



(12) **BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ**

(19) **Cộng hòa xã hội chủ nghĩa Việt Nam (VN)**

CỤC SỞ HỮU TRÍ TUỆ

(11) 

1-0021476

(51)⁷ **G06F 9/445**

(13) **B**

(21) 1-2014-00415

(22) 12.03.2013

(86) PCT/CN2013/072466 12.03.2013

(87) WO2013/152648A1 17.10.2013

(30) 201210106766.9 12.04.2012 CN

(45) 26.08.2019 377

(43) 26.01.2015 322

(73) TENCENT TECHNOLOGY (SHENZHEN) COMPANY LIMITED (CN)

Room 403, East Block 2, SEG Park, Zhenxing Road, Futian District, Shenzhen, Guangdong 518044, P.R. China

(72) LIN, Daozheng (CN), NIE, Kefeng (CN), FANG, Bin (CN), LI, Shiping (CN)

(74) Công ty TNHH Sở hữu trí tuệ WINCO (WINCO CO., LTD.)

(54) **PHƯƠNG PHÁP, THIẾT BỊ ĐẦU CUỐI, VẬT LƯU TRỮ BẤT KHẢ BIẾN CÓ THỂ ĐƯỢC ĐỌC BẰNG MÁY TÍNH ĐỂ CẢI THIẾN TỐC ĐỘ CHẠY CỦA ỨNG DỤNG**

(57) Sáng chế đề xuất phương pháp, thiết bị đầu cuối, vật ghi bất khả biến có thể đọc được bằng máy tính để cải thiện tốc độ chạy của ứng dụng liên quan đến lĩnh vực máy tính. Phương pháp theo sáng chế bao gồm các bước: thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong khi ứng dụng đang chạy, trong đó thông tin dữ liệu cục bộ này bao gồm tên của tệp tin chứa dữ liệu cục bộ này, độ dịch trong tệp tin và kích thước của dữ liệu cục bộ này; tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu cục bộ thu thập được; nạp dữ liệu cục bộ tiếp nhận được khi nhận được lệnh cải thiện tốc độ chạy của ứng dụng. Giải pháp theo sáng chế cho phép giảm thời gian mà ứng dụng thực hiện các thao tác vào/ra (I/O) và cải thiện tốc độ chạy của ứng dụng bằng cách nạp dữ liệu cục bộ tiếp nhận được theo thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy; tác dụng của việc cải thiện tốc độ có thể được tăng lên bởi vì thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang là không bị giới hạn trong khi truy cập, và các mức độ tối ưu tốc độ khác nhau có thể được thực hiện đối với các ứng dụng khác nhau, nhờ đó cải thiện tính phù hợp của việc tối ưu tốc độ và mở rộng không gian tối ưu tốc độ.

Thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang khi chạy ứng dụng, trong đó, thông tin dữ liệu cục bộ này bao gồm tên của tệp tin mà trong đó dữ liệu cục bộ này được chứa, độ dịch của dữ liệu cục bộ này trong tệp tin và kích thước của dữ liệu cục bộ;

101

Thu được dữ liệu cục bộ theo thông tin dữ liệu cục bộ này, và nạp dữ liệu cục bộ này sau khi nhận được lệnh cải thiện tốc độ chạy của ứng dụng;

102

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập tới phương pháp, thiết bị đầu cuối, vật lưu trữ bất khả biến có thể đọc được bằng máy tính để cải thiện tốc độ chạy của ứng dụng.

Tình trạng kỹ thuật của sáng chế

Cùng với sự phát triển nhanh chóng của công nghệ máy tính, ngày càng có nhiều loại ứng dụng. Để thoả mãn các yêu cầu chạy ứng dụng của người dùng đối với các ứng dụng khác nhau, thì vấn đề cải thiện tốc độ chạy của ứng dụng đã thu hút được sự chú ý của nhiều nhà nghiên cứu.

Theo giải pháp kỹ thuật đã biết, cơ chế đọc trước và cơ chế nạp trước được sử dụng để cải thiện tốc độ chạy của ứng dụng. Cụ thể là cơ chế đọc trước hoạt động như sau: khi tệp tin dữ liệu cục bộ được đọc, thì nội dung đằng sau dữ liệu vốn đang được yêu cầu hiện tại cũng được đọc vào bộ nhớ, bên cạnh dữ liệu vốn đang được yêu cầu hiện tại, sao cho nội dung này có thể được đọc trực tiếp từ bộ nhớ khi được sử dụng, nhờ đó giảm thời gian thực hiện các thao tác nhập/xuất (Input/Output - I/O) cho ứng dụng; còn cơ chế nạp trước thì có dạng là nội dung chính của các ứng dụng chung sẽ được đọc khi hệ điều hành được nạp tại thời điểm khởi động, để ngăn ngừa việc tốn quá nhiều thời gian đọc dữ liệu khi các ứng dụng được khởi động.

Cơ chế đọc trước chỉ phù hợp cho tình huống mà dữ liệu cục bộ được truy cập tuần tự khi ứng dụng được nạp, còn đối với tình huống mà dữ liệu cục bộ được truy cập ngẫu nhiên khi ứng dụng đang chạy thì hiệu quả của cơ chế đọc trước là không đáng kể; và cơ chế nạp trước thì tối ưu hoá tốc độ chạy đối với tất cả ứng

dụng, điều này có tính phổ dụng nhưng lại thiếu sự phù hợp, và bị hạn chế không gian tối ưu đối với tốc độ chạy của ứng dụng.

Bản chất kỹ thuật của sáng chế

Để cải thiện sự phù hợp của việc tối ưu tốc độ trong khi cải thiện tốc độ chạy của ứng dụng và tăng cường hiệu quả của việc cải thiện tốc độ, thì sáng chế đề xuất phương pháp, thiết bị và thiết bị đầu cuối để cải thiện tốc độ chạy của ứng dụng. Giải pháp kỹ thuật theo sáng chế là như sau.

Một khía cạnh của sáng chế đề xuất phương pháp cải thiện tốc độ chạy của ứng dụng, phương pháp này bao gồm các bước:

thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong khi ứng dụng đang chạy, trong đó thông tin dữ liệu cục bộ này bao gồm tên tệp tin mà dữ liệu cục bộ được chứa trong đó, độ dịch của dữ liệu cục bộ trong tệp tin này và kích thước của dữ liệu cục bộ; và

tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu cục bộ thu thập được, và nạp dữ liệu cục bộ tiếp nhận được sau khi nhận được lệnh cải thiện tốc độ chạy của ứng dụng.

Cụ thể là, bước thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy bao gồm các bước:

thu thập sự kiện thiếu trang cứng và sự kiện thiếu trang mềm trong khi ứng dụng đang chạy nhờ cơ chế theo dõi sự kiện, và tiếp nhận thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy từ nhật kí theo dõi sự kiện vốn ghi lại sự kiện thiếu trang cứng và sự kiện thiếu trang mềm;

hoặc, giám sát hệ thống đang được gọi bởi tiến trình HOOK, tiếp nhận mã

giám sát để ghi lại thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang, và tiếp nhận, theo mã giám sát, thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy.

Tốt hơn nếu bước tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu cục bộ thu thập được bao gồm các bước:

phân tích thông tin dữ liệu cục bộ thu thập được để thu thập tên của tệp tin mà dữ liệu cục bộ được chứa trong đó, độ dịch của dữ liệu cục bộ trong tệp tin này và kích thước của dữ liệu cục bộ;

tập hợp thông tin dữ liệu cục bộ thu thập được theo tên của tệp tin mà dữ liệu cục bộ được chứa trong đó;

hợp nhất thông tin dữ liệu cục bộ đã được tập hợp theo độ dịch của dữ liệu cục bộ trong tệp tin, và chọn thông tin dữ liệu cục bộ thoả mãn điều kiện nạp trước từ thông tin dữ liệu cục bộ đã được hợp nhất làm thông tin dữ liệu nạp trước; và

tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu nạp trước được chọn.

Ngoài ra, sau bước chọn thông tin dữ liệu cục bộ thoả mãn điều kiện nạp trước từ thông tin dữ liệu cục bộ đã được hợp nhất làm thông tin dữ liệu nạp trước, thì phương pháp này còn bao gồm các bước:

lưu lại thông tin dữ liệu nạp trước được chọn theo thông tin về ứng dụng; và bước nạp dữ liệu cục bộ tiếp nhận được bao gồm các bước:

tiếp nhận thông tin dữ liệu nạp trước theo thông tin về ứng dụng, tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu nạp trước, và nạp dữ liệu cục bộ

tiếp nhận được vào bộ nhớ.

Ngoài ra, trước bước nạp dữ liệu cục bộ tiếp nhận được, thì phương pháp này còn bao gồm các bước:

xuất ra danh sách ứng dụng, và yêu cầu người dùng chọn ứng dụng cần được cải thiện tốc độ chạy từ danh sách ứng dụng này; và

nhận, theo ứng dụng mà người dùng đã chọn từ danh sách ứng dụng, lệnh để cải thiện tốc độ chạy của ứng dụng và thông tin về ứng dụng.

Khía cạnh khác của sáng chế cũng đề xuất thiết bị để cải thiện tốc độ chạy của ứng dụng, thiết bị này bao gồm:

môđun thu thập được tạo cấu hình để thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong khi ứng dụng đang chạy, trong đó thông tin dữ liệu cục bộ này bao gồm tên tệp tin mà dữ liệu cục bộ được chứa trong đó, độ dịch của dữ liệu cục bộ trong tệp tin này và kích thước của dữ liệu cục bộ;

môđun tiếp nhận được tạo cấu hình để tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu cục bộ mà môđun thu thập thu thập được; và

môđun nạp được tạo cấu hình để nạp dữ liệu cục bộ mà môđun tiếp nhận tiếp nhận được sau khi nhận được lệnh cải thiện tốc độ chạy của ứng dụng.

Cụ thể là, môđun thu thập được tạo cấu hình để thu thập sự kiện thiếu trang cứng và sự kiện thiếu trang mềm trong khi ứng dụng đang chạy nhờ cơ chế theo dõi sự kiện, và để tiếp nhận thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy từ nhật kí theo dõi sự kiện vốn ghi lại sự kiện thiếu trang cứng và sự kiện thiếu trang mềm;

hoặc, môđun thu thập được tạo cấu hình để giám sát hệ thống đang được gọi

bởi tiến trình HOOK, tiếp nhận mã giám sát để ghi lại thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang, và tiếp nhận, theo mã giám sát, thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy.

Tốt hơn nếu môđun tiếp nhận bao gồm:

khối phân tích được tạo cấu hình để phân tích thông tin dữ liệu cục bộ mà môđun thu thập thu thập được để thu được tên của tệp tin mà trong đó dữ liệu cục bộ được chứa, độ dịch của dữ liệu cục bộ trong tệp tin này và kích thước của dữ liệu cục bộ;

khối tập hợp được tạo cấu hình để tập hợp thông tin dữ liệu cục bộ thu thập được theo tên của tệp tin mà trong đó dữ liệu cục bộ được chứa, vốn được phân tích bởi khối phân tích;

khối hợp nhất được tạo cấu hình để hợp nhất thông tin dữ liệu cục bộ được tập hợp bởi khối tập hợp theo độ dịch của dữ liệu cục bộ trong tệp tin được phân tích bởi khối phân tích;

khối lựa chọn được tạo cấu hình để chọn thông tin dữ liệu cục bộ thoả mãn điều kiện nạp trước từ thông tin dữ liệu cục bộ, được hợp nhất bởi khối hợp nhất, làm thông tin dữ liệu nạp trước; và

khối tiếp nhận được tạo cấu hình để tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu nạp trước được chọn bởi khối lựa chọn.

Ngoài ra, thiết bị này còn bao gồm:

môđun lưu trữ được tạo cấu hình để lưu thông tin dữ liệu nạp trước được chọn bởi khối lựa chọn theo thông tin về ứng dụng; và

môđun nạp bao gồm:

khối tiếp nhận thứ nhất được tạo cấu hình để tiếp nhận thông tin dữ liệu nạp trước theo thông tin về ứng dụng;

khối tiếp nhận thứ hai được tạo cấu hình để tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu nạp trước mà khối tiếp nhận thứ nhất tiếp nhận được; và

khối nạp được tạo cấu hình để nạp dữ liệu cục bộ, mà khối tiếp nhận thứ hai tiếp nhận được, vào bộ nhớ.

Ngoài ra, thiết bị này còn bao gồm:

môđun xuất được tạo cấu hình để xuất ra danh sách ứng dụng và yêu cầu người dùng chọn ứng dụng cần được cải thiện tốc độ chạy từ danh sách ứng dụng này; và

môđun nhận được tạo cấu hình để nhận, theo ứng dụng mà người dùng chọn từ danh sách ứng dụng được xuất ra bởi môđun xuất, lệnh để cải thiện tốc độ chạy của ứng dụng và thông tin về ứng dụng này.

Hiệu quả của giải pháp kỹ thuật theo sáng chế là nhờ việc thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy và nạp dữ liệu cục bộ tiếp nhận được theo thông tin dữ liệu cục bộ này mà thời gian thực hiện các thao tác I/O của ứng dụng có thể được giảm bớt, nhờ đó cải thiện tốc độ chạy của ứng dụng; ngoài ra, hiệu quả cải thiện tốc độ cũng có thể được tăng cường vì thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang là không bị giới hạn ở tình huống truy cập dữ liệu cục bộ; và vì thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc các ứng dụng khác nhau đang chạy là khác nhau, nên mức độ tối ưu tốc độ đối với các ứng

dụng khác nhau cũng khác nhau, nhờ đó tính phù hợp của việc tối ưu tốc độ sẽ được cải thiện và không gian tối ưu tốc độ cũng được mở rộng.

Mô tả vắn tắt các hình vẽ

Để minh họa rõ hơn giải pháp kỹ thuật của sáng chế, sau đây, các hình vẽ kèm theo, vốn cần thiết để mô tả sáng chế, sẽ được mô tả ngắn gọn. Tất nhiên là các hình vẽ kèm theo này trong phần mô tả sau đây chỉ minh họa một số phương án thực hiện sáng chế. Những người có hiểu biết trung bình trong lĩnh vực kỹ thuật này cũng có thể tạo ra các hình vẽ khác dựa vào các hình vẽ kèm theo này mà không cần đến hoạt động có tính sáng tạo nào.

Fig.1 là lưu đồ của phương pháp cải thiện tốc độ chạy của ứng dụng theo phương án thứ nhất của sáng chế;

Fig.2 là lưu đồ của phương pháp cải thiện tốc độ chạy của ứng dụng theo phương án thứ hai của sáng chế;

Fig.3 là sơ đồ cấu trúc của thiết bị để cải thiện tốc độ chạy của ứng dụng theo phương án thứ ba của sáng chế;

Fig.4 là sơ đồ cấu trúc của môđun tiếp nhận theo phương án thứ ba của sáng chế;

Fig.5 là sơ đồ cấu trúc của thiết bị khác để cải thiện tốc độ chạy của ứng dụng theo phương án thứ ba của sáng chế;

Fig.6 là sơ đồ cấu trúc của môđun nạp theo phương án thứ ba của sáng chế;

Fig.7 là sơ đồ cấu trúc của thiết bị khác để cải thiện tốc độ chạy của ứng dụng theo phương án thứ ba của sáng chế; và

Fig.8 là sơ đồ cấu trúc của thiết bị đầu cuối theo phương án thứ tư của sáng chế.

Mô tả chi tiết các phương án thực hiện sáng chế

Trong phần sau đây, các phương án thực hiện sáng chế sẽ được mô tả chi tiết hơn dựa vào các hình vẽ kèm theo để làm cho mục đích, giải pháp kỹ thuật và các ưu điểm của sáng chế rõ ràng hơn.

Phương án thực hiện thứ nhất

Phương án này đề xuất phương pháp cải thiện tốc độ chạy của ứng dụng. Dựa vào Fig.1, lưu đồ của phương pháp theo phương án này được mô tả như sau.

Bước 101: thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong khi ứng dụng đang chạy, trong đó thông tin dữ liệu cục bộ này bao gồm tên tệp tin mà dữ liệu cục bộ được chứa trong đó, độ dịch của dữ liệu cục bộ trong tệp tin này và kích thước của dữ liệu cục bộ.

Cụ thể là, cách thức để thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong khi ứng dụng đang chạy là không bị giới hạn theo phương án này, nó bao gồm, nhưng không bị giới hạn ở, những cách như sau:

thu thập sự kiện thiếu trang cứng và sự kiện thiếu trang mềm trong khi ứng dụng đang chạy nhờ cơ chế theo dõi sự kiện, và tiếp nhận thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy từ nhật kí theo dõi sự kiện vốn ghi lại sự kiện thiếu trang cứng và sự kiện thiếu trang mềm;

hoặc giám sát hệ thống đang được gọi bởi tiến trình HOOK, tiếp nhận mã giám sát để ghi lại thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu

trang, và tiếp nhận, theo mã giám sát, thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy.

Bước 102: tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu cục bộ thu thập được, và nạp dữ liệu cục bộ tiếp nhận được sau khi nhận được lệnh cải thiện tốc độ chạy của ứng dụng.

Tốt hơn nếu bước tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu cục bộ thu thập được bao gồm các bước:

phân tích thông tin dữ liệu cục bộ thu thập được để thu thập tên của tệp tin mà dữ liệu cục bộ được chứa trong đó, độ dịch của dữ liệu cục bộ trong tệp tin này và kích thước của dữ liệu cục bộ;

tập hợp thông tin dữ liệu cục bộ thu thập được theo tên của tệp tin mà dữ liệu cục bộ được chứa trong đó;

hợp nhất thông tin dữ liệu cục bộ đã được tập hợp theo độ dịch của dữ liệu cục bộ trong tệp tin, và chọn thông tin dữ liệu cục bộ thoả mãn điều kiện nạp trước từ thông tin dữ liệu cục bộ đã được hợp nhất làm thông tin dữ liệu nạp trước; và

tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu nạp trước được chọn.

Ngoài ra, sau bước chọn thông tin dữ liệu cục bộ thoả mãn điều kiện nạp trước từ thông tin dữ liệu cục bộ đã được hợp nhất làm thông tin dữ liệu nạp trước, thì phương pháp này còn bao gồm các bước:

lưu lại thông tin dữ liệu nạp trước được chọn theo thông tin về ứng dụng; và
bước nạp dữ liệu cục bộ tiếp nhận được bao gồm các bước:

tiếp nhận thông tin dữ liệu nạp trước theo thông tin về ứng dụng, tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu nạp trước, và nạp dữ liệu cục bộ tiếp nhận được vào bộ nhớ.

Ngoài ra, trước bước nạp dữ liệu cục bộ tiếp nhận được, thì phương pháp này còn bao gồm các bước:

xuất ra danh sách ứng dụng, và yêu cầu người dùng chọn ứng dụng cần được cải thiện tốc độ chạy từ danh sách ứng dụng này; và

nhận, theo ứng dụng mà người dùng đã chọn từ danh sách ứng dụng, lệnh để cải thiện tốc độ chạy của ứng dụng và thông tin về ứng dụng.

Nhờ việc thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy và nạp dữ liệu cục bộ tiếp nhận được theo thông tin dữ liệu cục bộ này, thì phương pháp theo phương án này có thể cho phép giảm thời gian thực hiện các thao tác I/O của ứng dụng, nhờ đó cải thiện tốc độ chạy của ứng dụng; ngoài ra, hiệu quả cải thiện tốc độ cũng có thể được tăng cường vì thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang là không bị giới hạn ở tình huống truy cập dữ liệu cục bộ; và vì thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc các ứng dụng khác nhau đang chạy là khác nhau, nên mức độ tối ưu tốc độ đối với các ứng dụng khác nhau cũng khác nhau, nhờ đó tính phù hợp của việc tối ưu tốc độ sẽ được cải thiện và không gian tối ưu tốc độ cũng được mở rộng.

Phương án thực hiện thứ hai

Phương án này đề xuất phương pháp cải thiện tốc độ chạy của ứng dụng. Theo phương pháp này, bằng cách thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong khi ứng dụng đang chạy và nạp dữ liệu cục

bộ tương ứng theo thông tin dữ liệu nạp trước sau khi thu được thông tin dữ liệu nạp trước bằng cách hướng dẫn thông tin dữ liệu cục bộ, thì thời gian mà ứng dụng thực hiện các thao tác I/O có thể được giảm, nhờ đó thực hiện được mục đích cải thiện tốc độ chạy của ứng dụng. Kết hợp với nội dung của phương án thứ nhất và lấy trường hợp mà dữ liệu cục bộ là dữ liệu trên đĩa làm ví dụ, sau đây, phương pháp để cải thiện tốc độ chạy của ứng dụng theo phương án này sẽ được mô tả làm ví dụ. Dựa vào Fig.2, lưu đồ của phương pháp theo phương án này được mô tả như sau.

Bước 201: thu thập thông tin dữ liệu trên đĩa tương ứng với sự gián đoạn do thiếu trang trong khi ứng dụng đang chạy, trong đó thông tin dữ liệu trên đĩa này bao gồm tên tệp tin mà dữ liệu trên đĩa được chứa trong đó, độ dịch của dữ liệu trên đĩa trong tệp tin này và kích thước của dữ liệu trên đĩa.

Cụ thể là, sự gián đoạn do thiếu trang là trường hợp mà trang cần được truy cập không nằm trong bộ nhớ, và hệ điều hành cần phải gọi trang này vào bộ nhớ để truy cập. Sự gián đoạn do thiếu trang có thể bao gồm sự thiếu trang cứng và sự thiếu trang mềm. Sự thiếu trang cứng là trường hợp khi xảy ra sự gián đoạn do thiếu trang, thì dữ liệu cần thiết không nằm trong bộ đệm và cũng không nằm trong bộ nhớ đã phân trang, và cần phải tiếp nhận dữ liệu trên đĩa thông qua các thao tác I/O. Sự thiếu trang mềm là trường hợp khi xảy ra sự thiếu trang thì dữ liệu cần thiết đã được nạp vào bộ nhớ rồi sau đó được chuyển sang ổ đĩa, do đó, cần nạp lại thông tin bộ nhớ này từ tệp tin hoán đổi trên đĩa vào bộ nhớ. Do đó, cho dù là thiếu trang cứng hay thiếu trang mềm thì sự gián đoạn do thiếu trang cũng sẽ có ảnh hưởng đến tốc độ chạy của ứng dụng. Vì vậy, để cải thiện tốc độ chạy của ứng dụng, thì phương án này áp dụng cách thức mà trong đó thông tin dữ liệu trên đĩa tương ứng với sự gián đoạn do thiếu trang sẽ được thu thập trong

lúc ứng dụng đang chạy, và dữ liệu trên đĩa tiếp nhận được theo thông tin dữ liệu trên đĩa sẽ được nạp trước ở bước sau đó, để tránh sự gián đoạn do thiếu trang, nhờ đó cải thiện tốc độ chạy của ứng dụng. Cách thức thu thập thông tin dữ liệu trên đĩa tương ứng với sự gián đoạn do thiếu trang trong khi ứng dụng đang chạy bao gồm, nhưng không bị giới hạn ở, hai cách như sau.

Cách thứ nhất: thu thập sự kiện thiếu trang cứng và sự kiện thiếu trang mềm trong khi ứng dụng đang chạy nhờ cơ chế theo dõi sự kiện, và tiếp nhận thông tin dữ liệu trên đĩa tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy từ nhật kí theo dõi sự kiện vốn ghi lại sự kiện thiếu trang cứng và sự kiện thiếu trang mềm.

Cụ thể là, cơ chế theo dõi sự kiện này có thể áp dụng kỹ thuật Event Tracing for Windows (ETW - theo dõi sự kiện đối với các cửa sổ của Windows). ETW là một kỹ thuật theo dõi sự kiện vận năng của hệ điều hành, và có ít ảnh hưởng đến hệ thống hơn so với kỹ thuật ghi nhật kí hoạt động và cảnh báo, và được sử dụng để theo dõi hiệu năng của hệ thống có tải. ETW chủ yếu được áp dụng cho ứng dụng máy chủ để thường xuyên ghi lại các sự kiện, lỗi, cảnh báo hoặc kiểm tra. ETW cung cấp cơ chế theo dõi sự kiện được kích hoạt bởi ứng dụng nào đó trong chế độ người dùng, hoặc bởi trình điều khiển thiết bị trong chế độ lỗi. Ngoài ra, ETW cũng có thể kích hoạt hoặc vô hiệu hoá bản ghi nhật kí một cách linh động để thực hiện việc theo dõi chi tiết mà không khởi động lại hệ điều hành hoặc ứng dụng. Do đó, sự kiện thiếu trang cứng và sự kiện thiếu trang mềm trong lúc ứng dụng đang chạy có thể được ETW thu thập, và có thể nhận được thông tin dữ liệu trên đĩa tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy từ nhật kí theo dõi sự kiện vốn ghi lại sự kiện thiếu trang cứng và sự kiện thiếu trang mềm.

Cách thứ hai: giám sát hệ thống đang được gọi bởi tiến trình HOOK, tiếp nhận mã giám sát để ghi lại thông tin dữ liệu trên đĩa tương ứng với sự gián đoạn do thiếu trang, và tiếp nhận, theo mã giám sát này, thông tin dữ liệu trên đĩa tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy.

Cụ thể là, tiến trình HOOK là một cơ chế xử lý thông báo của hệ điều hành Windows. Tiến trình HOOK này có thể giám sát và chặn các thông báo khác nhau trong hệ thống hoặc tiến trình nào đó. Sau khi chặn thông báo gửi tới cửa sổ đích thì tiến trình HOOK có thể xử lý thông báo này hoặc tiếp tục cho phân phối thông báo này mà không xử lý gì, hoặc buộc kết thúc quá trình phân phối thông báo này. Do đó, hệ thống đang được gọi có thể được giám sát nhờ tiến trình HOOK, và sau khi nhận được mã giám sát để ghi lại thông tin dữ liệu trên đĩa tương ứng với sự gián đoạn do thiếu trang, thì sẽ nhận được thông tin dữ liệu trên đĩa tương ứng với sự gián đoạn do thiếu trang trong khi ứng dụng đang chạy, theo mã giám sát này.

Tất nhiên là những cách thức thu thập khác cũng có thể được sử dụng ngoài hai cách thu thập nêu trên, và không bị giới hạn theo phương án này. Bất kể cách thức được sử dụng, sau khi được thu thập, thì toàn bộ thông tin dữ liệu trên đĩa tương ứng với sự gián đoạn do thiếu trang trong khi ứng dụng đang chạy có thể được biến đổi thành dữ liệu văn bản với định dạng mô tả nội dung đĩa là <file, offset, length>, trong đó “file” là tên của tệp tin mà dữ liệu trên đĩa được chứa trong đó, “offset” là độ dịch của dữ liệu trên đĩa trong tệp tin này, và “length” là kích thước của dữ liệu trên đĩa, và có thể được biểu diễn bằng byte.

Bước 202: thu thập thông tin dữ liệu nạp trước theo thông tin dữ liệu trên đĩa thu thập được.

Bước này là bước được ưu tiên, và cách thức để thu thập thông tin dữ liệu nạp

trước theo thông tin dữ liệu trên đĩa thu thập được bao gồm, nhưng không bị giới hạn ở, các bước như sau:

phân tích thông tin dữ liệu trên đĩa thu thập được để thu được tên của tệp tin mà dữ liệu trên đĩa được chứa trong đó, độ dịch của dữ liệu trên đĩa trong tệp tin này và kích thước của dữ liệu trên đĩa;

tập hợp thông tin dữ liệu trên đĩa thu thập được theo tên của tệp tin mà dữ liệu trên đĩa được chứa trong đó; và

hợp nhất thông tin dữ liệu trên đĩa đã được tập hợp theo độ dịch của dữ liệu trên đĩa trong tệp tin này, và chọn thông tin dữ liệu trên đĩa thoả mãn điều kiện nạp trước từ thông tin dữ liệu trên đĩa đã được hợp nhất làm thông tin dữ liệu nạp trước.

Cụ thể là cách thức để phân tích thông tin dữ liệu trên đĩa thu thập được là không bị giới hạn theo phương án này. Để tạo thuận lợi cho việc mô tả, theo phương án này, cách thức để thu được thông tin dữ liệu nạp trước ở bước này được minh hoạ bằng cách lấy thông tin dữ liệu trên đĩa đã được phân tích, như được thể hiện trên Bảng 1 dưới đây, làm ví dụ.

Tệp tin	Độ dịch	Độ dài (byte)
Example1. dll	0	1024
Example2. dll	4096	512
Example3. data	0	10240
Example1. dll	1024	1024
Example2. dll	0	512

Có thể thấy từ Bảng 1 nêu trên rằng 5 đoạn thông tin dữ liệu trên đĩa được phân tích, khi thông tin dữ liệu trên đĩa thu thập được được tập hợp theo tên của tệp tin mà trong đó dữ liệu trên đĩa được chứa trong thông tin dữ liệu trên đĩa, thì thông tin dữ liệu trên đĩa với tên tệp tin giống nhau có thể được tập hợp cùng nhau, ví dụ, hai đoạn thông tin dữ liệu trên đĩa với tên tệp tin là Example1.dll được tập hợp lại, và hai đoạn thông tin dữ liệu trên đĩa với tên tệp tin là Example2.dll được tập hợp lại, kết quả tập hợp thu được được thể hiện trên Bảng 2 dưới đây.

Bảng 2

Tệp tin	Độ dịch	Độ dài (byte)
Example1. dll	0	1024
	1024	1024
Example2. dll	0	512
	4096	512
Example3. data	0	10240

Khi thông tin dữ liệu trên đĩa đã được tập hợp được hợp nhất theo độ dịch của dữ liệu trên đĩa trong tệp tin, kết hợp với kết quả tập hợp trên Bảng 2, thì thông

tin dữ liệu trên đĩa tương ứng với tệp tin này có thể được hợp nhất tùy theo việc khoảng cách giữa các độ dịch của chúng có nằm trong khoảng định trước hay không. Cụ thể là khoảng định trước này có thể được thiết đặt bằng 1024, 2056, v.v., vốn không bị giới hạn theo phương án này; và ở đây, trường hợp mà khoảng định trước này bằng 2056 được lấy làm ví dụ. Đối với hai đoạn thông tin dữ liệu trên đĩa với tên tệp tin là Exampel2.dll, thì do độ dịch của một đoạn thông tin dữ liệu trên đĩa là bằng 0, và độ dịch của đoạn thông tin dữ liệu trên đĩa còn lại là 4096, nên có thể thấy rằng khoảng cách giữa các độ dịch của hai đoạn thông tin dữ liệu trên đĩa này là 4096, vượt quá khoảng định trước 2056; vì vậy, hai đoạn thông tin dữ liệu trên đĩa này không được hợp nhất.

Tuy nhiên, đối với hai đoạn thông tin dữ liệu trên đĩa với tên tệp tin là Example1.dll, thì do độ dịch của một đoạn thông tin dữ liệu trên đĩa là bằng 0 và độ dịch của đoạn thông tin dữ liệu trên đĩa còn lại là 1024 nên có thể thấy rằng khoảng cách giữa các độ dịch của hai đoạn của thông tin dữ liệu trên đĩa này là 1024, nằm trong khoảng định trước 2056; vì vậy, hai đoạn thông tin dữ liệu trên đĩa này có thể được hợp nhất, và kết quả hợp nhất được thể hiện trên Bảng 3.

Bảng 3

Tệp tin	Độ dịch	Độ dài (byte)
Example1. dll	0	2048
Example2. dll	0	512
	4096	512
Example3. data	0	10240

Với các quy trình tập hợp và hợp nhất nêu trên thì có thể thu được kết quả như được thể hiện trên Bảng 3 mỗi lần ứng dụng được chạy. Có thể thu được thông tin

dữ liệu nạp trước bằng cách thu thập thông tin dữ liệu trên đĩa tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đã được chạy nhiều lần và thực hiện thao tác hướng dẫn nêu trên đối với kết quả thu được. Tương tự như thông tin dữ liệu trên đĩa, thông tin dữ liệu nạp trước cũng được sử dụng để ghi lại thông tin dữ liệu trên đĩa của ứng dụng, vốn có thể nhận được thông qua các thao tác truy cập I/O. Các ứng dụng khác nhau thì tương ứng với các thông tin dữ liệu nạp trước khác nhau, và ngay cả đối với cùng một ứng dụng thì thông tin dữ liệu nạp trước trong các tình huống khác nhau vẫn có thể khác nhau, và các tình huống này bao gồm, nhưng không bị giới hạn ở, tình huống đang nạp và đang chạy. Nếu cần xác định xem trong tình huống nào mà ứng dụng có thể tiếp nhận dữ liệu trên đĩa thông qua các thao tác truy cập I/O, thì thông tin dữ liệu trên đĩa sẽ được thu thập trong tình huống đó, thông tin dữ liệu nạp trước có thể thu được dựa trên thông tin dữ liệu trên đĩa thu thập được và theo quy trình nêu trên, và dữ liệu trên đĩa tương ứng tiếp tục được tiếp nhận nhờ thông tin dữ liệu nạp trước này. Tình huống áp dụng được đối với phương pháp theo phương án này là không bị giới hạn theo phương án này.

Một cách tùy chọn, có thể thu được kết quả trên Bảng 3 mỗi lần ứng dụng được chạy. Nếu kết quả trên Bảng 3, vốn thu được mỗi lần ứng dụng được chạy, được hợp nhất trực tiếp, và kết quả hợp nhất được sử dụng làm thông tin dữ liệu nạp trước, thì phạm vi của dữ liệu trên đĩa được bao trùm bởi thông tin dữ liệu nạp trước này sẽ rộng. Tuy nhiên, một số dữ liệu trên đĩa không được sử dụng thường xuyên trong lúc ứng dụng đang chạy, và thực tế là chỉ có thông tin dữ liệu trên đĩa tương ứng với dữ liệu trên đĩa này là được thu thập trong quy trình thu thập; do đó, việc nạp dữ liệu trên đĩa này không giúp cải thiện tốc độ chạy của ứng dụng mà còn chiếm tài nguyên hệ thống. Để tiếp tục thu hẹp phạm vi của dữ

liệu trên đĩa được bao trùm bởi thông tin dữ liệu nạp trước với mục tiêu cải thiện tốc độ chạy của ứng dụng, để giảm số lần nạp dữ liệu không cần thiết trên đĩa và ngăn ngừa sự lãng phí tài nguyên, thì phương pháp theo phương án này sử dụng cách thức mà trong đó: sau khi thu được thông tin dữ liệu trên đĩa đã được hợp nhất thông qua quy trình nêu trên, thay vì trực tiếp sử dụng toàn bộ thông tin dữ liệu trên đĩa đã được hợp nhất này làm thông tin dữ liệu nạp trước, thì thông tin dữ liệu trên đĩa mà thoả mãn điều kiện nạp trước sẽ được chọn từ thông tin dữ liệu trên đĩa đã được hợp nhất này, sau đó, thông tin dữ liệu trên đĩa chọn được sẽ được sử dụng làm thông tin dữ liệu nạp trước.

Việc điều kiện nạp trước này có được thoả mãn hay không có thể được xác định bằng cách xác định xem số lần mà thông tin dữ liệu trên đĩa được thu thập có đạt con số định trước hay không. Ví dụ, nếu một đoạn thông tin dữ liệu trên đĩa được thu thập chỉ một lần trong một quá trình mà trong đó thao tác thu thập được thực hiện mười lần, thì có thể thấy rằng tần suất thu thập đối với đoạn thông tin dữ liệu trên đĩa này là thấp, và yêu cầu đối với dữ liệu trên đĩa tương ứng với đoạn thông tin dữ liệu trên đĩa này trong lúc ứng dụng đang chạy có thể không cao, nên đoạn thông tin dữ liệu trên đĩa này có thể được xoá khỏi thông tin dữ liệu trên đĩa thu thập được. Ngược lại, nếu một đoạn thông tin dữ liệu trên đĩa được thu thập sáu lần trong một quá trình mà trong đó thao tác thu thập được thực hiện mười lần, thì có thể thấy rằng tần suất thu thập đối với đoạn thông tin dữ liệu trên đĩa này là cao, và yêu cầu đối với dữ liệu trên đĩa tương ứng với đoạn thông tin dữ liệu trên đĩa này trong lúc ứng dụng đang chạy có thể tương đối cao, nên đoạn thông tin dữ liệu trên đĩa này có thể được giữ lại trong thông tin dữ liệu trên đĩa thu thập được. Tất nhiên, để xác định xem thông tin dữ liệu trên đĩa đã được hợp nhất có thoả mãn điều kiện nạp trước hay không, thì ngoài việc xác định xem số

lần mà thông tin dữ liệu trên đĩa được thu thập có đạt con số định trước hay không, thì các điều kiện nạp trước khác cũng có thể được chọn. Con số định trước và điều kiện nạp trước này đều không bị giới hạn theo phương án này.

Bước 203: lưu thông tin dữ liệu nạp trước thu được theo thông tin về ứng dụng.

Đối với bước này, nếu người dùng không cần chạy ứng dụng, để tránh việc chiếm tài nguyên hệ thống vốn bị gây ra do việc nạp dữ liệu trên đĩa tương ứng với ứng dụng, thì phương pháp theo phương án này sử dụng cách thức mà trong đó: sau khi thu được thông tin dữ liệu nạp trước, thay vì nạp ngay dữ liệu trên đĩa nhận được dựa trên thông tin dữ liệu nạp trước, thì sẽ xác định xem có nạp dữ liệu trên đĩa nhận được dựa trên thông tin dữ liệu nạp trước này hay không, theo yêu cầu của người dùng đối với việc chạy ứng dụng. Lý do mà thông tin dữ liệu nạp trước thu được được lưu lại theo thông tin về ứng dụng là vì mỗi ứng dụng đều có thể được hướng dẫn để nhận thông tin dữ liệu nạp trước tương ứng. Để phân biệt thông tin dữ liệu nạp trước của các ứng dụng khác nhau thì thông tin dữ liệu nạp trước tương ứng có thể được lưu theo thông tin về mỗi ứng dụng. Cụ thể, thông tin về ứng dụng bao gồm, nhưng không bị giới hạn ở, tên, đường dẫn, số phiên bản, v.v., của ứng dụng. Thông tin về ứng dụng là không bị giới hạn theo phương án này.

Thông tin dữ liệu nạp trước thu được có thể được lưu cục bộ hoặc lưu trên mạng, và tất nhiên là nó cũng có thể được lưu cục bộ và trên mạng đồng thời. Nơi lưu và cách thức lưu là không bị giới hạn theo phương án này.

Bước 204: xuất ra danh sách ứng dụng, và yêu cầu người dùng chọn ứng dụng cần được cải thiện tốc độ chạy từ danh sách ứng dụng này; và nhận, theo ứng

dụng mà người dùng đã chọn từ danh sách ứng dụng này, lệnh để cải thiện tốc độ chạy của ứng dụng và thông tin về ứng dụng.

Cụ thể là, khi danh sách ứng dụng được xuất ra, thì có thể chỉ có tên của ứng dụng được xuất ra, hoặc tên của ứng dụng cùng với biểu tượng của ứng dụng có thể được xuất ra đồng thời, cách thức xuất và nội dung được xuất ra của danh sách ứng dụng là không bị giới hạn theo phương án này. Lý do xuất ra danh sách ứng dụng và yêu cầu người dùng chọn ứng dụng cần được cải thiện tốc độ chạy từ danh sách ứng dụng là để thực hiện mục đích cải thiện lần lượt tốc độ chạy của từng ứng dụng. Ví dụ, thông tin dữ liệu nạp trước của mười ứng dụng được thu thập và được lưu lại thông qua các bước từ 201 đến 203 nêu trên, tuy nhiên, không phải tất cả mười ứng dụng này cần phải chạy trong ứng dụng thực tiễn. Do đó, để tránh sự lãng phí tài nguyên hệ thống vốn bị gây ra do việc đồng thời nạp dữ liệu trên đĩa của 10 ứng dụng, thì phương án này sử dụng chế độ người dùng chọn để nạp riêng rẽ dữ liệu trên đĩa cần thiết. Nếu một ứng dụng nào đó được người dùng chọn từ danh sách ứng dụng, thì điều này có nghĩa là người dùng muốn cải thiện tốc độ chạy của ứng dụng đó, và thao tác gửi lệnh để cải thiện tốc độ chạy của ứng dụng này và thao tác gửi thông tin về ứng dụng này sẽ được kích hoạt khi ứng dụng này được người dùng chọn từ danh sách ứng dụng, để báo hiệu cho bước sau nạp dữ liệu trên đĩa tương ứng theo lệnh và thông tin này. Cụ thể là, thông tin về ứng dụng là thông tin về ứng dụng được người dùng chọn.

Bước 205: tiếp nhận thông tin dữ liệu nạp trước theo thông tin về ứng dụng, tiếp nhận dữ liệu trên đĩa tương ứng theo thông tin dữ liệu nạp trước, và nạp dữ liệu trên đĩa tiếp nhận được này vào bộ nhớ.

Cụ thể là, cho dù thông tin dữ liệu nạp trước được lưu cục bộ hay lưu trên mạng ở bước 203 nêu trên, do thông tin này được lưu theo thông tin về ứng dụng,

nên sẽ có mối quan hệ tương ứng giữa mỗi trong số thông tin dữ liệu nạp trước và thông tin về ứng dụng. Sau khi nhận được thông tin về ứng dụng ở bước 204 nêu trên, thì có thể nhận được thông tin dữ liệu nạp trước tương ứng theo thông tin về ứng dụng. Do thông tin dữ liệu trên đĩa được ghi trong thông tin dữ liệu nạp trước, nên dữ liệu trên đĩa tương ứng có thể được tiếp nhận theo thông tin dữ liệu nạp trước.

Dữ liệu trên đĩa tiếp nhận được có thể được nạp vào bộ nhớ không chỉ trong lúc hệ thống đang rảnh mà còn trong lúc ứng dụng đang chạy, thời điểm mà dữ liệu trên đĩa được nạp là không bị giới hạn theo phương án này. Trong trường hợp mà dữ liệu trên đĩa tiếp nhận được được nạp trong lúc hệ thống đang rảnh, thì đầu tiên có thể xác định xem dữ liệu trên đĩa cần nạp có mặt trong bộ nhớ hay không, nếu xác định được rằng dữ liệu trên đĩa này là có mặt trong bộ nhớ thì không cần phải nạp dữ liệu trên đĩa này vào bộ đệm, còn nếu xác định được rằng dữ liệu trên đĩa này là không có mặt trong bộ nhớ thì bước nạp dữ liệu trên đĩa này sẽ được thực hiện. Một cách tương ứng, trong trường hợp mà dữ liệu trên đĩa tiếp nhận được được nạp khi ứng dụng được khởi động, thì đầu tiên có thể xác định xem ứng dụng được khởi động chưa, nếu xác định được rằng ứng dụng đã được khởi động thì dữ liệu trên đĩa tiếp nhận được sẽ được nạp, còn nếu ứng dụng chưa được khởi động thì không cần phải nạp dữ liệu trên đĩa tiếp nhận được này. Cho dù dữ liệu trên đĩa tiếp nhận được được nạp trong lúc hệ thống đang rảnh hay lúc ứng dụng được khởi động, thì quá trình nạp đều có thể được thực hiện bằng cách dựa vào lưu đồ nạp hiện có, và cách thức nạp là không bị giới hạn theo phương án này.

Nhờ việc thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy và nạp dữ liệu cục bộ tiếp nhận được

theo thông tin dữ liệu cục bộ này, thì phương pháp theo phương án này có thể cho phép giảm thời gian thực hiện các thao tác I/O của ứng dụng, nhờ đó cải thiện tốc độ chạy của ứng dụng; ngoài ra, hiệu quả cải thiện tốc độ cũng có thể được tăng cường vì thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang là không bị giới hạn ở tình huống truy cập dữ liệu cục bộ; và vì thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc các ứng dụng khác nhau đang chạy là khác nhau, nên mức độ tối ưu tốc độ đối với các ứng dụng khác nhau cũng khác nhau, nhờ đó tính phù hợp của việc tối ưu tốc độ sẽ được cải thiện và không gian tối ưu tốc độ cũng được mở rộng.

Phương án thực hiện thứ ba

Như được thể hiện trên Fig.3, phương án này của sáng chế đề xuất thiết bị để cải thiện tốc độ chạy của ứng dụng, thiết bị này bao gồm:

môđun thu thập 301 được tạo cấu hình để thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong khi ứng dụng đang chạy, trong đó thông tin dữ liệu cục bộ này bao gồm tên tệp tin mà dữ liệu cục bộ được chứa trong đó, độ dịch của dữ liệu cục bộ trong tệp tin này và kích thước của dữ liệu cục bộ;

môđun tiếp nhận 302 được tạo cấu hình để tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu cục bộ mà môđun thu thập 301 thu thập được; và

môđun nạp 303 được tạo cấu hình để nạp dữ liệu cục bộ mà môđun tiếp nhận 302 tiếp nhận được sau khi nhận được lệnh cải thiện tốc độ chạy của ứng dụng.

Cụ thể là, môđun thu thập 301 được tạo cấu hình để thu thập sự kiện thiếu trang cứng và sự kiện thiếu trang mềm trong khi ứng dụng đang chạy nhờ cơ chế theo dõi sự kiện, và để tiếp nhận thông tin dữ liệu cục bộ tương ứng với sự gián

đoạn do thiếu trang trong lúc ứng dụng đang chạy từ nhật kí theo dõi sự kiện vốn ghi lại sự kiện thiếu trang cứng và sự kiện thiếu trang mềm;

hoặc, môđun thu thập 301 được tạo cấu hình để giám sát hệ thống đang được gọi bởi tiến trình HOOK, tiếp nhận mã giám sát để ghi lại thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang, và tiếp nhận, theo mã giám sát, thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy.

Cách thức để môđun thu thập 301 thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong khi ứng dụng đang chạy có thể được tìm thấy ở phần mô tả liên quan ở bước 201 của phương án thứ hai nêu trên nên sẽ không được mô tả lại ở đây nữa.

Như được thể hiện trên Fig.4, tốt hơn nếu môđun tiếp nhận 302 bao gồm:

khối phân tích 302a được tạo cấu hình để phân tích thông tin dữ liệu cục bộ mà môđun thu thập 301 thu thập được để thu được tên của tệp tin mà trong đó dữ liệu cục bộ được chứa, độ dịch của dữ liệu cục bộ trong tệp tin này và kích thước của dữ liệu cục bộ;

khối tập hợp 302b được tạo cấu hình để tập hợp thông tin dữ liệu cục bộ thu thập được theo tên của tệp tin mà trong đó dữ liệu cục bộ được chứa, vốn được phân tích bởi khối phân tích 302a;

khối hợp nhất 302c được tạo cấu hình để hợp nhất thông tin dữ liệu cục bộ được tập hợp bởi khối tập hợp 302b theo độ dịch của dữ liệu cục bộ trong tệp tin được phân tích bởi khối phân tích 302a;

khối lựa chọn 302d được tạo cấu hình để chọn thông tin dữ liệu cục bộ thoả mãn điều kiện nạp trước từ thông tin dữ liệu cục bộ, vốn được hợp nhất bởi khối

hợp nhất 302c, làm thông tin dữ liệu nạp trước; và

khởi tiếp nhận 302e được tạo cấu hình để tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu nạp trước được chọn bởi khối lựa chọn 302d.

Cụ thể là, cách thức để khối phân tích 302a phân tích thông tin dữ liệu cục bộ, cách thức để khối tập hợp 302b tập hợp thông tin dữ liệu cục bộ thu thập được, cách thức để khối hợp nhất 302c hợp nhất thông tin dữ liệu cục bộ vốn được tập hợp bởi khối tập hợp 302b, và cách thức để khối lựa chọn 302d chọn thông tin dữ liệu cục bộ mà thoả mãn điều kiện nạp trước từ thông tin dữ liệu cục bộ được hợp nhất bởi khối hợp nhất 302c là có thể được tìm thấy ở phần mô tả liên quan ở bước 202 của phương án thứ hai nêu trên nên sẽ không được mô tả lại ở đây nữa.

Ngoài ra, như được thể hiện trên Fig.5, thiết bị này còn bao gồm:

môđun lưu trữ 304 được tạo cấu hình để lưu thông tin dữ liệu nạp trước được chọn bởi khối lựa chọn 302d theo thông tin về ứng dụng;

và như được thể hiện trên Fig.6, môđun nạp 303 bao gồm:

khởi tiếp nhận thứ nhất 3031 được tạo cấu hình để tiếp nhận thông tin dữ liệu nạp trước, vốn được lưu bởi môđun lưu trữ 304, theo thông tin về ứng dụng;

khởi tiếp nhận thứ hai 3032 được tạo cấu hình để tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu nạp trước mà khởi tiếp nhận thứ nhất 3031 tiếp nhận được; và

khối nạp 3033 được tạo cấu hình để nạp dữ liệu cục bộ, mà khởi tiếp nhận thứ hai 3032 tiếp nhận được, vào bộ nhớ.

Cụ thể là cách thức để môđun lưu trữ 304 lưu thông tin dữ liệu nạp trước có thể được tìm thấy ở phần mô tả liên quan ở bước 203 của phương án thứ hai nêu

trên nên sẽ không được mô tả lại ở đây nữa. Cách thức để khởi tiếp nhận thứ nhất 3031 tiếp nhận thông tin dữ liệu nạp trước, cách thức để khởi tiếp nhận thứ hai 3032 tiếp nhận dữ liệu cục bộ, và cách thức để khởi nạp 3033 nạp dữ liệu cục bộ là có thể được tìm thấy ở phần mô tả liên quan ở bước 205 của phương án thứ hai nêu trên nên sẽ không được mô tả lại ở đây nữa.

Ngoài ra, như được thể hiện trên Fig.7, thiết bị này còn bao gồm:

môđun xuất 305 được tạo cấu hình để xuất ra danh sách ứng dụng và yêu cầu người dùng chọn ứng dụng cần được cải thiện tốc độ chạy từ danh sách ứng dụng này; và

môđun nhận 306 được tạo cấu hình để nhận, theo ứng dụng mà người dùng chọn từ danh sách ứng dụng được xuất ra bởi môđun xuất 305, lệnh để cải thiện tốc độ chạy của ứng dụng và thông tin về ứng dụng này.

Cách thức để môđun xuất 305 xuất ra danh sách ứng dụng và cách thức để môđun nhận 306 nhận lệnh để cải thiện tốc độ chạy của ứng dụng và thông tin về ứng dụng là có thể được tìm thấy ở phần mô tả liên quan ở bước 204 của phương án thứ hai nêu trên nên sẽ không được mô tả lại ở đây nữa.

Nhờ việc thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy và nạp dữ liệu cục bộ tiếp nhận được theo thông tin dữ liệu cục bộ này, thì thiết bị theo phương án này có thể cho phép giảm thời gian thực hiện các thao tác I/O của ứng dụng, nhờ đó cải thiện tốc độ chạy của ứng dụng; ngoài ra, hiệu quả cải thiện tốc độ cũng có thể được tăng cường vì thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang là không bị giới hạn ở tình huống truy cập dữ liệu cục bộ; và vì thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc các ứng dụng khác nhau

đang chạy là khác nhau, nên mức độ tối ưu tốc độ đối với các ứng dụng khác nhau cũng khác nhau, nhờ đó tính phù hợp của việc tối ưu tốc độ sẽ được cải thiện và không gian tối ưu tốc độ cũng được mở rộng.

Phương án thực hiện thứ tư

Như được thể hiện trên Fig.8, phương án này đề xuất thiết bị đầu cuối. Fig.8 thể hiện lược đồ cấu trúc của thiết bị đầu cuối có bề mặt nhạy tiếp xúc theo phương án này của sáng chế, và thiết bị đầu cuối này có thể được sử dụng để thực hiện phương pháp cải thiện tốc độ chạy của ứng dụng theo các phương án đã được mô tả trên đây.

Thiết bị đầu cuối 800 có thể bao gồm các thành phần như mạch tần số vô tuyến (Radio Frequency - RF) 110, bộ nhớ 120 gồm một hoặc nhiều phương tiện lưu trữ đọc được bằng máy tính, khối nhập liệu 130, khối hiển thị 140, bộ cảm biến 150, mạch audio 160, môđun truyền 170, bộ xử lý 180 gồm một hoặc nhiều lõi xử lý, và nguồn cấp 190. Những người có hiểu biết trung bình trong lĩnh vực kỹ thuật này cần hiểu rằng thiết bị đầu cuối này là không bị giới hạn ở cấu trúc như được thể hiện trên Fig.8, và có thể bao gồm các thành phần khác ngoài các thành phần được minh họa này, tổ hợp giữa một số thành phần, hoặc cách bố trí khác của các thành phần này.

Mạch RF 110 có thể được sử dụng để nhận và gửi thông tin, hoặc nhận hoặc gửi tín hiệu trong một cuộc gọi. Cụ thể là, mạch RF 110 nhận thông tin tuyến xuống từ trạm gốc và phân phối thông tin nhận được cho một hoặc nhiều bộ xử lý 180 để xử lý; ngoài ra, mạch RF 110 còn gửi dữ liệu tuyến lên đến trạm gốc này. Nói chung, mạch RF 110 bao gồm, nhưng không bị giới hạn ở, ăng ten, ít nhất một bộ khuếch đại, bộ cộng hưởng, một hoặc nhiều bộ dao động, thẻ SIM

(Subscriber Identity Module - môđun nhận dạng thuê bao), bộ thu phát, bộ ghép, bộ khuếch đại nhiễu thấp (Low Noise Amplifier - LNA), bộ song công, v.v.. Ngoài ra, mạch RF 110 cũng có thể truyền thông với mạng hoặc các thiết bị khác bằng kỹ thuật truyền thông không dây. Kỹ thuật truyền thông không dây này có thể sử dụng tiêu chuẩn hoặc giao thức truyền thông bất kì, bao gồm, nhưng không bị giới hạn ở, giao thức GSM (Global System of Mobile communication - hệ thống truyền thông di động toàn cầu), giao thức GPRS (General Packet Radio Service - dịch vụ vô tuyến gói chung), giao thức CDMA (Code Division Multiple Access - đa truy nhập phân chia theo mã), giao thức WCDMA (Wideband Code Division Multiple Access - đa truy cập phân chia theo mã băng rộng), giao thức LTE (Long Term Evolution - phát triển lâu dài), e-mail, SMS (Short Messaging Service - dịch vụ tin nhắn ngắn), v.v..

Bộ nhớ 120 có thể được sử dụng để lưu các chương trình và các môđun phần mềm, ví dụ, chương trình và môđun phần mềm tương ứng với thiết bị cải thiện tốc độ chạy của ứng dụng theo phương án thứ ba. Bộ xử lý 180 thực hiện các chức năng và các quá trình xử lý dữ liệu khác nhau, chẳng hạn cải thiện tốc độ chạy của ứng dụng, bằng cách chạy các chương trình và các môđun phần mềm được lưu trong bộ nhớ 120. Bộ nhớ 120 có thể chủ yếu bao gồm vùng bộ nhớ chương trình và vùng bộ nhớ dữ liệu. Cụ thể là, vùng bộ nhớ chương trình có thể chứa hệ điều hành, ứng dụng cần thiết cho ít nhất một chức năng (chẳng hạn chức năng phát âm thanh và chức năng mở ảnh), v.v.; và vùng bộ nhớ dữ liệu có thể chứa dữ liệu (chẳng hạn dữ liệu audio và danh bạ điện thoại) được tạo ra nhờ sử dụng thiết bị đầu cuối 800, v.v.. Ngoài ra, bộ nhớ 120 có thể bao gồm bộ nhớ truy cập ngẫu nhiên tốc độ cao, và cũng có thể bao gồm bộ nhớ bất biến, chẳng hạn ít nhất một thiết bị nhớ dạng đĩa từ, thiết bị nhớ flash, hoặc các thiết bị nhớ thể rắn

khả biến khác. Theo đó, bộ nhớ 120 cũng có thể bao gồm bộ điều khiển bộ nhớ để cho phép bộ xử lý 180 và khối nhập liệu 130 truy cập bộ nhớ 120.

Khối nhập liệu 130 có thể được sử dụng để nhận thông tin số hoặc thông tin ký tự được nhập vào, và tạo ra tín hiệu nhập bằng bàn phím, chuột, cần gạt, tín hiệu nhập quang học hoặc tín hiệu nhập bằng bi xoay, liên quan đến các thiết đặt và việc điều khiển chức năng của người dùng. Cụ thể là khối nhập liệu 130 có thể bao gồm bề mặt nhạy tiếp xúc 131 và các thiết bị nhập liệu 132 khác. Bề mặt nhạy tiếp xúc 131, còn được gọi là màn hình cảm ứng hoặc bảng cảm ứng, có thể nhận các thao tác chạm (ví dụ, các thao tác trên bề mặt nhạy tiếp xúc 131 hoặc ở gần bề mặt nhạy tiếp xúc 131, mà người dùng thực hiện nhờ sử dụng đồ vật hoặc phụ kiện thích hợp bất kỳ, chẳng hạn ngón tay và bút cảm ứng) trên đó hoặc xung quanh đó mà người dùng thực hiện, sau đó điều khiển thiết bị kết nối tương ứng theo chương trình định trước. Một cách tùy chọn, bề mặt nhạy tiếp xúc 131 có thể bao gồm thiết bị dò sự kiện chạm và bộ điều khiển chạm. Cụ thể là thiết bị dò sự kiện chạm sẽ dò thao tác chạm của người dùng và tín hiệu sinh ra do thao tác chạm này, rồi truyền tín hiệu này đến bộ điều khiển chạm; và bộ điều khiển chạm sẽ nhận thông tin chạm từ thiết bị dò sự kiện chạm, biến đổi thông tin này thành các tọa độ tiếp xúc, sau đó truyền các tọa độ tiếp xúc này đến bộ xử lý 180, ngoài ra, bộ điều khiển chạm cũng có thể nhận lệnh được gửi từ bộ xử lý 180 để thực hiện lệnh này. Ngoài ra, bề mặt nhạy tiếp xúc 131 có thể được thực hiện bằng nhiều loại cảm ứng, chẳng hạn cảm ứng điện trở, cảm ứng điện dung, cảm ứng hồng ngoại, và cảm ứng sóng âm bề mặt. Ngoài bề mặt nhạy tiếp xúc 131 ra, khối nhập liệu 130 cũng có thể bao gồm các thiết bị nhập liệu 132 khác. Cụ thể là, các thiết bị nhập liệu 132 khác này có thể bao gồm, nhưng không bị giới hạn ở, một hoặc nhiều thành phần trong số bàn phím vật lý, phím chức năng (chẳng hạn nút

điều chỉnh âm lượng và nút chuyển chế độ), bi xoay, chuột, cần gạt, v.v..

Khối hiển thị 140 có thể được sử dụng để hiển thị thông tin mà người dùng nhập vào, thông tin được cung cấp cho người dùng và các giao diện đồ họa người dùng khác nhau của thiết bị đầu cuối 800, các giao diện đồ họa người dùng này có thể được tạo ra từ các thành phần đồ họa, văn bản, biểu tượng, video, và tổ hợp bất kì giữa các thành phần này. Khối hiển thị 140 có thể bao gồm bảng hiển thị 141, và một cách tùy chọn, bảng hiển thị 141 có thể được tạo cấu hình, ví dụ, dưới dạng màn hình LCD (Liquid Crystal Display - màn hình tinh thể lỏng) hoặc màn hình OLED (Organic Light Emitting Diode - điốt phát sáng hữu cơ). Ngoài ra, bề mặt nhạy tiếp xúc 131 có thể bao trùm lên bảng hiển thị 141. Khi bề mặt nhạy tiếp xúc 131 dò thấy thao tác chạm trên đó hoặc quanh đó, thì bề mặt nhạy tiếp xúc 131 sẽ gửi thao tác chạm này đến bộ xử lý 180 để xác định loại của sự kiện chạm này; sau đó, bộ xử lý 180 tạo ra tín hiệu ra thị giác tương ứng trên bảng hiển thị 141 theo loại sự kiện chạm. Mặc dù bề mặt nhạy tiếp xúc 131 và bảng hiển thị 141 là hai thành phần riêng biệt để thực hiện chức năng nhập liệu và chức năng xuất, như được thể hiện trên Fig.8, nhưng theo một số phương án thực hiện, bề mặt nhạy tiếp xúc 131 và bảng hiển thị 141 có thể được hợp lại với nhau để thực hiện chức năng nhập liệu và chức năng xuất.

Thiết bị đầu cuối 800 cũng có thể bao gồm ít nhất một bộ cảm biến 150, chẳng hạn bộ cảm biến ánh sáng, bộ cảm biến chuyển động, và các bộ cảm biến khác. Cụ thể là bộ cảm biến ánh sáng có thể bao gồm bộ cảm biến ánh sáng môi trường và bộ cảm biến tiệm cận, trong đó bộ cảm biến ánh sáng môi trường có thể điều chỉnh độ sáng của bảng hiển thị 141 theo cường độ của ánh sáng môi trường, và bộ cảm biến tiệm cận có thể tắt bảng hiển thị 141 và/hoặc đèn nền khi thiết bị đầu cuối 800 được đưa lên tai người dùng. Một trong số các bộ cảm biến chuyển

động là bộ cảm biến gia tốc trọng trường, nó có thể dò độ lớn của gia tốc theo mỗi chiều (nói chung là theo ba trục), và có thể dò độ lớn và chiều trọng lực trong trạng thái đứng yên, do đó, nó có thể được dùng cho ứng dụng xác định tư thế của điện thoại (chẳng hạn để chuyển qua lại giữa chế độ xoay ngang và chế độ xoay dọc, cho các game liên quan, và để cân chỉnh tư thế từ kế), chức năng liên quan đến kiểu rung động (chẳng hạn máy đo bước và sự kiện chạm nhẹ vào điện thoại), v.v.. Các bộ cảm biến khác, chẳng hạn con quay hồi chuyển, khí áp kế, ẩm kế, nhiệt kế và bộ cảm biến hồng ngoại, vốn cũng có thể được trang bị cho thiết bị đầu cuối 800, không được mô tả chi tiết ở đây.

Giao diện audio giữa người dùng và thiết bị đầu cuối 800 có thể được tạo ra với mạch audio 160, loa ngoài 161 và micrô 162. Mạch audio 160 có thể biến đổi dữ liệu audio nhận được thành tín hiệu điện để truyền đến loa 161, và loa 161 sẽ biến đổi tín hiệu điện này thành tín hiệu âm thanh để xuất ra; ngược lại, micrô 162 thì biến đổi tín hiệu âm thanh nhận được thành tín hiệu điện để gửi đến mạch audio 160, và mạch audio 160 sẽ biến đổi tín hiệu điện nhận được này thành dữ liệu audio rồi xuất ra cho bộ xử lý 180, và bộ xử lý 180 xử lý dữ liệu audio này, gửi dữ liệu audio đã xử lý đến, ví dụ, thiết bị đầu cuối khác, thông qua mạch RF 110, hoặc xuất dữ liệu audio đã xử lý đến bộ nhớ 120 để xử lý tiếp. Mạch audio 160 cũng có thể bao gồm giắc cắm tai nghe để cho phép sự truyền thông giữa tai nghe ngoài và thiết bị đầu cuối 800.

Thiết bị đầu cuối 800 có thể hỗ trợ người dùng trong việc gửi và nhận e-mail, duyệt web và truy cập các luồng phương tiện, v.v., thông qua môđun truyền 170, vốn cung cấp khả năng truy cập mạng Internet bằng rộng không dây hoặc bằng dây cho người dùng. Mặc dù môđun truyền 170 được minh họa trên Fig.8 nhưng cần hiểu rằng môđun truyền 170 không phải là thành phần bắt buộc đối với thiết

bị đầu cuối 800, và nó có thể được lược bỏ nếu muốn, mà không làm thay đổi phạm vi của sáng chế.

Bộ xử lý 180, vốn là trung tâm điều khiển của thiết bị đầu cuối 800, được sử dụng để kết nối tất cả các bộ phận của chiếc điện thoại với nhau nhờ sử dụng các giao diện và các đường khác nhau, và thực hiện các chức năng khác nhau của thiết bị đầu cuối 800 và việc xử lý dữ liệu bằng cách chạy hoặc thực thi các chương trình và/hoặc các môđun phần mềm được chứa trong bộ nhớ 120 và gọi ra dữ liệu được chứa trong bộ nhớ 120, nhờ đó giám sát toàn bộ chiếc điện thoại. Một cách tùy chọn, bộ xử lý 180 có thể bao gồm một hoặc nhiều lõi xử lý; tốt hơn nếu bộ xử lý ứng dụng và bộ xử lý môđem có thể được tích hợp vào bộ xử lý 180, trong đó bộ xử lý ứng dụng thì chủ yếu xử lý hệ điều hành, giao diện người dùng, ứng dụng, v.v., còn bộ xử lý môđem thì chủ yếu xử lý việc truyền thông không dây. Cần hiểu rằng bộ xử lý môđem nêu trên có thể không được tích hợp vào bộ xử lý 180.

Thiết bị đầu cuối 800 còn bao gồm nguồn cấp 190 (ví dụ, pin) để cấp nguồn cho các thành phần riêng rẽ. Tốt hơn nếu nguồn cấp có thể được nối về mặt logic với bộ xử lý 180 thông qua hệ thống quản lý nguồn cấp, để thực hiện các chức năng như sạc điện, xả điện, và quản lý mức tiêu thụ công suất, nhờ hệ thống quản lý nguồn cấp. Nguồn cấp 190 có thể còn bao gồm một hoặc nhiều thành phần trong số nguồn cấp DC (Direct Current - điện một chiều) hoặc nguồn cấp AC (Alternating Current - điện xoay chiều), hệ thống sạc lại, mạch dò lỗi nguồn cấp, bộ chuyển đổi nguồn cấp hoặc bộ nghịch lưu, đèn báo trạng thái nguồn cấp, và các thành phần bất kì khác.

Mặc dù không được minh họa, nhưng thiết bị đầu cuối 800 cũng có thể bao gồm camera (máy ghi hình), môđun Bluetooth, v.v., vốn không được mô tả ở đây.

Theo phương án này, khối hiển thị của thiết bị đầu cuối này là màn hình cảm ứng, và thiết bị đầu cuối này còn bao gồm bộ nhớ và một hoặc nhiều chương trình được lưu trong bộ nhớ này. Một hoặc nhiều bộ xử lý nêu trên được tạo cấu hình để thực hiện các lệnh được bao gồm trong một hoặc nhiều chương trình này để thực hiện các thao tác như sau:

thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong khi ứng dụng đang chạy, trong đó thông tin dữ liệu cục bộ này bao gồm tên tệp tin mà dữ liệu cục bộ được chứa trong đó, độ dịch của dữ liệu cục bộ trong tệp tin này và kích thước của dữ liệu cục bộ; và

tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu cục bộ thu thập được, và nạp dữ liệu cục bộ tiếp nhận được sau khi nhận được lệnh cải thiện tốc độ chạy của ứng dụng.

Nếu gọi phần mô tả nêu trên là phương án khả thi thứ nhất, thì theo phương án khả thi thứ hai dựa trên phương án khả thi thứ nhất này, bộ nhớ của thiết bị đầu cuối còn bao gồm các lệnh để thực hiện các thao tác như sau:

thu thập sự kiện thiếu trang cứng và sự kiện thiếu trang mềm trong khi ứng dụng đang chạy nhờ cơ chế theo dõi sự kiện, và tiếp nhận thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy từ nhật kí theo dõi sự kiện vốn ghi lại sự kiện thiếu trang cứng và sự kiện thiếu trang mềm;

hoặc giám sát hệ thống đang được gọi bởi tiến trình HOOK, tiếp nhận mã giám sát để ghi lại thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang này, và tiếp nhận, theo mã giám sát, thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy.

Theo phương án khả thi thứ ba dựa trên phương án khả thi thứ nhất, bộ nhớ của thiết bị đầu cuối này còn bao gồm các lệnh để thực hiện các thao tác như sau:

phân tích thông tin dữ liệu cục bộ thu thập được để thu thập tên của tệp tin mà dữ liệu cục bộ được chứa trong đó, độ dịch của dữ liệu cục bộ trong tệp tin này và kích thước của dữ liệu cục bộ;

tập hợp thông tin dữ liệu cục bộ thu thập được theo tên của tệp tin mà dữ liệu cục bộ được chứa trong đó;

hợp nhất thông tin dữ liệu cục bộ đã được tập hợp theo độ dịch của dữ liệu cục bộ trong tệp tin, và chọn thông tin dữ liệu cục bộ thoả mãn điều kiện nạp trước từ thông tin dữ liệu cục bộ đã được hợp nhất làm thông tin dữ liệu nạp trước; và

tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu nạp trước được chọn.

Theo phương án khả thi thứ tư dựa trên phương án khả thi thứ ba, bộ nhớ của thiết bị đầu cuối này còn bao gồm các lệnh để thực hiện các thao tác như sau:

lưu lại thông tin dữ liệu nạp trước được chọn theo thông tin về ứng dụng; và bước nạp dữ liệu cục bộ tiếp nhận được bao gồm các bước:

tiếp nhận thông tin dữ liệu nạp trước theo thông tin về ứng dụng, tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu nạp trước, và nạp dữ liệu cục bộ tiếp nhận được vào bộ nhớ.

Theo phương án khả thi thứ năm dựa trên phương án khả thi thứ nhất, phương án khả thi thứ hai, phương án khả thi thứ ba và phương án khả thi thứ tư, thì bộ nhớ của thiết bị đầu cuối này còn bao gồm các lệnh để thực hiện các thao tác như

sau:

xuất ra danh sách ứng dụng, và yêu cầu người dùng chọn ứng dụng cần được cải thiện tốc độ chạy từ danh sách ứng dụng này; và

nhận, theo ứng dụng mà người dùng đã chọn từ danh sách ứng dụng, lệnh để cải thiện tốc độ chạy của ứng dụng và thông tin về ứng dụng.

Tóm lại, nhờ việc thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy và nạp dữ liệu cục bộ tiếp nhận được theo thông tin dữ liệu cục bộ này, thì thiết bị đầu cuối theo phương án này có thể cho phép giảm thời gian thực hiện các thao tác I/O của ứng dụng, nhờ đó cải thiện tốc độ chạy của ứng dụng; ngoài ra, hiệu quả cải thiện tốc độ cũng có thể được tăng cường vì thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang là không bị giới hạn ở tình huống truy cập dữ liệu cục bộ; và vì thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc các ứng dụng khác nhau đang chạy là khác nhau, nên mức độ tối ưu tốc độ đối với các ứng dụng khác nhau cũng khác nhau, nhờ đó tính phù hợp của việc tối ưu tốc độ sẽ được cải thiện và không gian tối ưu tốc độ cũng được mở rộng.

Phương án thực hiện thứ năm

Phương án này của sáng chế đề xuất phương tiện lưu trữ đọc được bằng máy tính, vốn có thể là phương tiện lưu trữ đọc được bằng máy tính được bao gồm trong bộ nhớ theo phương án đã được mô tả trên đây, và cũng có thể là phương tiện lưu trữ riêng biệt đọc được bằng máy tính vốn không phải của thiết bị đầu cuối. Một hoặc nhiều chương trình được chứa trong phương tiện lưu trữ đọc được bằng máy tính này và được một hoặc nhiều bộ xử lý sử dụng để thực hiện phương pháp điều khiển, và phương pháp này bao gồm các bước:

thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong khi ứng dụng đang chạy, trong đó thông tin dữ liệu cục bộ này bao gồm tên tệp tin mà dữ liệu cục bộ được chứa trong đó, độ dịch của dữ liệu cục bộ trong tệp tin này và kích thước của dữ liệu cục bộ; và

tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu cục bộ thu thập được, và nạp dữ liệu cục bộ tiếp nhận được sau khi nhận được lệnh cải thiện tốc độ chạy của ứng dụng.

Nếu gọi phần mô tả nêu trên là phương án khả thi thứ nhất, thì theo phương án khả thi thứ hai thu được dựa trên phương án khả thi thứ nhất này, bước thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy bao gồm các bước:

thu thập sự kiện thiếu trang cứng và sự kiện thiếu trang mềm trong lúc ứng dụng đang chạy nhờ cơ chế theo dõi sự kiện, và tiếp nhận thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy từ nhật kí theo dõi sự kiện vốn ghi lại sự kiện thiếu trang cứng và sự kiện thiếu trang mềm;

hoặc, giám sát hệ thống đang được gọi bởi tiến trình HOOK, tiếp nhận mã giám sát để ghi lại thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang, và tiếp nhận, theo mã giám sát, thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy.

Theo phương án khả thi thứ ba thu được dựa trên phương án khả thi thứ nhất, bước tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu cục bộ thu thập được bao gồm các bước:

phân tích thông tin dữ liệu cục bộ thu thập được để thu thập tên của tệp tin

mà dữ liệu cục bộ được chứa trong đó, độ dịch của dữ liệu cục bộ trong tệp tin này và kích thước của dữ liệu cục bộ;

tập hợp thông tin dữ liệu cục bộ thu thập được theo tên của tệp tin mà dữ liệu cục bộ được chứa trong đó;

hợp nhất thông tin dữ liệu cục bộ đã được tập hợp theo độ dịch của dữ liệu cục bộ trong tệp tin, và chọn thông tin dữ liệu cục bộ thoả mãn điều kiện nạp trước từ thông tin dữ liệu cục bộ đã được hợp nhất làm thông tin dữ liệu nạp trước; và

tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu nạp trước được chọn.

Theo phương án khả thi thứ tư thu được dựa trên phương án khả thi thứ ba, sau bước chọn thông tin dữ liệu cục bộ thoả mãn điều kiện nạp trước từ thông tin dữ liệu cục bộ đã được hợp nhất làm thông tin dữ liệu nạp trước, phương pháp này còn bao gồm các bước:

lưu lại thông tin dữ liệu nạp trước được chọn theo thông tin về ứng dụng; và bước nạp dữ liệu cục bộ tiếp nhận được bao gồm các bước:

tiếp nhận thông tin dữ liệu nạp trước theo thông tin về ứng dụng, tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu nạp trước, và nạp dữ liệu cục bộ tiếp nhận được vào bộ nhớ.

Theo phương án khả thi thứ năm thu được dựa trên phương án khả thi thứ nhất, phương án khả thi thứ hai, phương án khả thi thứ ba hoặc phương án khả thi thứ tư, trước bước nạp dữ liệu cục bộ tiếp nhận được, phương pháp này còn bao gồm các bước:

xuất ra danh sách ứng dụng, và yêu cầu người dùng chọn ứng dụng cần được cải thiện tốc độ chạy từ danh sách ứng dụng này; và

nhận, theo ứng dụng mà người dùng đã chọn từ danh sách ứng dụng, lệnh để cải thiện tốc độ chạy của ứng dụng và thông tin về ứng dụng.

Nhờ việc thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy và nạp dữ liệu cục bộ tiếp nhận được theo thông tin dữ liệu cục bộ này, thì phương tiện lưu trữ đọc được bằng máy tính theo phương án này của sáng chế có thể cho phép giảm thời gian thực hiện các thao tác I/O của ứng dụng, nhờ đó cải thiện tốc độ chạy của ứng dụng; ngoài ra, hiệu quả cải thiện tốc độ cũng có thể được tăng cường vì thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang là không bị giới hạn ở tình huống truy cập dữ liệu cục bộ; và vì thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc các ứng dụng khác nhau đang chạy là khác nhau, nên mức độ tối ưu tốc độ đối với các ứng dụng khác nhau cũng khác nhau, nhờ đó tính phù hợp của việc tối ưu tốc độ sẽ được cải thiện và không gian tối ưu tốc độ cũng được mở rộng.

Phương án thực hiện thứ sáu

Phương án này của sáng chế đề xuất giao diện đồ họa người dùng áp dụng cho thiết bị đầu cuối. Thiết bị đầu cuối này bao gồm màn hình cảm ứng, bộ nhớ và một hoặc nhiều bộ xử lý được tạo cấu hình để thực hiện một hoặc nhiều chương trình, giao diện đồ họa người dùng này có thể thực hiện các thao tác như sau:

hiển thị, trên màn hình cảm ứng, thông tin dữ liệu cục bộ thu thập được tương ứng với sự gián đoạn do thiếu trang trong khi ứng dụng đang chạy, trong đó thông tin dữ liệu cục bộ này bao gồm tên tệp tin mà dữ liệu cục bộ được chứa

trong đó, độ dịch của dữ liệu cục bộ trong tệp tin này và kích thước của dữ liệu cục bộ này; và

tiếp nhận dữ liệu cục bộ tương ứng theo thông tin dữ liệu cục bộ thu thập được, và nạp dữ liệu cục bộ tiếp nhận được sau khi nhận được lệnh cải thiện tốc độ chạy của ứng dụng.

Nhờ việc thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy và nạp dữ liệu cục bộ tiếp nhận được theo thông tin dữ liệu cục bộ này, thì giao diện đồ hoạ người dùng theo phương án này của sáng chế có thể cho phép giảm thời gian thực hiện các thao tác I/O của ứng dụng, nhờ đó cải thiện tốc độ chạy của ứng dụng; ngoài ra, hiệu quả cải thiện tốc độ cũng có thể được tăng cường vì thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang là không bị giới hạn ở tình huống truy cập dữ liệu cục bộ; và vì thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc các ứng dụng khác nhau đang chạy là khác nhau, nên mức độ tối ưu tốc độ đối với các ứng dụng khác nhau cũng khác nhau, nhờ đó tính phù hợp của việc tối ưu tốc độ sẽ được cải thiện và không gian tối ưu tốc độ cũng được mở rộng.

Cần lưu ý rằng để cải thiện tốc độ chạy của ứng dụng thì thiết bị để cải thiện tốc độ chạy của ứng dụng theo các phương án nêu trên đã được nêu làm ví dụ chỉ nhờ dựa vào nhóm môđun trong số các môđun chức năng riêng rẽ nêu trên. Trong ứng dụng thực tiễn, các chức năng nêu trên có thể được thực hiện bằng các môđun chức năng khác nhau nếu cần, tức là cấu trúc bên trong của thiết bị có thể được chia thành các môđun chức năng khác nhau, để thực hiện toàn bộ hoặc một phần trong số các chức năng nêu trên. Ngoài ra, thiết bị và thiết bị đầu cuối để cải thiện tốc độ chạy của ứng dụng theo các phương án nêu trên là có cùng nguyên lý với phương án về phương pháp để cải thiện tốc độ chạy của ứng dụng, nên

phương án thực hiện của chúng có thể được tìm thấy ở phần phương án về phương pháp, và không được mô tả lại ở đây nữa.

Các số thứ tự của các phương án nêu trên của sáng chế chỉ được sử dụng để mô tả chứ không nhằm thể hiện sự ưu việt hay sự yếu kém của mỗi phương án.

Những người có hiểu biết trung bình trong lĩnh vực kỹ thuật này cần hiểu rằng toàn bộ hoặc một phần các bước theo các phương án nêu trên có thể được thực hiện bằng phần cứng, và cũng có thể được thực hiện bằng cách ra lệnh cho phần cứng liên quan nhờ sử dụng chương trình, chương trình này có thể được chứa trong phương tiện lưu trữ đọc được bằng máy tính, và phương tiện lưu trữ này có thể là bộ nhớ chỉ đọc, đĩa từ, đĩa quang, hoặc các phương tiện tương tự.

Các từ “thứ nhất”, “thứ hai”, “thứ ba”, “thứ tư”, v.v., (nếu có) trong phần mô tả, các điểm yêu cầu bảo hộ và các hình vẽ kèm theo của sáng chế chỉ được sử dụng để phân biệt các đối tượng tương tự nhau chứ không nhất thiết được sử dụng để thể hiện bất kì mối quan hệ ưu tiên nào giữa các từ này.

Các phương án nêu trên chỉ là các phương án được ưu tiên của sáng chế chứ không nhằm giới hạn sáng chế. Các phương án thay thế, các phương án tương đương, các phương án cải biến, v.v., mà được tạo ra dựa trên ý tưởng và nguyên lý của sáng chế thì cũng nằm trong phạm vi bảo hộ của sáng chế.

YÊU CẦU BẢO HỘ

1. Phương pháp để cải thiện tốc độ chạy của ứng dụng bao gồm các bước:

chạy ứng dụng được lưu trong thiết bị đầu cuối;

thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang khi chạy ứng dụng, trong đó thông tin dữ liệu cục bộ này bao gồm tên của tệp tin mà trong đó dữ liệu cục bộ này được chứa, độ dịch của dữ liệu cục bộ trong tệp tin và kích thước của dữ liệu cục bộ; và

thu được dữ liệu cục bộ theo thông tin dữ liệu cục bộ, và nạp dữ liệu cục bộ sau khi nhận được lệnh cải thiện tốc độ chạy của ứng dụng;

trong đó việc thu được dữ liệu cục bộ theo thông tin dữ liệu cục bộ bao gồm các bước:

phân tích thông tin dữ liệu cục bộ để thu được tên của tệp tin mà trong đó dữ liệu cục bộ được chứa, độ dịch của dữ liệu cục bộ trong tệp tin và kích thước của dữ liệu cục bộ;

tập hợp thông tin dữ liệu cục bộ theo tên của tệp tin mà trong đó dữ liệu cục bộ được chứa;

hợp nhất thông tin dữ liệu cục bộ được tập hợp nếu khoảng cách giữa các độ dịch của thông tin dữ liệu cục bộ nằm trong khoảng định trước, và chọn thông tin dữ liệu cục bộ đáp ứng điều kiện nạp trước từ thông tin dữ liệu cục bộ được hợp nhất như thông tin dữ liệu nạp trước; và

thu được dữ liệu cục bộ theo thông tin dữ liệu nạp trước được chọn.

2. Phương pháp theo điểm 1, trong đó bước thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang khi chạy ứng dụng bao gồm các bước:

thu thập sự kiện thiếu trang cứng và sự kiện thiếu trang mềm trong khi ứng dụng đang chạy bằng cơ chế theo dõi sự kiện, và thu được thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy từ nhật ký theo dõi sự kiện ghi lại sự kiện thiếu trang cứng và sự kiện thiếu trang mềm;

hoặc, giám sát hệ thống đang được gọi bởi tiến trình HOOK, thu được mã giám sát để ghi lại thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang, và thu được, theo mã giám sát, thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy.

3. Phương pháp theo điểm 1, trong đó sau bước chọn thông tin dữ liệu cục bộ đáp ứng điều kiện nạp trước từ thông tin dữ liệu cục bộ được hợp nhất như thông tin dữ liệu nạp trước, thì phương pháp này còn bao gồm các bước:

lưu thông tin dữ liệu nạp trước được chọn theo thông tin về ứng dụng; và

việc nạp dữ liệu cục bộ thu được bao gồm bước:

thu được thông tin dữ liệu nạp trước theo thông tin về ứng dụng, thu được dữ liệu cục bộ theo thông tin dữ liệu nạp trước, và nạp dữ liệu cục bộ thu được vào bộ nhớ.

4. Phương pháp theo điểm từ 1, trong đó trước bước nạp dữ liệu cục bộ thu được, phương pháp này còn bao gồm các bước:

xuất ra danh sách ứng dụng, và yêu cầu người dùng chọn ứng dụng cần được cải thiện tốc độ chạy từ danh sách ứng dụng này; và

nhận, theo ứng dụng được chọn bởi người dùng từ danh sách ứng dụng, lệnh để cải thiện tốc độ chạy của ứng dụng và thông tin về ứng dụng.

5. Thiết bị đầu cuối bao gồm: một hoặc nhiều bộ xử lý, bộ nhớ, và một hoặc nhiều chương trình, trong đó một hoặc nhiều chương trình này được lưu trong bộ nhớ và

được thực hiện bởi một hoặc nhiều bộ xử lý này, và một hoặc nhiều chương trình này bao gồm lệnh để thực hiện các thao tác như sau:

thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang khi chạy ứng dụng, trong đó thông tin dữ liệu cục bộ này bao gồm tên của tệp tin mà trong đó dữ liệu cục bộ này được chứa, độ dịch của dữ liệu cục bộ này trong tệp tin và kích thước của dữ liệu cục bộ;

phân tích thông tin dữ liệu cục bộ để thu được tên của tệp tin mà trong đó dữ liệu cục bộ được chứa, độ dịch của dữ liệu cục bộ trong tệp tin và kích thước của dữ liệu cục bộ;

tập hợp thông tin dữ liệu cục bộ theo tên của tệp tin mà trong đó dữ liệu cục bộ được chứa;

hợp nhất thông tin dữ liệu cục bộ được tập hợp nếu khoảng cách giữa các độ dịch của thông tin dữ liệu cục bộ nằm trong khoảng định trước, và chọn thông tin dữ liệu cục bộ đáp ứng điều kiện nạp trước từ thông tin dữ liệu cục bộ được hợp nhất như thông tin dữ liệu định trước; và

thu được dữ liệu cục bộ tương ứng theo thông tin dữ liệu cục bộ, và nạp dữ liệu cục bộ sau khi nhận được lệnh cải thiện tốc độ chạy của ứng dụng.

6. Thiết bị đầu cuối theo điểm 5, trong đó một hoặc nhiều chương trình nêu trên bao gồm các lệnh để thực hiện các thao tác như sau:

thu thập sự kiện thiếu trang cứng và sự kiện thiếu trang mềm trong lúc ứng dụng đang chạy bằng cơ chế theo dõi sự kiện, và thu được thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy từ nhật ký theo dõi sự kiện ghi lại sự kiện thiếu trang cứng và sự kiện thiếu trang mềm;

hoặc, giám sát hệ thống đang được gọi bởi tiến trình HOOK, thu được mã giám

sát để ghi lại thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang, và thu được, theo mã giám sát, thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy.

7. Thiết bị đầu cuối theo điểm 5, trong đó một hoặc nhiều chương trình nêu trên bao gồm các lệnh để thực hiện các thao tác như sau:

lưu thông tin dữ liệu nạp trước được chọn theo thông tin về ứng dụng; và

việc nạp dữ liệu cục bộ thu được bao gồm bước:

thu được thông tin dữ liệu nạp trước theo thông tin về ứng dụng, thu được dữ liệu cục bộ tương ứng theo thông tin dữ liệu nạp trước, và nạp dữ liệu cục bộ thu được vào bộ nhớ.

8. Thiết bị đầu cuối theo điểm 5, trong đó một hoặc nhiều chương trình nêu trên bao gồm các lệnh để thực hiện các thao tác như sau:

xuất ra danh sách ứng dụng, và yêu cầu người dùng chọn ứng dụng cần được cải thiện tốc độ chạy từ danh sách ứng dụng này; và

nhận, theo ứng dụng được chọn bởi người dùng từ danh sách ứng dụng, lệnh để cải thiện tốc độ chạy của ứng dụng và thông tin về ứng dụng.

9. Vật lưu trữ bất khả biến đọc được bằng máy tính, trong đó vật lưu trữ đọc được bằng máy tính này lưu một hoặc nhiều chương trình, một hoặc nhiều chương trình này được thực hiện bởi một hoặc nhiều bộ xử lý để thực hiện phương pháp điều khiển, và phương pháp điều khiển này bao gồm các bước:

thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang khi chạy ứng dụng, trong đó thông tin dữ liệu cục bộ này bao gồm tên của tệp tin mà trong đó dữ liệu cục bộ này được chứa, độ dịch của dữ liệu cục bộ này trong tệp tin

và kích thước của dữ liệu cục bộ;

phân tích thông tin dữ liệu cục bộ để thu được tên của tệp tin mà trong đó dữ liệu cục bộ này được chứa, độ dịch của dữ liệu cục bộ trong tệp tin và kích thước của dữ liệu cục bộ;

tập hợp thông tin dữ liệu cục bộ theo tên của tệp tin mà trong đó dữ liệu cục bộ được chứa;

hợp nhất thông tin dữ liệu cục bộ được tập hợp nếu khoảng cách giữa các độ dịch của thông tin dữ liệu cục bộ nằm trong khoảng định trước, và chọn thông tin dữ liệu cục bộ đáp ứng điều kiện nạp trước từ thông tin dữ liệu cục bộ được hợp nhất như thông tin dữ liệu định trước; và

thu được dữ liệu cục bộ tương ứng theo thông tin dữ liệu cục bộ, và nạp dữ liệu cục bộ thu được sau khi nhận được lệnh cải thiện tốc độ chạy của ứng dụng.

10. Vật lưu trữ bất khả biến đọc được bằng máy tính theo điểm 9, trong đó việc thu thập thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy bao gồm các bước:

thu thập sự kiện thiếu trang cứng và sự kiện thiếu trang mềm trong lúc ứng dụng đang chạy bằng cơ chế theo dõi sự kiện, và thu được thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy từ nhật ký theo dõi sự kiện ghi lại sự kiện thiếu trang cứng và sự kiện thiếu trang mềm;

hoặc, giám sát hệ thống đang được gọi bởi tiến trình HOOK, thu được mã giám sát để ghi lại thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang, và thu được, theo mã giám sát, thông tin dữ liệu cục bộ tương ứng với sự gián đoạn do thiếu trang trong lúc ứng dụng đang chạy.

11. Vật lưu trữ bất khả biến đọc được bằng máy tính theo điểm 9, trong đó sau khi

việc chọn thông tin dữ liệu cục bộ đáp ứng điều kiện nạp trước từ thông tin dữ liệu cục bộ được hợp nhất như thông tin dữ liệu nạp trước, thì phương pháp nêu trên còn bao gồm các bước:

lưu thông tin dữ liệu nạp trước được chọn theo thông tin về ứng dụng; và

việc nạp dữ liệu cục bộ thu được bao gồm bước:

thu được thông tin dữ liệu nạp trước theo thông tin về ứng dụng, thu được dữ liệu cục bộ tương ứng theo thông tin dữ liệu nạp trước, và nạp dữ liệu cục bộ thu được vào bộ nhớ.

12. Vật lưu trữ bất khả biến đọc được bằng máy tính theo điểm 9, trong đó trước bước nạp dữ liệu cục bộ thu được, phương pháp nêu trên còn bao gồm các bước:

xuất ra danh sách ứng dụng, và yêu cầu người dùng chọn ứng dụng cần được cải thiện tốc độ chạy từ danh sách ứng dụng này; và

nhận, theo ứng dụng được chọn bởi người dùng từ danh sách ứng dụng, lệnh để cải thiện tốc độ chạy của ứng dụng và thông tin về ứng dụng.

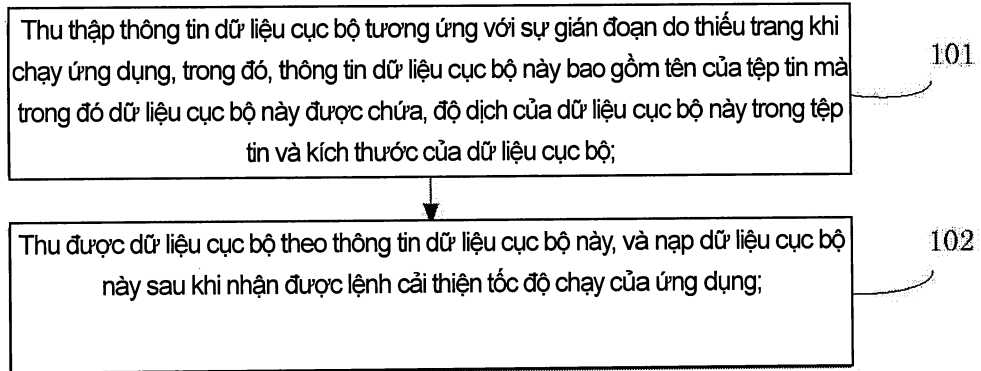


Fig.1

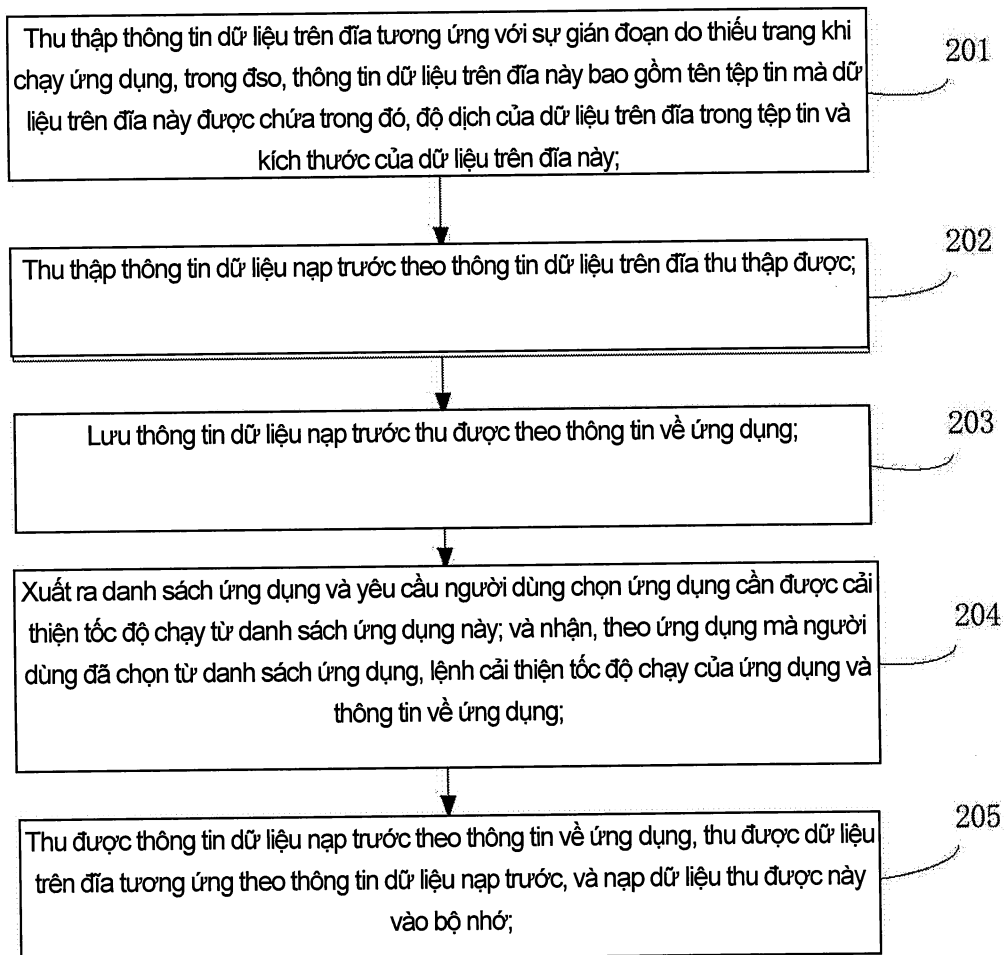
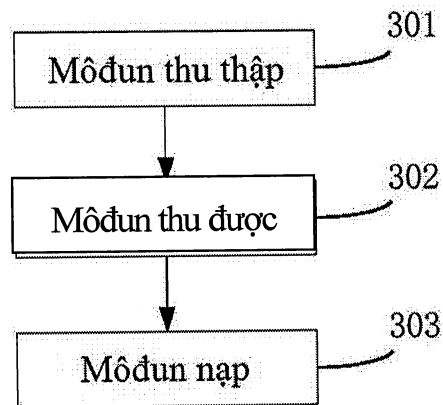
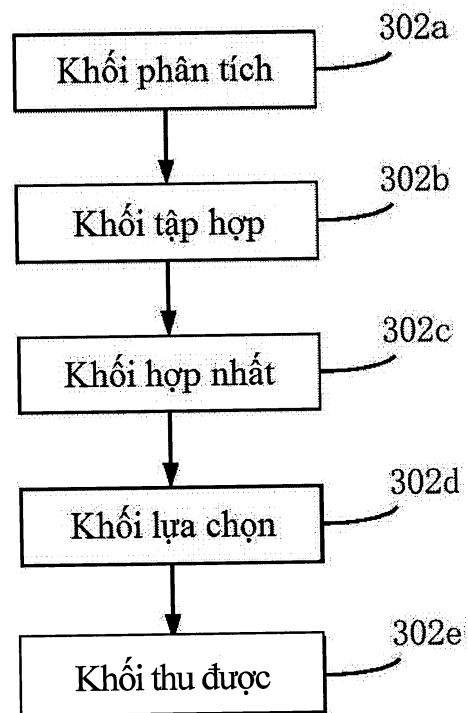
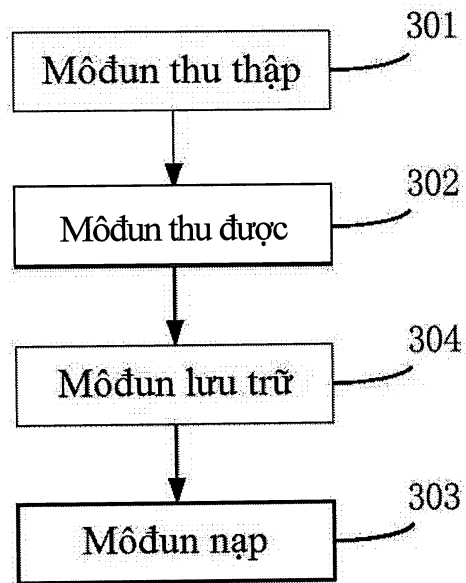
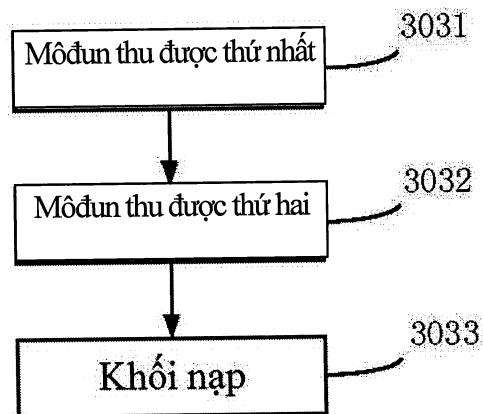
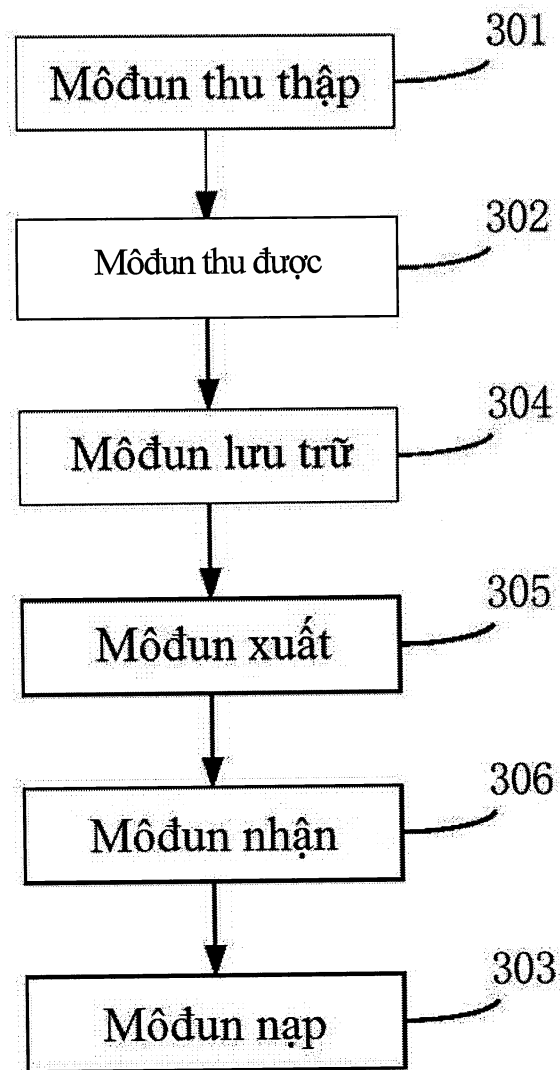
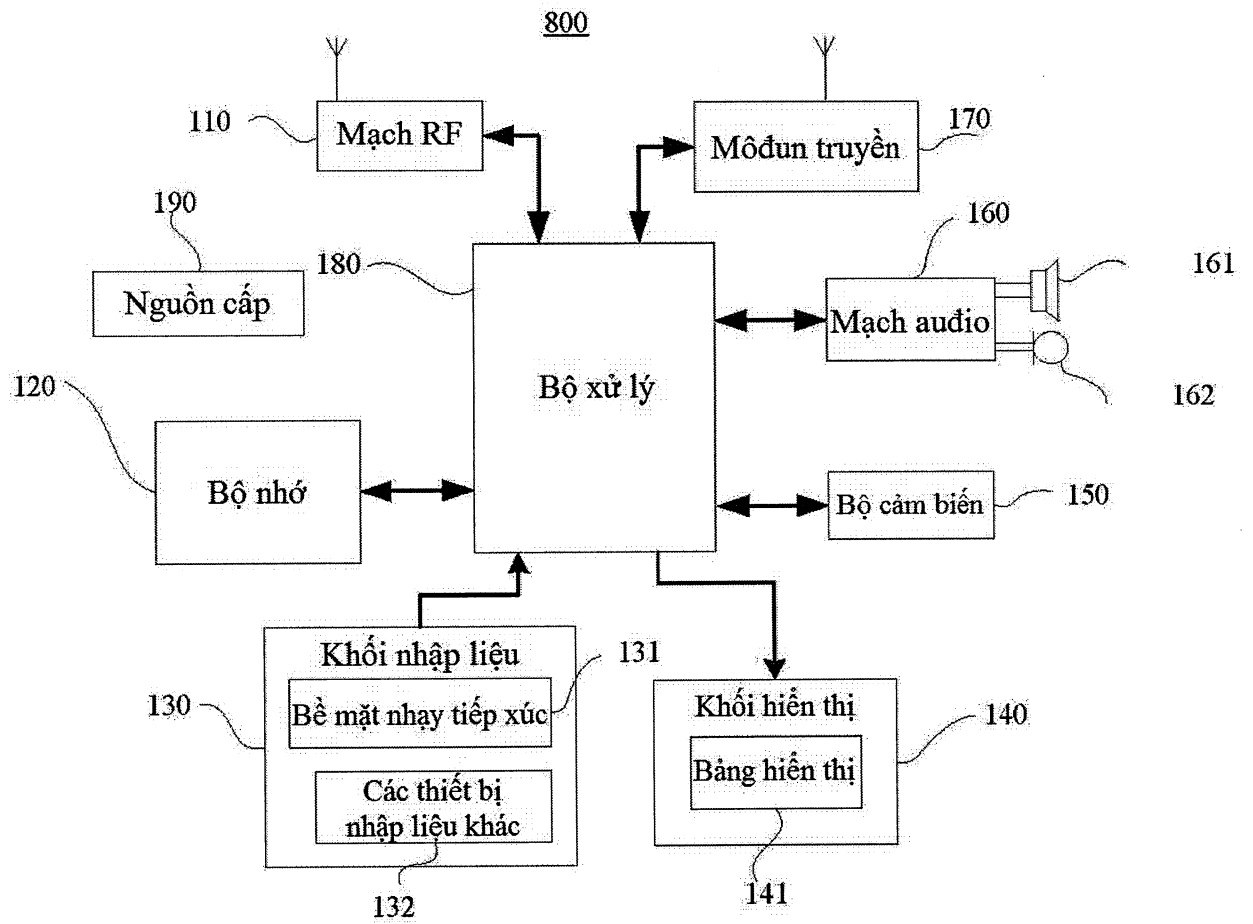


Fig.2

**Fig.3****Fig.4**

**Fig.5****Fig.6**

**Fig.7**

**Fig.8**