US 20090094378A1

(19) **United States**

(12) **Patent Application Publication**

(10) **Pub. No.: US 2009/0094378 A1**

(43) **Pub. Date: Apr. 9, 2009**

Reus et al.

(54) **SOFTWARE DEPLOYMENT USING CLIENT LOCATION**

(75) Inventors: **Edward Reus**, Woodinville, WA (US); **Jianbo Hou**, Issaquah, WA (US); **Jefferson Davis**, Sammamish, WA (US); **Jimin Li**, Bellevue, WA (US)

Correspondence Address:
MICROSOFT CORPORATION
ONE MICROSOFT WAY
REDMOND, WA 98052 (US)

(73) Assignee: **MICROSOFT CORPORATION**, Redmond, WA (US)

(21) Appl. No.: **11/869,104**

(22) Filed: **Oct. 9, 2007**

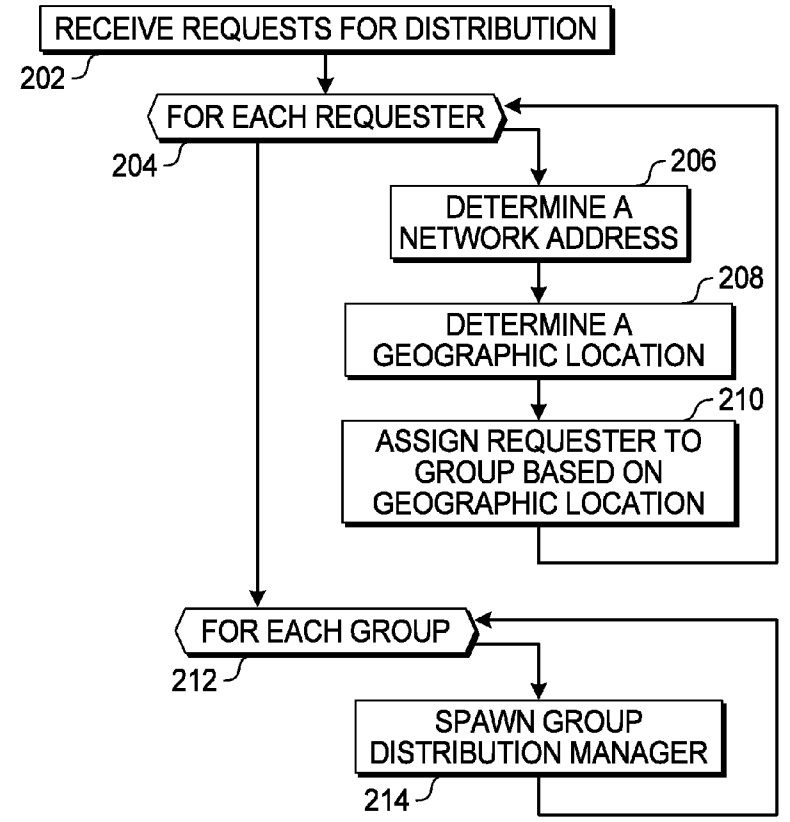
Publication Classification

(51) **Int. Cl.**
G06F 15/16 (2006.01)

(52) **U.S. Cl.** **709/235**

(57) **ABSTRACT**

A software distribution mechanism evaluates network addresses of requesting clients to determine a location for each client. The clients from a particular location are grouped together and a fraction of those clients in a particular group are recipients of a software distribution. The faction is adjusted to enable more or fewer clients to download, thus effectively throttling the amount of bandwidth consumed by a mass distribution event. The fraction may be adjusted for particular geographical locations and the time of day to make more effective use of network bandwidth.



200
METHOD FOR GROUPING SOFTWARE REQUESTERS

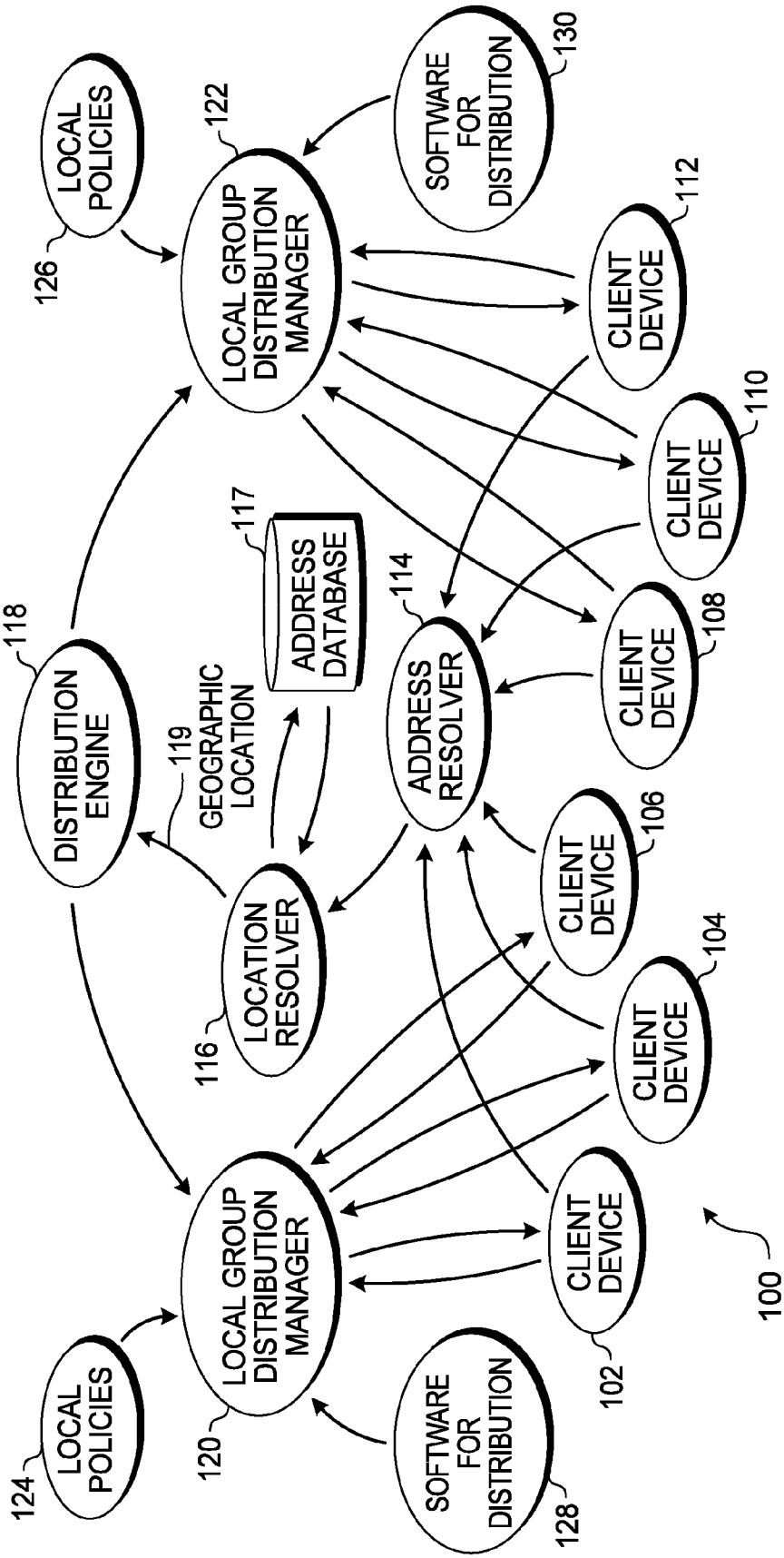

LOCAL POLICIES
126
LOCAL GROUP DISTRIBUTION MANAGER
122
SOFTWARE FOR DISTRIBUTION
130
CLIENT DEVICE
112
CLIENT DEVICE
110
CLIENT DEVICE
108
ADDRESS DATABASE
117
ADDRESS RESOLVER
114
DISTRIBUTION ENGINE
118
119
GEOGRAPHIC LOCATION
LOCATION RESOLVER
116
CLIENT DEVICE
106
CLIENT DEVICE
104
LOCAL GROUP DISTRIBUTION MANAGER
120
CLIENT DEVICE
102
LOCAL POLICIES
124
SOFTWARE FOR DISTRIBUTION
128
100
SYSTEM FOR SOFTWARE DISTRIBUTION

FIG. 1

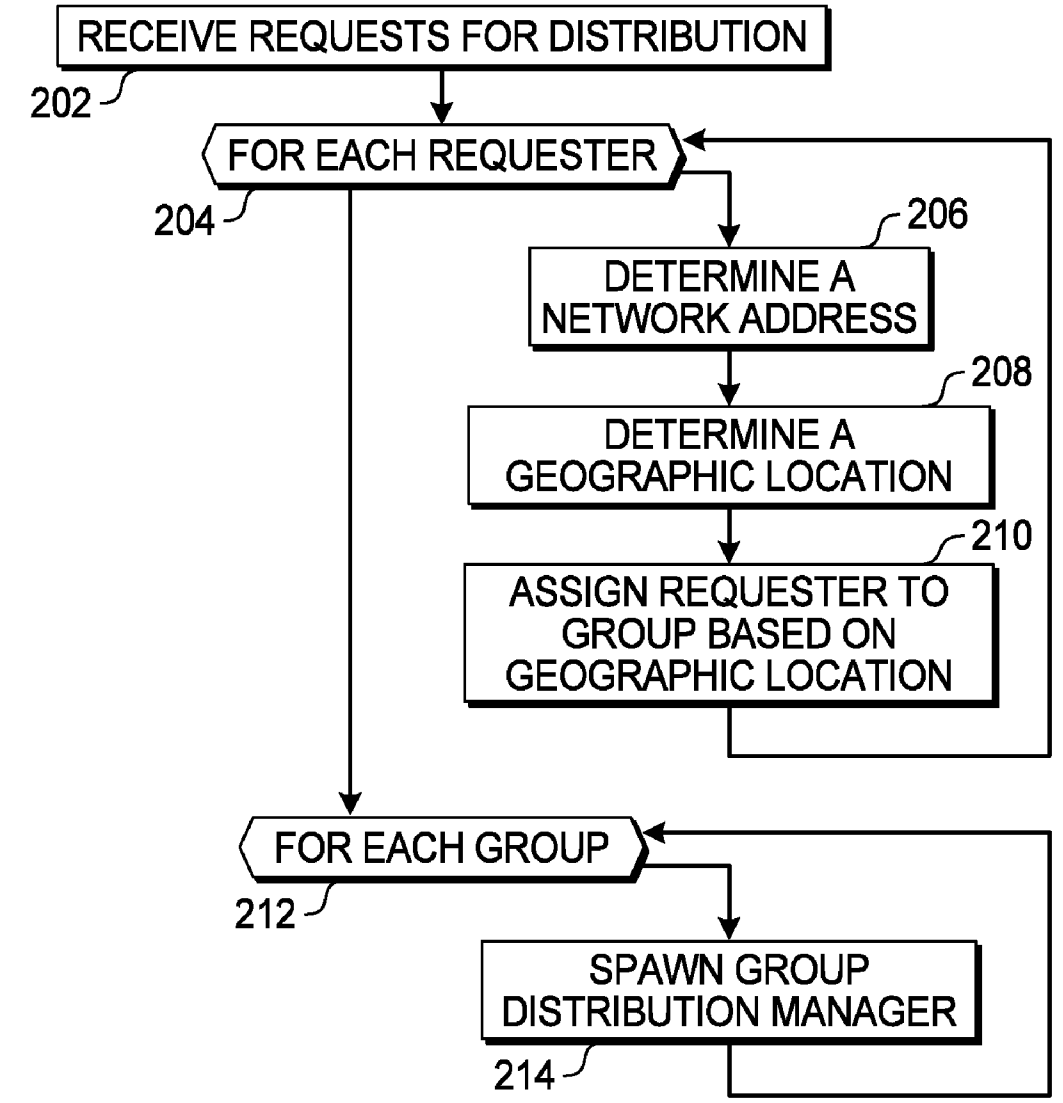

RECEIVE REQUESTS FOR DISTRIBUTION
202
FOR EACH REQUESTER
204
206
DETERMINE A NETWORK ADDRESS
208
DETERMINE A GEOGRAPHIC LOCATION
210
ASSIGN REQUESTER TO GROUP BASED ON GEOGRAPHIC LOCATION
FOR EACH GROUP
212
SPAWN GROUP DISTRIBUTION MANAGER
214
200
METHOD FOR GROUPING SOFTWARE REQUESTERS

FIG. 2

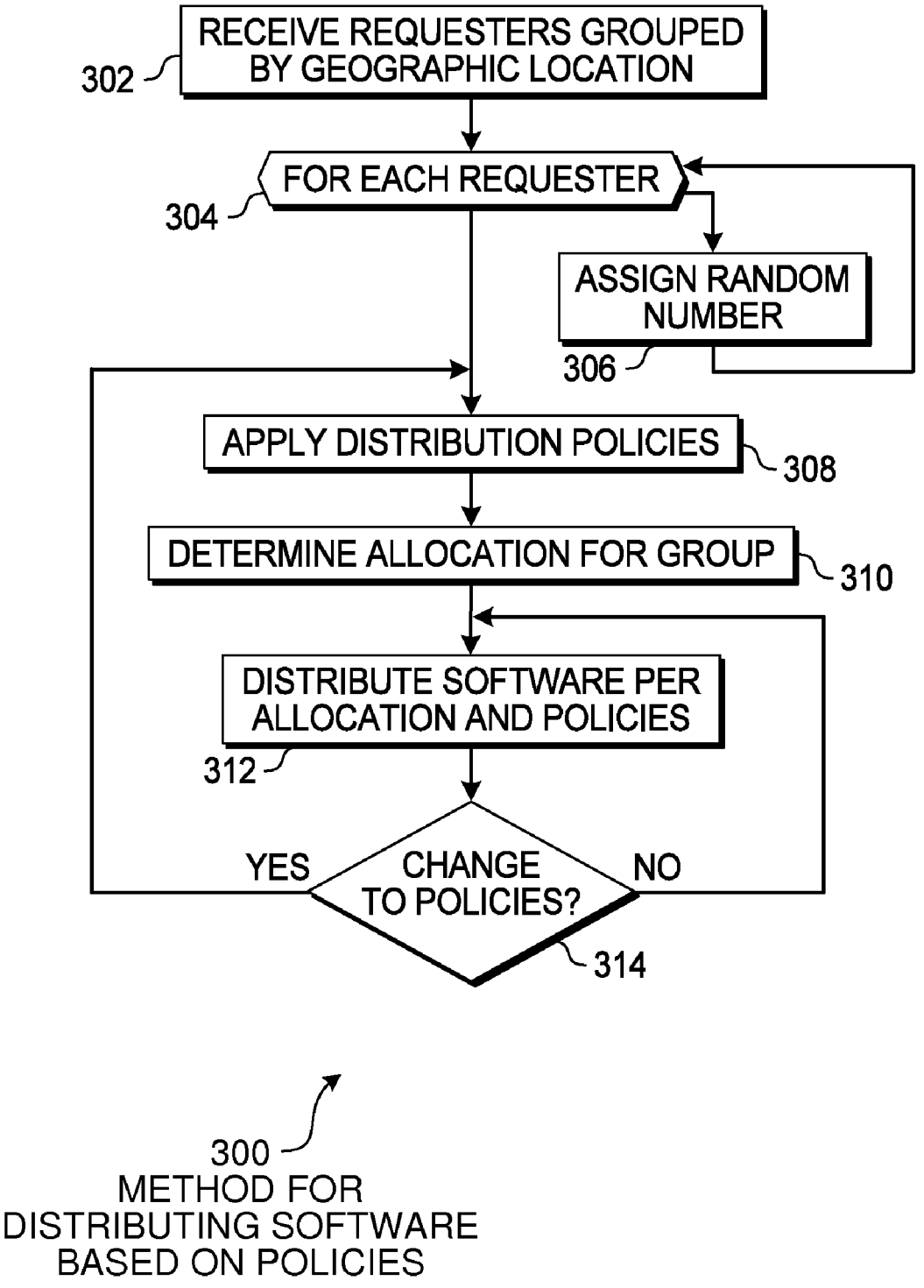

RECEIVE REQUESTERS GROUPED BY GEOGRAPHIC LOCATION
302
FOR EACH REQUESTER
304
ASSIGN RANDOM NUMBER
306
APPLY DISTRIBUTION POLICIES
308
DETERMINE ALLOCATION FOR GROUP
310
DISTRIBUTE SOFTWARE PER ALLOCATION AND POLICIES
312
YES
CHANGE TO POLICIES?
NO
314
300
METHOD FOR DISTRIBUTING SOFTWARE BASED ON POLICIES

FIG. 3

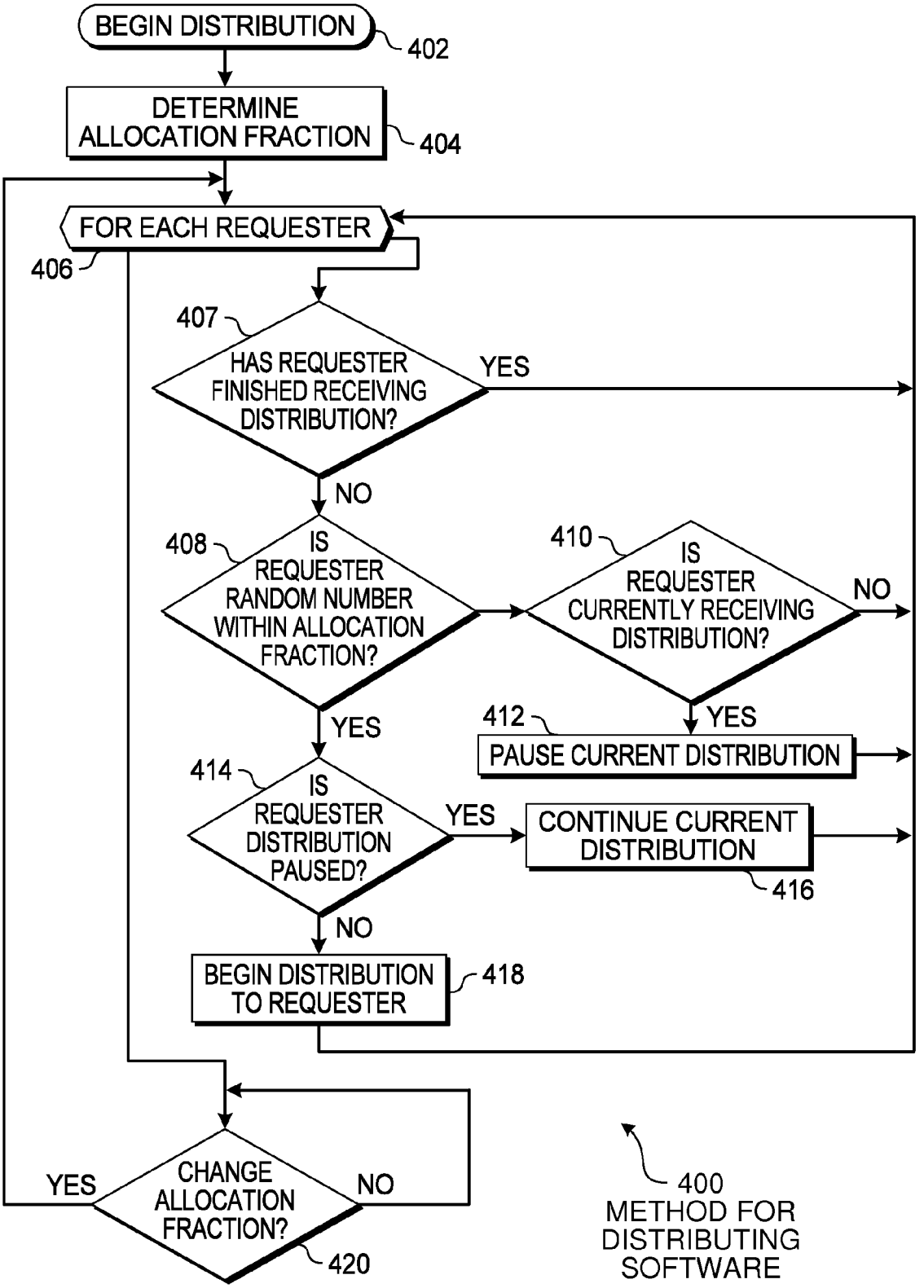

BEGIN DISTRIBUTION
402
DETERMINE ALLOCATION FRACTION
404
FOR EACH REQUESTER
406
407
HAS REQUESTER FINISHED RECEIVING DISTRIBUTION?
YES
NO
408
IS REQUESTER RANDOM NUMBER WITHIN ALLOCATION FRACTION?
410
IS REQUESTER CURRENTLY RECEIVING DISTRIBUTION?
NO
YES
412
PAUSE CURRENT DISTRIBUTION
414
IS REQUESTER DISTRIBUTION PAUSED?
YES
CONTINUE CURRENT DISTRIBUTION
416
NO
BEGIN DISTRIBUTION TO REQUESTER
418
CHANGE ALLOCATION FRACTION?
YES
NO
420
400
METHOD FOR DISTRIBUTING SOFTWARE

FIG. 4

SOFTWARE DEPLOYMENT USING CLIENT LOCATION

BACKGROUND

[0001] Distribution of software across the Internet can take a large toll on Internet bandwidth, especially when many clients download software at the same time. For very large distributions of software, such as updates to various applications and operating systems that may have hundreds of millions of client devices that each download many megabytes or even gigabytes of data, the sheer volume of downloaded data may use an extremely large amount of Internet bandwidth.

SUMMARY

[0002] A software distribution mechanism evaluates network addresses of requesting clients to determine a location for each client. The clients from a particular location are grouped together and a fraction of those clients in a particular group are recipients of a software distribution. The faction is adjusted to enable more or fewer clients to download, thus effectively throttling the amount of bandwidth consumed by a mass distribution event. The fraction may be adjusted for particular geographical locations and the time of day to make more effective use of network bandwidth.

[0003] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] In the drawings,

[0005] FIG. **1** is a diagram illustration of an embodiment showing a system for software distribution.

[0006] FIG. **2** is a flowchart illustration of an embodiment showing a method for grouping software requesters.

[0007] FIG. **3** is a flowchart illustration of an embodiment showing a method for distributing software based on policies.

[0008] FIG. **4** is a flowchart illustration of an embodiment showing a method for distributing software.

DETAILED DESCRIPTION

[0009] A software distribution system may resolve a geographic location from a network address and use the geographic location in metering or controlling software distributions to client devices. Each client or requester may be able to receive a software distribution based on policies applied to groups established for various geographic locations.

[0010] In an example use of an embodiment, a worldwide software distribution may be performed by grouping each client into groups defined by continents or time zones. During the course of a day, those clients in a nighttime location may be permitted to receive a software distribution while clients in a daytime location may be restricted from the distribution. Such an embodiment may make use of the fact that bandwidth is more available during nighttime than daytime.

[0011] Each group may have a mechanism for permitting distribution to a fraction of the client devices. One mechanism may be to assign a random number to each client device, producing a random distribution of devices. An allocation fraction may be set for the group that defines the fraction of devices that may receive the distribution. The allocation fraction may be used to select those devices with a random number below a certain number and commence the distribution. Over time, the allocation fraction may be increased so that each group member may eventually receive the distribution.

[0012] One of the benefits of a software distribution system is that network bandwidth and server capacity may not be overwhelmed by a large number of requests or large sized distributions. The allocation mechanism may be capable of increasing and decreasing the number of clients being serviced, enabling dynamic control over the distribution.

[0013] A software distribution may be any type of computer data that may be distributed to client devices. For example, operating system and application updates may be periodically distributed to client devices. In some instances, a software distribution may be in the form of data, while in other instances, a software distribution may be in the form of executable code. Some embodiments may have an executable installer on each client that accepts the distribution and installs the distribution on the client device.

[0014] The software distribution may be performed in many different manners. In some cases, a client device may download a distribution in a pull-type arrangement. In other cases, a server device may push a distribution to the client device. Other systems may use other mechanisms, protocols, and techniques for transferring a software distribution from servers to clients.

[0015] Throughout this specification, like reference numbers signify the same elements throughout the description of the figures.

[0016] When elements are referred to as being "connected" or "coupled," the elements can be directly connected or coupled together or one or more intervening elements may also be present. In contrast, when elements are referred to as being "directly connected" or "directly coupled," there are no intervening elements present.

[0017] The subject matter may be embodied as devices, systems, methods, and/or computer program products. Accordingly, some or all of the subject matter may be embodied in hardware and/or in software (including firmware, resident software, micro-code, state machines, gate arrays, etc.) Furthermore, the subject matter may take the form of a computer program product on a computer-usable or computer-readable storage medium having computer-usable or computer-readable program code embodied in the medium for use by or in connection with an instruction execution system. In the context of this document, a computer-usable or computer-readable medium may be any medium that can contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device.

[0018] The computer-usable or computer-readable medium may be, for example but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, device, or propagation medium. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media.

[0019] Computer storage media includes volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks

(DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can accessed by an instruction execution system. Note that the computer-usable or computer-readable medium could be paper or another suitable medium upon which the program is printed, as the program can be electronically captured, via, for instance, optical scanning of the paper or other medium, then compiled, interpreted, of otherwise processed in a suitable manner, if necessary, and then stored in a computer memory.

[0020] Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of the any of the above should also be included within the scope of computer readable media.

[0021] When the subject matter is embodied in the general context of computer-executable instructions, the embodiment may comprise program modules, executed by one or more systems, computers, or other devices. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Typically, the functionality of the program modules may be combined or distributed as desired in various embodiments.

[0022] FIG. **1** is a diagram of an embodiment **100** showing a system for software distribution. Embodiment **100** is a functional illustration of various components that may make up a software distribution system. In some embodiments, several of the functional aspects illustrated may be performed on a single device having a processor and adapted to perform each function. In other embodiments, some of the individual functional aspects may be performed by different devices. Some such embodiments may have several devices that are used to perform one specific function or a subset of a function. For example, a cluster of server devices may be used to perform one specific function.

[0023] The diagram of FIG. **1** illustrates functional components of a system. In some cases, the component may be a hardware component, a software component, or a combination of hardware and software. Some of the components may be application level software, while other components may be operating system level components. In some cases, the connection of one component to another may be a close connection where two or more components are operating on a single hardware platform. In other cases, the connections may be made over network connections spanning long distances. Each embodiment may use different hardware, software, and interconnection architectures to achieve the various components described.

[0024] Embodiment **100** illustrates a system for software distribution. After an initial connection, a client's address may be used to determine a geographic location for the client. Based on the geographic location, the client may be assigned to various local group distribution managers that may be servers which perform the software distribution transfer to the client devices.

[0025] Multiple client devices **102**, **104**, **106**, **108**, **110**, and **112** may connect to an address resolver **114** during an initial step of receiving a software distribution. The address resolver **114** may determine a network address from the client. In many cases, the network address may be an Internet Protocol or IP address.

[0026] The location resolver **116** may use an address database **117** to determine a geographic location **119** of the individual clients. Various embodiments may be able to resolve a client's geographic location with different resolutions. In some cases, the location resolver **116** may be able to resolve a location down to a country or broad geographic region. In other cases, the location resolver **116** may be able to resolve the location down to a state, city, or municipality. In still other cases, the location resolver **116** may be able to identify a client's internet service provider and an approximate street address.

[0027] Various mechanisms may be used to resolve a location from an address. In cases where a gross location may be used, such as determining a continent on which an address may be located, the address database **117** may be compressed or summarized so that fast and efficient lookups may be performed. When searching for more detailed location information, a more detailed address database **117** may be used.

[0028] Some embodiments may be used for metering bandwidth and server capacity for a distribution server. In such embodiments, the location determination may be approximated with enough accuracy so that the distribution server is not overloaded with requests or so that bandwidth is consumed in a controlled manner.

[0029] For example, a server located in North America may be configured to perform software distribution during periods of low network traffic, such as between midnight and 5:00 am Eastern time. When a request from a client device is received, the location may be determined to be somewhere within North America and the request may be routed to a distribution server within North America.

[0030] In some instances, the location resolver **116** may incorrectly determine that a client device is within North America when the client device is actually in Europe or some other geographic region. For those devices with incorrectly determined location, the distributed software may be routed from the North American server and reach the device across an intercontinental connection. In another instance, an IP subnet may fall across states or even country borders. When the location resolver **116** determines that an address is within the subnet, the location resolver **116** may or may not be able to determine on which side of the border the actual device resides. In many embodiments, the inaccuracy of location detection may be offset by the speed in which location may be determined.

[0031] Other embodiments may use location determination to tailor the content of the software distribution. In such embodiments, the accuracy of location determination may be of higher importance and more complex and detailed location searches may be performed. Such an embodiment may download specific content to devices within a precise geographical region. For example, a governmental regulation may limit distribution of specific content to a specific geographical region. An example of such content may be encryption or security algorithms that are regulated against transmitting to

specific foreign countries. Another example may be language or location specific software, where the distribution to other areas may be confusing or otherwise undesirable.

[0032] Some embodiments may use multiple databases for resolving a location for an address. A first database may be used to locate an approximate location, such as a continent, country, or state. A second group of databases may be used to locate a more precise location where the second databases may be tailored for the specific continent, country, or states.

[0033] In some embodiments, the network address may also indicate a specific internet service provider. Based on the unique characteristics of the internet service provider, a software distribution may be packaged or transmitted in a specific way that may optimize the characteristics of the specific internet service provider. In some cases, the network address may include knowledge about the speed of the connection, such as a low speed dial-in connection, a high bandwidth fiber optic connection, a wireless connection, cable modem, digital subscriber line (DSL) connection, or other type of connection. Such knowledge may be used to estimate the data transmission time and approximate the bandwidth requirements of portions of a network.

[0034] Based on the geographic location **119**, a distribution engine **118** may assign the client to one of several local group distribution managers **120** or **122**. The local group distribution managers **120** and **122** may coordinate software distribution to client devices within specific geographic groups. For example, local group distribution manager **120** may service client devices **102**, **104**, and **106** while local group distribution manager **122** may service client devices **108**, **110**, and **112**.

[0035] Each local group distribution manager **120** and **122** may handle the distribution of software to the assigned client devices. In many embodiments, a local group distribution manager **120** or **122** may be physically located within or near the geographic location of the various clients assigned to the specific software distribution manager. In some embodiments, the group distribution managers **120** and **122** may be collocated yet may separately service devices in different geographic regions.

[0036] In some embodiments, the functions of local group distribution managers **120** and **122** may be performed by one server device or by many devices. In some embodiments, a single server device may act as a local group distribution manager for two or more different groups of clients. In other devices, clusters of servers may separately act as group distribution managers to various groups. In some embodiments, a single server may be located in a sparsely populated geographic region and perform local group distribution services, while a cluster of servers may service a more densely populated region

[0037] Each local group distribution manager **120** and **122** may have a set of local policies **124** and **126** that may define the operation of each group distribution manager. The policies **124** and **126** may define any type of behavior of the local group distribution manager. In an example, the policies **124** and **126** may define the maximum bandwidth or number of connections for a server, which may also be defined with respect to time. The bandwidth or connection limitations may vary for the time of day, day of the week, or other configurations. Such a limitation may enable more connections and transfers during certain periods of time and may restrict or pause transfers during other periods.

[0038] Some policies may define a handshaking or protocol used by each local group distribution manager. One group of client devices in one locale may use one type of communication protocol while another may use a different type. Such protocol differences may include any type of difference in setting up, operating, and tearing down a communication session.

[0039] Some embodiments may be able to group certain client devices within a geographical location and within a specific type of network. For example, an internet service provider may have a block of network addresses that are used to connect many devices along a high bandwidth connection, such as a cable television connection or digital subscriber line connection. Some groups of addresses may be assigned to a particular enterprise or company. When client devices may be identified with such precision, some local group distribution managers may be adapted to apply specific policies to groups of client devices from specific internet service providers. Such a policy may define specific handshaking or communication protocols that may be used to transfer the software distribution, or may have different provisions that take advantage of features on the local network.

[0040] Some such policies may tailor the timing or bandwidth restrictions to the group of client devices to correspond with unique characteristics for the internet service provider. For example, the bandwidth availability for cable television connections may be restricted in the evening and late evening hours but less restricted during the late night and daytime hours. An internet service provider that has largely commercial clients may have restricted bandwidth during business hours but unrestricted bandwidth in the evening and nighttime hours.

[0041] The software for distribution **128** and **130** may be used by the local group distribution managers **120** and **122** to distribute to the various clients. The software may be customized or localized by the local group distribution managers **120** and **122**. In some cases, the software for distribution **128** and **130** may include different versions of distributed software and the local group distribution managers **120** and **122** may select an appropriate version to distribute to a particular client device or group of client devices. In some cases, the local group distribution managers **120** and **122** may package the distributed software in a specific form by zipping, encrypting, or compressing various files together or by organizing the software in a particular form. In some cases, some clients or groups of clients may receive a subset of the software for distribution **128** and **130**.

[0042] The various clients may receive distributed software using various mechanisms. In some embodiments, the clients may connect to the local group distribution manager or some other server and receive a list of available software distributions. The client may then contact the local group distribution manager **120** or **122** and request a download of all or a portion of the available software.

[0043] In other embodiments, the local group distribution managers **120** and **122** may initiate a communication session with the various client devices and transfer the distributed software using a push-type distribution mechanism.

[0044] FIG. **2** is a flowchart illustration of an embodiment **200** showing a method for grouping software requesters. Embodiment **200** is an example of a method or sequence for processing a distribution request and grouping the requester. Embodiment **200** is merely one example of such a method. Other embodiments may use additional steps or may consoli-

date steps together, and may use different terminology or nomenclature to perform the functional steps described herein.

[0045] Requests for distribution may be received in block **202**. In some cases, the requests may be sent by a device as part of a subscription service for software updates, as part of a registration and activation procedure performed during the activation of new software, or for some other procedure or reason.

[0046] For each requester in block **204**, a network address may be determined in block **206** from which a geographical location may be determined in block **208**. Based on the geographical location, the requester may be assigned a group in block **210**.

[0047] The network address of block **206** may be an Internet Protocol or IP address. Some other address mechanism may be used in certain cases, such as an address for a connection point used by the client device.

[0048] For example, a wireless device may have the ability to connect to the Internet through many different wireless connection points. For example, a portable device may connect through a wireless connection point in a coffee shop or airport. In a communication with a group distribution manager, such a device may connect using the network address of a connection point and may not have a dedicated network address assigned to the device. In such a case, the network address of the connection point may be used to determine a geographical location.

[0049] The groups defined in block **210** may correspond with the resolution or granularity by which a location may be determined in block **208**. Some embodiments may determine a location that is relatively precise, such as the street address of a device. Some embodiments may determine that a device is within a specific county, city, state, or country.

[0050] When determining a location based on a network address, a database of network addresses may be queried. In embodiments where an approximate location is used, such as a state or country, such databases may be compressed or summarized so that the queries may be performed quickly. In such cases, some accuracy may be lost during the summarization or compression and some addresses may not be correctly mapped to their actual geographic location.

[0051] After the requesters are processed in block **204**, for each group in block **212**, a group distribution manager may be spawned in block **214**.

[0052] The group distribution manager of block **214** may communicate with the various client devices and transfer software to the client devices using various mechanisms. In some instances, the group distribution manager may be one of many processes on a distribution server that distributes software to a group of client devices.

[0053] In many embodiments, the process of receiving requests in block **202** and processing each requester in blocks **204-210** may be performed by a central server. Such a central server may be located in a company headquarters or some other centralized location and accessed by client devices from many different locations. After processing, further communications may be performed by the group distribution manager server that may be located within or near the geographical area of the assigned client devices.

[0054] FIG. **3** is a flowchart illustration of an embodiment **300** showing a method for distributing software based on policies. Embodiment **300** is an example of method for distributing software to a variety of requesters using a random allocation. Embodiment **300** is merely one example of such a method. Other embodiments may use additional steps or may consolidate steps together, and may use different terminology or nomenclature to perform the functional steps described herein.

[0055] The requesters for a particular geographical group may be received in block **302**. For each requester in block **304**, a random number may be assigned in block **306**.

[0056] The random number in block **306** may be used to allocate bandwidth, communication sessions, or other factors to specific client devices and not to other client devices. In an example, each requester may be assigned an evenly distributed random number between 0 and 1. For example, in order to limit total bandwidth utilization, those requester client devices with random numbers less than 0.15 may be permitted to receive a distribution.

[0057] Each embodiment may have a different mechanism for allocating distributions, and the embodiment **300** is illustrated here as an example.

[0058] Distribution policies may be applied in block **308**. In some embodiments, the distribution policies may define the timing, configuration, or other factors associated with the software distribution. In many embodiments, larger volumes of distributions may be performed at certain hours of the day and certain days of the week. In many embodiments, a limited amount of distribution may be performed during certain periods of time, such as daytime or early evening when bandwidth usage is high. In some embodiments, no distribution may be performed during such periods.

[0059] The distribution policies of block **308** may include configuration parameters for the entire group or subgroups of client devices. Such configuration parameters may include a predefined handshaking or protocol, the manner in which software may be packaged or formatted, or various options that may be specified for each client device.

[0060] The allocation for the group may be determined in block **310**. The allocation may refer to a certain percentage or fraction of client devices that may be permitted to receive a distribution. In large distributions, such as operating system updates that may be received by tens or hundreds of millions of devices and may be very large distribution packages, the distributions may take place over several days, weeks, or even months, with a small fraction of devices being able to receive a distribution at a given period of time.

[0061] The software may be distributed per the allocation and policies in block **312**. In many embodiments, a client device may request a distribution package and may download the package. In some cases, the download process may take several minutes or several hours or more, depending on the connection and size of the distribution. In other cases, the distribution server may contact the requester client device and transmit the distribution using a push-type distribution mechanism.

[0062] If there is a change to the policies in block **314**, the process may return to block **308**. In some instances, the policies may change over time. For example, during a period of high available bandwidth, a large number of client devices may begin receiving a distribution package. Even though many requesters may be in the process of receiving the distribution package, the policy may change to pause or halt ongoing distributions. In such a case, the distribution polices may be re-applied in block **308**, a new allocation determined in block **310**, and the software may be distributed according to the polices in block **312**.

[0063] FIG. **4** is a flowchart diagram of an embodiment **400** showing a method for distributing software. Embodiment **400** is an example of a method by which an allocation fraction may be increased or decreased and more or fewer requester or client devices may receive a software distribution. Embodiment **400** is merely one example of such a method. Other embodiments may use additional steps or may consolidate steps together, and may use different terminology or nomenclature to perform the functional steps described herein.

[0064] The distribution may begin in block **402** and the allocation fraction determined in block **404**. In the initial phase of a distribution, the allocation fraction may be small and as client devices may complete the software distribution, the allocation fraction may be raised over time. In some embodiments, the allocation fraction may be raised or lowered depending on the time of day or other factors.

[0065] For each requester in block **406**, if the requester has finished receiving the distribution in block **407**, the requester is left unchanged. If the requester's random number is not within the allocation fraction in block **408**, and the requester is not currently receiving a distribution in block **410**, the requester is left unchanged.

[0066] For the purposes of this specification and claims, the terms requester and client device are used interchangeably.

[0067] If the requester is currently receiving the distribution in block **410** but not within the allocation fraction in block **408**, the current distribution may be paused in block **412**. Such a situation may occur if an allocation fraction is changed from a large fraction to a small fraction, where the fraction is defined as the portion of requesters or client devices that may be permitted to receive the software distribution.

[0068] If the requester is within the allocation fraction in block **408** and the requester's distribution is currently paused in block **414**, the distribution is continued in block **416**. Such a situation may occur when an allocation fraction is was lowered, causing the client device to have the distribution paused, then the allocation fraction may be raised, allowing the distribution to continue.

[0069] If the requester is within the allocation fraction in block **408** and the requester's distribution is not paused in block **414**, the distribution is started in block **418**.

[0070] After each requester is processed in block **406**, if a change in the allocation fraction is made in block **420**, each requester is evaluated again starting in block **406**.

[0071] The embodiment **400** is an example of a method by which software distributions may be started or paused.

[0072] In a typical use scenario, a software distribution may begin with a small fraction of client devices being able to receive the distribution. Over time, those client devices may complete the distribution and the fraction may be periodically raised higher until all of the requesters are able to receive the software distribution. During the distribution period, the allocation may be raised and lowered to respond to time policies, bandwidth restrictions, or other factors.

[0073] The foregoing description of the subject matter has been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the subject matter to the precise form disclosed, and other modifications and variations may be possible in light of the above teachings. The embodiment was chosen and described in order to best explain the principles of the invention and its practical application to thereby enable others skilled in the art to best utilize the invention in various embodiments and various modifications as are suited to the particular use contemplated. It is intended that the appended claims be construed to include other alternative embodiments except insofar as limited by the prior art.

What is claimed is:

1. A method comprising:
- determining a network address for a plurality of client devices to receive a software distribution;
- determining a geographical location for each of said client devices based on said network address;
- grouping said client devices into a first group and a second group based on said geographical location; and
- transferring said software distribution to a first fraction of said first group and a second fraction of said second group.

2. The method of claim **1**, said transferring said software distribution comprising pushing said software distribution to said first fraction and said second fraction.

3. The method of claim **1**, said transferring said software distribution comprising permitting said first fraction and said second fraction to download said software distribution.

4. The method of claim **1** further comprising assigning said client devices to a distribution manager.

5. The method of claim **4**, said distribution manager being adapted to:
- determining a random number for each of said client devices; and
- allocating each of said client devices based on said random number.

6. The method of claim **1** further comprising:
- adjusting said first fraction to transfer said software distribution to a larger number of said first group of said client devices.

7. The method of claim **1** further comprising:
- adjusting said first fraction to transfer said software distribution to a smaller number of said first group of said client devices.

8. The method of claim **7** further comprising:
- halting an ongoing transfer for at least a portion of said first fraction of said first group of said client devices.

9. The method of claim **7**, said adjusting being based at least on a time of day for said geographical location for said first group of said client devices.

10. The method of claim **1**, said software distribution comprising at least one of a group composed of:
- an operating system update;
- an application update; and
- a data file.

11. A computer readable medium comprising computer executable code adapted to perform the method of claim **1**.

12. A system comprising:
- a connection to a network;
- an address resolver adapted to determine a network address for each of a plurality of clients;
- a location resolver adapted to determine a geographical location for each of said plurality of clients based on said network address and group said plurality of clients into a first group and a second group based on said geographical location;
- a distribution engine adapted to transfer a software distribution to a first fraction of said first group and a second fraction of said second group.

13. The system of claim **12**, said distribution engine further adapted to transfer said software distribution by a method

comprising pushing said software distribution to said first fraction and said second fraction.

14. The system of claim **12**, said distribution engine further adapted to transfer said software distribution by a method comprising permitting said software distribution to be downloaded by said first fraction and said second fraction.

15. The system of claim **12**, said distribution engine further adapted to:

adjust said first fraction to transfer said software distribution to a larger number of said first group of said client devices.

16. The system of claim **12**, said distribution engine further adapted to:

adjust said first fraction to transfer said software distribution to a smaller number of said first group of said client devices.

17. The system of claim **16**, said distribution engine further adapted to:

halt an ongoing transfer for at least a portion of said first fraction of said first group of said client devices.

18. A method comprising:

determining a network address for a plurality of client devices to receive a software distribution;

determining a geographical location for each of said client devices based on said network address;

grouping said client devices into a first group and a second group based on said geographical location; and

transferring a first version of said software distribution to a first fraction of said first group and a second version of said software distribution to a second fraction of said second group.

19. The method of claim **18**, said transferring said first version of said software distribution comprising permitting said first fraction and said second fraction to download.

20. A computer readable medium comprising computer executable instructions adapted to perform the method of claim **18**.

* * * * *