Horizontal scaling across multiple machines, hay còn gọi là "scale out" là một chiến lược mở rộng hệ thống bằng cách thêm các máy chủ hoặc các node mới vào hệ thống thay vì nâng cấp tài nguyên của các máy chủ hiện có. Chiến lược này được sử dụng để tăng khả năng xử lý và chịu tải của hệ thống bằng cách phân chia công việc và dữ liệu ra giữa nhiều máy chủ hoặc node.

Một số điểm chính của horizontal scaling bao gồm:

1. **Phân chia công việc**: Hệ thống được phân chia thành các phần nhỏ hơn và mỗi phần được xử lý bởi một máy chủ hoặc node riêng biệt. Điều này giúp giảm bớt áp lực đối với mỗi máy chủ hoặc node cụ thể.
2. **Tăng khả năng mở rộng**: Horizontal scaling cho phép hệ thống tăng khả năng mở rộng một cách linh hoạt bằng cách thêm các máy chủ hoặc node mới vào hệ thống khi nhu cầu tăng lên.
3. **Tăng tính sẵn sàng**: Với horizontal scaling, nếu một máy chủ hoặc node gặp sự cố, các máy chủ hoặc node khác trong hệ thống vẫn có thể tiếp tục hoạt động, giúp đảm bảo tính sẵn sàng của hệ thống.
4. **Tối ưu hiệu suất**: Horizontal scaling có thể giúp tối ưu hiệu suất bằng cách phân chia tải công việc ra giữa nhiều máy chủ hoặc node, giảm thiểu thời gian đáp ứng và tăng khả năng xử lý tổng thể của hệ thống.
5. **Chi phí hiệu quả**: Thay vì nâng cấp tài nguyên của các máy chủ hiện có, horizontal scaling cho phép bạn chỉ chi trả cho các máy chủ hoặc node mới khi cần thiết, giúp tối ưu hóa chi phí.

Trong ngữ cảnh của cơ sở dữ liệu, horizontal scaling thường được áp dụng để tăng khả năng xử lý của cơ sở dữ liệu bằng cách phân chia dữ liệu ra giữa nhiều máy chủ hoặc node. Điều này giúp cải thiện hiệu suất và tính sẵn sàng của cơ sở dữ liệu khi có tải công việc cao.

Query parallelization across servers, hay còn được gọi là horizontal scaling, là một phương pháp để tăng hiệu suất và khả năng mở rộng của hệ thống cơ sở dữ liệu bằng cách chia nhỏ các tác vụ truy vấn và phân phối chúng đồng thời đến nhiều máy chủ khác nhau.

Cơ chế này thường được sử dụng trong các hệ thống phân tán, nơi dữ liệu được lưu trữ trên nhiều máy chủ và các truy vấn có thể được phân phối đến các máy chủ khác nhau để xử lý song song. Khi một truy vấn được gửi đến hệ thống, nó được chia thành các phần nhỏ hơn và gửi đến các máy chủ khác nhau để xử lý đồng thời. Kết quả từ các máy chủ riêng lẻ sau đó được tổng hợp hoặc kết hợp để tạo ra kết quả cuối cùng cho truy vấn.

Multi-tenancy là một kiến trúc hoặc mô hình trong đó một ứng dụng hoặc hệ thống được thiết kế để phục vụ nhiều khách hàng (hay "tenants") khác nhau trên cùng một cơ sở hạ tầng và instance của ứng dụng. Mỗi khách hàng trong mô hình multi-tenancy thường không nhận biết sự tồn tại của các khách hàng khác và được xem xét như là một thực thể riêng biệt, mặc dù chúng chia sẻ cùng một hạ tầng và tài nguyên.

Các yếu tố chính của multi-tenancy bao gồm:

1. **Phân chia logic**: Logic của ứng dụng phải được thiết kế để phân biệt và xử lý dữ liệu của từng khách hàng một cách độc lập. Điều này thường bao gồm việc sử dụng các phần mềm đa quỹ, nơi dữ liệu của mỗi khách hàng được lưu trữ và quản lý một cách riêng biệt.
2. **Tài nguyên chia sẻ**: Các khách hàng chia sẻ cùng một cơ sở hạ tầng và tài nguyên, bao gồm cả cơ sở dữ liệu, máy chủ, lưu trữ, và mạng.
3. **Tính đa chủ đề**: Mỗi khách hàng thường có thể tùy chỉnh và cấu hình ứng dụng theo nhu cầu của riêng mình, mà không ảnh hưởng đến các khách hàng khác.
4. **Quản lý đa tenant**: Hệ thống phải cung cấp các tính năng quản lý mạnh mẽ để quản lý và giám sát nhiều khách hàng một cách hiệu quả. Điều này bao gồm quản lý đăng nhập, quản lý dữ liệu, bảo mật, và tính sẵn sàng.