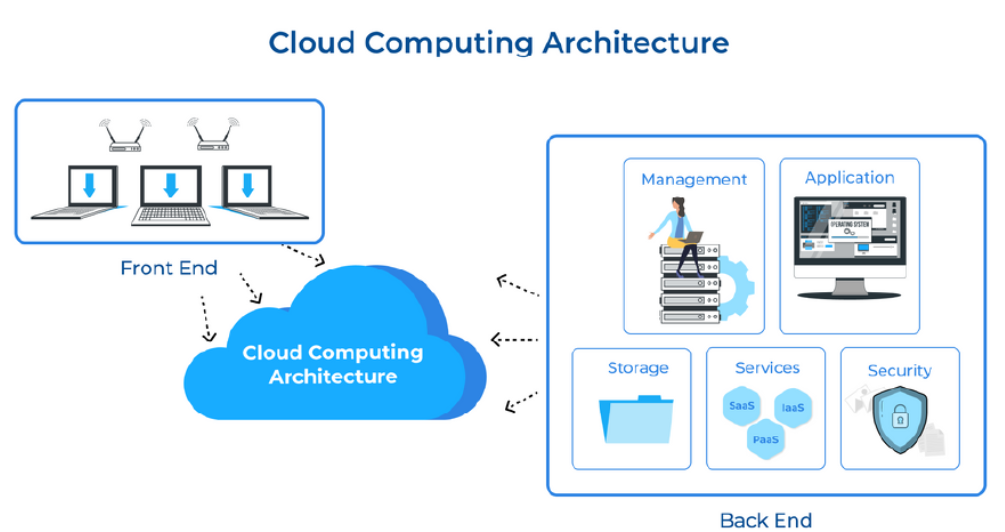
**Folding@home (Folding at home)** là dự án **tính toán phân tán (Distributed computing)** nhằm giúp các nhà khoa học phát triển các phương pháp điều trị mới cho nhiều loại bệnh bằng cách mô phỏng động học của protein. Điều này bao gồm quá trình gấp protein và các chuyển động của protein, và dựa vào các mô phỏng chạy trên máy tính cá nhân của các tình nguyện viên. Dự án hiện đang được đặt tại Đại học Pennsylvania và được lãnh đạo bởi Greg Bowman, một cựu sinh viên của Vijay Pande.

***4/ Cloud Computing (Điện toán đám mây):*** Là mô hình cung cấp dịch vụ máy tính như máy chủ, lưu trữ mạng, và ứng dụng thông qua Internet.

* Kiến trúc đám mây được chia thành 2 phần, bao gồm:
* Phía trước (Frontend): Frontend của kiến ​​trúc đám mây đề cập đến phía máy khách của hệ thống điện toán đám mây. Có nghĩa là nó chứa tất cả các giao diện người dùng và ứng dụng được khách hàng sử dụng để truy cập các tài nguyên/dịch vụ điện toán đám mây.

Ví dụ: sử dụng trình duyệt web để truy cập nền tảng đám mây.

* Client Infrastructure – Cơ sở hạ tầng khách hàng là một phần của thành phần giao diện người dùng. Nó bao gồm các ứng dụng và giao diện người dùng cần thiết để truy cập nền tảng đám mây.
* Nói cách khác, nó cung cấp GUI (Giao diện người dùng đồ họa) để tương tác với đám mây.
* Phía sau (Backend): Backend đề cập đến việc chính đám mây được sử dụng bởi nhà cung cấp dịch vụ. Nó chứa các tài nguyên cũng như quản lý tài nguyên và cung cấp các cơ chế bảo mật. Bên cạnh đó, backend cũng bao gồm bộ lưu trữ khổng lồ, ứng dụng ảo, máy ảo (VM), cơ chế kiểm soát lưu lượng, mô hình triển khai, v.v.
* Ứng dụng (Application) - Ứng dụng trong phần backend đề cập đến một phần mềm hoặc nền tảng mà khách hàng truy cập. Có nghĩa là nó cung cấp dịch vụ ở phía backend theo yêu cầu của khách hàng.
* Dịch vụ (Service) - Dịch vụ ở phía backend đề cập đến ba loại dịch vụ dựa trên đám mây chính như SaaS, PaaS và IaaS. Đồng thời nó cũng quản lý loại dịch vụ mà người dùng truy cập.
* Runtime Cloud - Runtime Cloud cung cấp nền tảng/môi trường thực thi và thời gian chạy cho máy ảo.
* Lưu trữ (Storage) - Lưu trữ ở backend cung cấp dịch vụ lưu trữ linh hoạt, khả năng mở rộng và quản lý dữ liệu đã lưu trữ.
* Cơ sở hạ tầng (Infrastructure) – Cơ sở hạ tầng đám mây đề cập đến các thành phần phần cứng và phần mềm của đám mây như bao gồm máy chủ, bộ lưu trữ, thiết bị mạng, phần mềm ảo hóa, v.v.
* Quản lý (Management) - Quản lý ở backend là việc quản lý các thành phần phụ trợ như ứng dụng, dịch vụ, runtime cloud, lưu trữ, cơ sở hạ tầng và các cơ chế bảo mật khác, v.v.
* Bảo mật (Security) - Bảo mật nói đến việc triển khai các cơ chế bảo mật khác nhau trong backend để đảm bảo an toàn cho tài nguyên, hệ thống, tệp và cơ sở hạ tầng trên đám mây cho người dùng cuối.
* Internet - Kết nối Internet đóng vai trò là phương tiện hoặc cầu nối giữa frontend và backend và thiết lập sự tương tác, giao tiếp giữa frontend và backend.
* Cơ sở dữ liệu (Database) – Cơ sở dữ liệu đề cập đến việc cung cấp cơ sở dữ liệu để lưu trữ dữ liệu có cấu trúc, chẳng hạn như cơ sở dữ liệu SQL và NOSQL. Ví dụ về các dịch vụ Cơ sở dữ liệu bao gồm Amazon RDS, cơ sở dữ liệu Microsoft Azure SQL và Google Cloud SQL.
* Kết nối mạng (Networking) – Kết nối mạng cung cấp cơ sở hạ tầng mạng cho ứng dụng trên đám mây, chẳng hạn như cân bằng tải, DNS và mạng riêng ảo.
* Phân tích (Analytics) – Phân tích trong dịch vụ backend cung cấp khả năng phân tích dữ liệu trên đám mây, chẳng hạn như trí tuệ doanh nghiệp (business intelligence) và học máy.



* Cách kiến trúc đám mây hoạt động:
* Trong kiến ​​trúc đám mây, mỗi thành phần hoạt động cùng nhau để tạo ra một nền tảng điện toán đám mây cung cấp cho người dùng quyền truy cập tài nguyên và dịch vụ theo yêu cầu.
* Backend chứa tất cả các tài nguyên, dịch vụ, lưu trữ dữ liệu và ứng dụng điện toán đám mây do nhà cung cấp dịch vụ đám mây cung cấp. Mạng được sử dụng để kết nối các thành phần kiến ​​trúc đám mây Backend và Frontend, cho phép dữ liệu được chuyển đổi giữa chúng. Khi người dùng tương tác với Frontend (giao diện phía khách (middleware), nơi mô hình dịch vụ thực hiện nhiệm vụ hoặc yêu cầu cụ thể.
* Có ba mô hình dịch vụ điện toán đám mây chính:
* Cơ sở hạ tầng dưới dạng dịch vụ (IaaS)
* Nền tảng dưới dạng dịch vụ (PaaS)
* Phần mềm dưới dạng dịch vụ (SaaS)
* Các lớp kiến trúc đám mây:
* Phần cứng (Hardware): Máy chủ, bộ lưu trữ, thiết bị mạng và phần cứng khác cung cấp năng lượng cho đám mây.
* Ảo hóa (Virtualization): Tạo ra một lớp trừu tượng (virtual layer) giữa phần cứng vật lý và các ứng dụng, hệ điều hành, hoặc người dùng cuối. Điều này cho phép nhiều ứng dụng sử dụng cùng một tài nguyên.
* Ứng dụng và dịch vụ: Lớp này điều phối và hỗ trợ các yêu cầu từ giao diện người dùng frontend, cung cấp các dịch vụ khác nhau dựa trên mô hình dịch vụ đám mây, từ phân bổ tài nguyên đến các công cụ phát triển ứng dụng và các ứng dụng dựa trên web.
* Lợi ích của Kiến trúc Điện toán đám mây
* Giúp cho tổng thể hệ thống điện toán đám mây đơn giản hơn.
* Cải thiện yêu cầu xử lý dữ liệu.
* Mang lại tính bảo mật cao.
* Khả năng khắc phục sau thảm họa tốt hơn.
* Tính khả dụng cao cho người dùng.
* Giảm chi phí vận hành CNTT.
* Cung cấp độ tin cậy cao.
* Khả năng mở rộng linh hoạt.

***!!!So sánh các tool Cloud Computing phổ biến hiện nay: AWS, Google Cloud, IBM, Alibaba***

| Tiêu chí | AWS | Azure | Google Cloud |
| --- | --- | --- | --- |
| Logo Thương hiệu |  |  |  |
| Thị phần | 33% | 18% | 9% |
| Tốt với số lượng services (Suite of services) | 200+ | 200+ | 100+ |
| Cloud regions (khu vực có cloud) | 31 | 60+ | 37 |
| Availability zones ( tính availability – sẵn sàng) | 99 | 3 cái mỗi region | 112 |
| Độ phủ sóng (Global coverage) | 245 quốc gia và vùng lãnh thổ | 140 quốc gia | 200+ quốc gia và vùng lãnh thổ |
| Compliance offerings (dịch vụ cung cấp) | 143 | 100+ | 100+ |
| Compute (tính toán) | Amazon Elastic Compute Cloud | Azure Virtual Machines, Azure Container Instances, Azure App Service, etc | Google Compute Engine |
| Relational and non-relational databases (Database RDS và non RDS) | 15 | 10 | 9 |
| Storage (Lưu trữ) | Simple Storage Service | Azure Blob Storage | Cloud Storage |
| Hybrid and multi-cloud strategy support | AWS Outposts | Azure Stack | Anthos |
| IoT | AWS IoT Core | Azure IoT Hub | Cloud IoT Core |
| Serverless | AWS Lambda | Azure Functions | Cloud Run |
| AI/ML (trí tuệ nhân tạo, học máy) | Amazon SageMaker, Amazon Augmented AI, Amazon Forecast, Amazon Personalize, … | Applied AI Services, Amazon Bot Services, Azure Cognitive Services, Anomaly Detector, Kinect DK | Vertex AI, Speech-to-Text, Text-to-Speech, Translation AI, Dialogflow |
| Giá cả | Pay-as-you-go | Pay-as-you-go | Pay-as-you-go |
| Có free tier | Free account + 12-month | Free account + $200 credit to spend in the first 30 days | Free account + $300 credit to spend |