**HỆ PHÂN TÁN**

Đề bài:  Hãy dựa vào bài tiểu luận của em phân tích các nguyên lý của hệ phân tán đó để làm rõ tính mở, tính co dãn, tính trong suốt, tính chia sẻ tài nguyên

Bài làm:

1. Trình bày tổng quan về hệ thống OTS:

Hệ thống hỗ trợ giao thông đô thị OTS cung cấp cho người dùng thông tin về tình trạng ách tắc theo thời gian thực, hiện đang được triển khai ở những thành phố lớn của Việt Nam. Với đặc thù về cập nhật, chia sẻ thông tin ách tắc từ nhiều nguồn thông tin, trên nhiều khu vực khác nhau, hệ thống OTS mang đặc trưng của một hệ phân tán thông tin (Distributed Information System).

1. Phân tích các nguyên lí của hệ phân tán trong hệ thống OTS
   1. Tình chia sẻ tài nguyên (Making Resource Accessible)

Hệ thống OTS cập nhật thông tin về tình trạng giao thông theo thời gian thực. Chính vì vậy, bất kì người dùng nào khi sử dụng hệ thống cũng đều có thể truy cập thông tin một cách dễ dàng, nhanh chóng nhất.

* Môi trường Web: hệ thống OTS phát triển ứng dụng trên web với giao diện thân thiện, dễ hiểu, dễ sử dụng.

+ Khi người dùng mở trang web OTS tại địa chỉ <http://www.ots.edu.tf>, màn hình chính sẽ hiển thị bản đồ số cho biết tình trạng tắc đường ở các con phố, khu vực.

+ Các chức năng cho phép người dùng có thể thực hiện: tìm kiếm địa điểm, xem thông tin ách tắc, gửi thông tin ách tắc lên hệ thống, chia sẻ các thông tin đó qua việc tích hợp với các mạng xã hội nổi tiếng. Như vậy, hệ thống OTS đã tạo nên một môi trường chia sẻ thông tin rất nhanh chóng, hiệu quả.

+ Cho dù ở bất kì địa điểm nào, người dùng đều có thể truy cập hệ thống để cập nhật tình trạng giao thông theo thời gian thực. Chẳng hạn: bạn đang ở Hà Nội vẫn có thể cập nhật tình trạng giao thông ở TP.HCM. Việc đó sẽ tiết kiệm được rất nhiều chi phí đối với người sử dụng khi muốn cập nhật thông tin. Toàn bộ dữ liệu dùng chung đó sẽ được quản lí bởi Data Center, nơi đặt các cơ sở dữ liệu phân tán và các web service chuyên thực hiện việc trao đổi dữ liệu.

+ Với đặc thù lượng thông tin trao đổi rất lớn, liên tục trên mạng giữa rất nhiều người dùng khác nhau, hệ thống OTS tăng cường việc bảo mật bằng cách yêu cầu người dùng đăng kí các tài khoản cá nhân để đảm bảo tính xác thực đối tượng người dùng, tăng cường tính tin cậy và an toàn của thông tin được trao đổi và hạn chế các nguy cơ tấn công xấu đối với hệ thống.

* Ứng dụng di động (mobile):

+ Để tăng cường mức độ nhanh chóng, kịp thời trong việc cập nhật thông tin giao thông, hệ thống OTS phát triển ứng dụng trên thiết bị di động. Tại bất cứ thời điểm, địa điểm nào, người dùng đều có thể truy cập hệ thống thông qua thiết bị cầm tay (sử dụng mạng GPRS hoặc 3G)

+ Người dùng đang di chuyển có thể cập nhật và gửi thông tin về tình trạng ách tắc ngay lập tức lên hệ thống. Việc này sẽ tăng cường tính linh động (portable) trong việc trao đổi thông tin của người dùng.

* 1. Tính trong suốt (Distributed Transparency)

Mức độ trong suốt (che giấu tài nguyên hay tác vụ) của hệ thống khác nhau đối với từng nhóm đối tượng người dùng. Đối với nhóm người dùng cuối (khách vãng lai, thành viên tham gia cộng đồng), họ chỉ cần biết những thông tin về tình trạng giao thông do hệ thống cung cấp mà không biết hệ thống gồm các thành phần nào, tổ chức ra sao... Đối với ban quản trị, họ được phép tham gia vào quá trình kiểm soát thông tin giao thông, các thành phần, module làm việc bên trong của hệ thống. Đối với đội ngũ phát triển hệ thống, họ có thêm các công cụ kiểm soát hoạt động chi tiết của hệ thống như vết giao dịch (logs), lỗi hệ thống (exception)...

1. Access

* Bên trong hệ thống OTS, lượng dữ liệu được trao đổi giữa các máy chủ trạm (server) rất lớn. Mỗi một thông tin đều được biểu diễn dưới nhiều kiểu định dạng khác nhau, chứa những tham số khác nhau mà tại mỗi máy sẽ có những cơ chế để xử lí dữ liệu khác nhau. Tuy nhiên, khi cung cấp ra bên ngoài cho người dùng, hệ thống OTS đóng gói những thông tin đó theo dạng giao diện chuẩn, đáp ứng đầy đủ yêu cầu, bao gồm: Quốc gia, Thành phố, Đường phố, Thời gian, Tình trạng ách tắc. Thực tế, để có được khuôn dạng thông tin như trên, hệ thống OTS đòi hỏi việc tiếp nhận rất nhiều dạng dữ liệu như: tọa độ người dùng theo kinh độ, vĩ độ, múi giờ (timezone) tương ứng với vị trí địa lí của người dùng và các máy chủ trạm, các tệp dữ liệu chứa thông tin về quốc gia, thành phố được cung cấp bởi dịch vụ GeoLocation... Như vậy, người dùng không biết được dữ liệu bên trong của hệ thống mà chỉ biết được những thông tin đã được định dạng (format).
* Các thành phần bên trong hệ thống OTS sẽ giao tiếp với nhau thông qua phương thức gọi thủ tục từ xa RPC (Remote Procedure Call) để tăng tính trong suốt về truy xuất với phía người dùng. Toàn bộ các giao dịch này sẽ được Web Service thực thi trên đối tượng dữ liệu là các thông điệp (message) đã được định dạng.

1. Location

* Khi người dùng truy cập thông tin trên hệ thống OTS, dữ liệu sẽ được thu thập từ rất nhiều nguồn hay các máy chủ trạm khác nhau. Tuy nhiên, dữ liệu cuối cùng mà người dùng nhận được sẽ được đóng gói, hiển thị trên giao diện ứng dụng tại máy client của người dùng. Khi đó, người dùng không thể biết được dữ liệu đó được gửi lên từ máy chủ nào trong hệ thống. Việc xác định vị trí dữ liệu cần truy xuất sẽ do tầng Middleware đảm nhận với các cluster có chức năng riêng biệt: business process service (xác định tác vụ được yêu cầu), business entity service (xác định các đối tượng dữ liệu được yêu cầu), digital map service (xác định các thông tin về giao thông được yêu cầu)
* VD: thông tin ách tắc ở phố Thái Hà, Đống Đa, Hà Nội sẽ được lưu trữ ở máy chủ cụm Hà Nội. Khi người dùng gửi yêu cầu lên hệ thống, máy chủ trung tâm sẽ xác định yêu cầu, tìm dữ liệu liên quan tới tình trạng ách tắc ở phố Thái Hà nằm trên máy chủ cụm Hà Nội và trả về thông tin cho người dùng (sử dụng cơ chế Naming/Addess để xác định vị trí lưu trữ dữ liệu). Như vậy, người dùng không thể biết được cách tổ chức lưu trữ vật lí của hệ thống cũng như nguồn cung cấp dữ liệu.

1. Migration

Trong quá trình hoạt động, các thông tin/dữ liệu giao thông được các máy chủ trong hệ thống chuyển vận, trao đổi liên tục tùy thuộc vào các yêu cầu từ phía người dùng. Người dùng sẽ nhận được thông tin cuối cùng do hệ thống cung cấp nhưng không thể biết thông tin đó được truyền về từ phía máy chủ nào. Tầng Middleware sẽ đảm nhận việc di trú mã hay tiến trình bên trong hệ thống, và toàn bộ dữ liệu đi kèm sẽ được lưu vào cái tệp XML do XML Web Service đảm nhận. Như vậy, toàn bộ quá trình này sẽ trong suốt với người dùng vì chỉ có hệ thống bên dưới mới có thể xác định được dữ liệu di trú nằm ở đâu, trong tệp tin XML nào. VD: người dùng tìm kiếm thông tin ở Hà Nội sẽ nhận được dữ liệu từ máy chủ Hà Nội gửi về trung tâm, sau đó, người dùng này tìm kiếm thông tin ở TP.HCM thì sẽ nhận được dữ liệu từ máy chủ TP.HCM gửi về trung tâm. Như vậy, tại cùng 1 địa điểm, người dùng vẫn có thể nhận được thông tin từ nhiều nơi khác nhau. Quá trình di chuyển dữ liệu từ nhiều nơi khác nhau đó đều trong suốt với người dùng.

1. Relocation

Việc tổ chức, định vị dữ liệu bên trong hệ thống được che giấu hoàn toàn với người dùng. Với cơ chế sử dụng XML Web Serice, toàn bộ các thông tin giao thông cần thiết sẽ được truy xuất từ các tệp XML cụ thể với yêu cầu từ phía người dùng. VD: khi máy trạm ở Hà Nội gặp sự cố, dữ liệu sẽ được sao lưu ở máy trạm Hải Phòng (tái định vị). Khi người dùng ở Hà Nội yêu cầu, dữ liệu sẽ được trả về từ máy trạm Hải Phòng. Như vậy, người dùng không hề biết đến việc tái định vị dữ liệu của hệ thống.

1. Replication

Các thông tin giao thông luôn được hệ thống OTS nhân bản ra nhiều nhằm mục đích tăng tính chịu lỗi và tính sẵn sàng cho hệ thống. Các bản sao này được lưu dưới dạng tệp .xml, mỗi một record sẽ được tổ chức dưới dạng các element trong tệp xml theo những trường thông tin đã định dạng như: kinh độ, vĩ độ, timestamps, status...Hệ thống sử dụng cơ chế định danh cho các tệp tin xml để thuận tiện cho việc truy xuất khi có rất nhiều bản sao trên các máy chủ trạm khác nhau. Trên các máy chủ trạm chứa những dữ liệu cần thiết sau khi được nhân bản. Người dùng chỉ nhận được thể hiện cuối cùng của dữ liệu (instance) mà không thể biết dữ liệu đó được tổ chức như thế nào, đó là bản sao dữ liệu nào, có bao nhiêu bản sao dữ liệu như vậy được lưu trữ?

1. Concurency

Đây là yếu tố rất quan trọng với hệ thống phân tán OTS. Tại cùng một thời điểm, nhiều người dùng có thể truy xuất, tương tác thông tin trên cùng một khu vực. Với mỗi thông điệp (message) được trao đổi đều được gắn một nhãn thời gian (timestamp) tương ứng để xác định phiên làm việc. Cơ chế quản lí timestamp trên thông điệp sẽ được Web Service đảm nhận, các tiến trình làm việc được tổ chức song song trên máy chủ trung tâm đảm bảo tính đồng thời của giao dịch. VD: vào lúc 17h05, có 1 người báo tắc ở phố Giảng Võ, có 1 người truy cập từ mobile cập nhật tình trạng ách tắc ở phố Giảng Võ. Với 2 yêu cầu này, hệ thống sẽ xử lí độc lập, song song, đảm bảo đáp ứng yêu cầu của tất cả người dùng mà không bị tương tranh gây lỗi cho hệ thống. Mỗi một người dùng không hề biết đến việc cùng truy xuất hệ thống này mà chỉ biết thông tin/dữ liệu cần thiết được trả về theo yêu cầu. Ngoài ra, hệ thống còn có nhiều cơ chế chống chịu lỗi khi xảy ra tương tranh, thực hiện đồng thời các thao tác như: tổ chức theo phiên làm việc trên trình duyệt web...

1. Failure

Hệ thống OTS có một số biện pháp phòng tránh lỗi như backup, sao lưu, nhân bản dữ liệu...Việc đảm bảo hệ thống vẫn hoạt động hay trong suốt với người dùng khi gặp sự cố là hết sức cần thiết. Hệ thống tổ chức cơ chế đưa ra các thông điệp, thông báo về tình trạng của hệ thống mà không để người dùng biết được lỗi nằm ở đâu, tại sao...

Try

{

//Do something

}

Catch (throw Exception)

{

//Exception Handling

}

VD: khi máy chủ Hà Nội gặp sự cố, hệ thống sẽ xử dụng dữ liệu đã được nhân bản hay dữ liệu trong bộ nhớ đệm cache để tiếp tục duy trì hoạt động, không để người dùng biết được sự cố cho đến khi khắc phục được. Nếu hệ thống gặp trục trặc lớn thì sẽ có các hình thức cảnh báo, đóng ngắt để không gây phiền toái cho người dùng.

* 1. Tính mở (Openess)
* Hệ thống OTS được tổ chức bên trong dưới dạng các module và cung cấp bên ngoài các giao diện tương tác chuẩn (API) để có thể thực hiện kết nối, giao tiếp dễ dàng với các hệ thống khác. Dựa trên cơ chế giao tiếp thông qua Web Service, các dịch vụ làm việc trên hệ thống được mô tả bởi giao diện ngôn ngữ WSDL (Web Service Description Language), tổ chức theo định dạng tệp tin XML rất linh động, hỗ trợ hữu hiệu cho việc gắn thêm các thành phần mở rộng sau này. VD: hệ thống OTS có thể được tích hợp với Facebook Application, người dùng hoàn toàn có thể truy cập hệ thống OTS thông qua giao diện ứng dụng trên mạng xã hội Facebook, cập nhật thông tin trên Twitter.
* Việc tổ chức theo module cũng giúp cho việc mở rộng các thành phần chức năng hệ thống một cách dễ dàng (adaptive). Kiến trúc phần mềm được tổ chức theo các module, block theo chuẩn Enterprise Library Application Block do Microsoft đề xuất. Việc đưa thêm các module sau này là hết sức tiện lợi, dễ dàng mở rộng phạm vi, qui mô cho hệ thống.
  1. Tính co giãn (Scalability)
* Kích thước: hệ thống có thể đáp ứng được lượng người sử dụng, các truy cập rất lớn. Khi lượng người sử dụng tăng lên, thông tin tăng nhanh, cơ sở hạ tầng của hệ thống cần đáp ứng kịp thời. Lúc này, hệ thống sẽ được bổ sung thêm các máy chủ trạm phục vụ việc lưu trữ, xử lí dữ liệu một cách linh hoạt.
* Địa lí: với đặc thù về giao thông, hệ thống được tổ chức vật lí theo mô hình các cụm máy chủ (cluster server) nên việc di chuyển, mở rộng, thay thế theo điều kiện vật lí được đảm bảo nhanh chóng, độc lập, không gây ảnh hưởng tới toàn bộ hệ thống.
* Quản trị: mỗi một máy chủ sẽ được tổ chức dưới dạng một máy tính tự trị, có đủ khả năng tiếp nhận, xử lí và trao đổi thông tin với các máy khách hay với các thành phần khác trong hệ thống, dựa trên việc tổ chức thành các máy chủ cụm như đã trình bày ở trên.

1. Một số hạn chế

* Đòi hỏi lượng máy chủ đủ lớn, đủ mạnh để đáp ứng được lượng truy cập lớn.
* Việc giao dịch giữa nhiều máy chủ trong mạng có thể gây ra tắc nghẽn trên đường truyền, cần các cơ chế tổ chức phù hợp để điều phối lưu lượng đường truyền.
* Việc phân tán các máy chủ trạm cũng đòi hỏi việc tăng cường bảo mật trên các máy trạm và đường truyền cao hơn.