



(51) International Patent Classification:

G06F 11/30 (2006.01) H04W 24/00 (2009.01)
G06F 11/34 (2006.01) G06F 11/36 (2006.01)

(21) International Application Number:

PCT/MY2014/000147

(22) International Filing Date:

3 June 2014 (03.06.2014)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

PI2013004361 3 December 2013 (03.12.2013) MY

(71) Applicant: MIMOS BERHAD [MY/MY]; Technology
Park Malaysia, 57000 Kuala Lumpur (MY).

(72) Inventors: FAHRURAZI, Fairul Rizal Bin; c/o Mimos
Berhad, Technology Park Malaysia, 57000 Kuala Lumpur
(MY). LOO, Fook Ann; c/o Mimos Berhad, Technology
Park Malaysia, 57000 Kuala Lumpur (MY).

(74) Agent: MIRANDAH, Patrick; Mirandah Asia (Malaysia)
sdn bhd, Suite 3B-19-3, Plaza Sentral, Jalan Stesen Sentral
5, 50470 Kuala Lumpur (MY).

(81) Designated States (unless otherwise indicated, for every
kind of national protection available): AE, AG, AL, AM,

AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,
BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM,
DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR,
KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME,
MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ,
OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA,
SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM,
TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM,
ZW.

(84) Designated States (unless otherwise indicated, for every
kind of regional protection available): ARIPO (BW, GH,
GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ,
UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ,
TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK,
EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV,
MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM,
TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW,
KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to applicant's entitlement to apply for and be granted a
patent (Rule 4.17(ii))
- of inventorship (Rule 4.17(iv))

[Continued on next page]

(54) Title: A SYSTEM AND METHOD FOR EMULATING MULTIPLE INDEPENDENT WIRELESS CLIENT DEVICES IN THE CLOUD

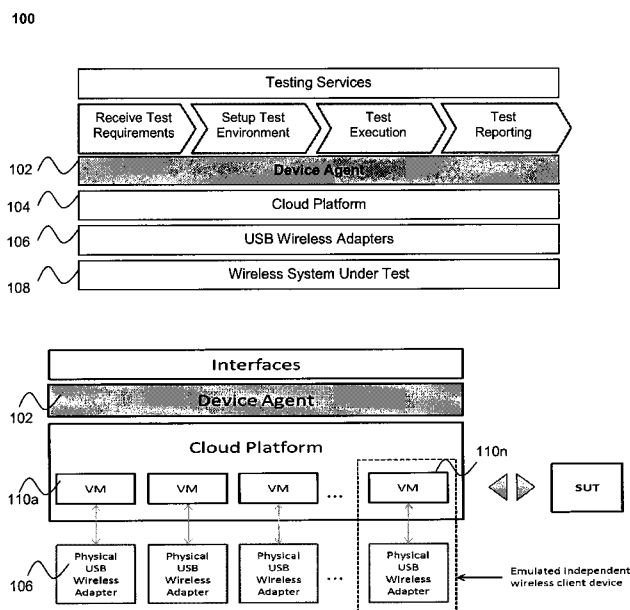


FIG. 1.0

(57) Abstract: Wireless network performance test is initiated and test results are gathered from multiple independent wireless client devices which simplify and assist in the process of analysis and generation of test reports. The system for emulating multiple independent wireless client device in the cloud comprising a cloud platform (104) consisting a plurality of virtual machines (110a-110n); a plurality of USB wireless adapters (106); and a device agent (102). The method for emulating multiple independent wireless client devices in the cloud comprising steps of setting up test environment by user (202); reading files by Device Agent and providing list of USB devices in cloud platform which can be used (204); specifying number of wireless client to be created by user (206); launching different virtual machines in the cloud platform by the Device Agent by using virtual machine images provided by the user based on settings in the configuration files (208); mapping and attaching physical USB devices to virtual machines in the cloud platform to create multiple independent wireless client device by the Device Agent (210); injecting test automation scripts into all virtual machines by the Device Agent and initiating test execution in all virtual machine in the cloud platform based on settings in the configuration files (212); and gathering results from all virtual machines by the Device Agent upon completion of test execution based on the settings in the configuration files (214); and analyzing test results received

from all virtual machines and generating test report (216).

WO 2015/084140 A1



Published:

— *with international search report (Art. 21(3))*

- 1 -

A SYSTEM AND METHOD FOR EMULATING MULTIPLE INDEPENDENT WIRELESS CLIENT DEVICES IN THE CLOUD

5 FIELD OF INVENTION

The present invention relates to a system and method for emulating multiple independent wireless client devices in the cloud. In particular, the invention initiates wireless network performance test and gathers test results from multiple independent wireless client devices which simplifies and assists in the process of analyzing and reporting test results received from multiple independent wireless client devices.

BACKGROUND ART

15 Currently available wireless network performance testing is expensive and test environment used in the performance tests need to be as close as possible to the real environment in order to generate a meaningful result. Further, the testing environment requires multiple wireless client devices to be set up such as computers and personnel are required during the test to generate user load that closely resembles the real environment (e.g. if we need to generate a user load of 100 wireless client devices, it is a waste of resources to call up 100 computers and personnel). The drawback of the current available situation is resolved in the present invention which proposes a system and method for emulating multiple independent wireless client devices in the cloud that can be used for wireless network testing that are able to reduce time and cost.

25 One example which provides a method and system for emulating a user population for wireless local area network access point performance, optimization and configuration testing is proposed in United States Patent No. US 7,277,395 B2 entitled "Method and Apparatus for Wireless Network Load Emulation" (hereinafter referred to as the US 395 Patent). In the US 395 Patent, all emulated client devices is provided in a load emulator which shares the same wireless interface in the Input Output Controller to connect to the SUT. In contrast, in the present invention, each emulated wireless client device is attached to one physical USB wireless adapter. All emulated client devices have exclusive access to its own USB wireless adapter to connect to the SUT. Further, the invention as disclosed in the US 395 Patent does not emulate independent wireless

- 2 -

client devices as compared to the present invention which provides emulation of independent wireless client devices in the cloud.. All emulated wireless client devices in the invention as disclosed in the US 395 Patent shares the same physical wireless interface and sends network traffic to and from the SUT in Round-Robin fashion as compared to the present invention in which all emulated wireless client devices are able to use its own USB adapter to send network traffic to and from the SUT in parallel (i.e. concurrent).

Another example that provides a system that introduces virtual Ethernet and virtual Wireless Ethernet interfaces for client using a single host device is disclosed in United States Patent Publication No. US 20030202486 A1 entitled "Method and System for Simulating Multiple Independent Client Devices in a Wired or Wireless Network" (hereinafter referred to as the US 486 Publication). In the invention as disclosed in US 486 Publication, all emulated client devices shares two physical wireless network cards in the host machine to connect to the SUT as compared to the present invention wherein each emulated wireless client device is attached to one physical USB wireless adapter. Further, in the US 486 Publication, only one host system is utilized as compared to in the present invention which utilizes cloud platform and USB adapter to emulate multiple independent wireless client devices. The invention as disclosed in the US 486 Publication is not scalable as it utilizes only one host system as compared to the present invention which is scalable (i.e. by adding more commodity hardware if required).

Another mechanism which provides emulation of wireless application through a scalable network architecture is disclosed in an IEEE paper entitled "Scalable Emulation of Dynamic Multi-Hop Topologies" by Anders Nickelsen, Morten N. Jensen, Erling V. Matthiesen, Hans-Peter Schwefel; IEEE, 2008. In the said paper, emulation testbed is provided for wireless application wherein wireless link is replaced with wired links and a topology emulator is provided in which all end-nodes connected to a central network switch by wired links is provided. In contrast, in the present invention, each emulated wireless client device is attached to one separate physical USB wireless adapter. Further, in the said paper, all emulated wireless client devices is not provided to use its own USB adapter to send network traffic to and from the SUT in parallel (i.e. concurrently) as proposed in the present invention.

SUMMARY OF INVENTION

The present invention relates to a system and method for emulating multiple independent wireless client devices in the cloud. In particular, the invention initiates
5 wireless network performance test and gathers test results from multiple independent wireless client devices which simplifies and assists in the process of analyzing and reporting test results received from multiple independent wireless client devices.

One aspect of the present invention provides a system (100) for emulating multiple
10 independent wireless client devices in the cloud comprising. The system comprising a cloud platform (104) consisting a plurality of virtual machines (110a-110n); a plurality of USB wireless adapters (106); and a device agent (102). The device agent (102) having means for reading configuration files sent by users; scanning and discovering the plurality of USB wireless adapters attached to the cloud platform; launching virtual
15 machines in the cloud platform based on settings in configuration files and number of available USB wireless adapters; mapping USB wireless adapters to virtual machines by attaching USB wireless adapters to virtual machines in the cloud platform to create multiple independent wireless client devices; injecting test automation scripts into all virtual machines; initiating test execution in all virtual machines; gathering test results
20 from all virtual machines; and generating report of test results.

Another aspect of the invention provides that configuration files sent by user further comprising vendor ID and product ID of the USB wireless adapters to be used in the system; list of base virtual machine images selected to be used in the system; list of test
25 automation scripts selected to be used in the system; mapping of USB wireless adapters to virtual machine images; and number of emulated devices for each combination of virtual machine images and test automation scripts and USB wireless adapters.

A further aspect of the invention provides a method (200) for emulating multiple
30 independent wireless client devices in the cloud. The method comprising steps of setting up test environment by user (202); reading files by Device Agent and providing list of USB devices in cloud platform which can be used (204); specifying number of wireless client to be created by user (206); launching different virtual machines in the cloud platform by the Device Agent by using virtual machine images provided by the user

- 4 -

based on settings in the configuration files (208); mapping and attaching physical USB devices to virtual machines in the cloud platform to create multiple independent wireless client device by the Device Agent (210); injecting test automation scripts into all virtual machines by the Device Agent and initiating test execution in all virtual machine in the cloud platform based on settings in the configuration files (212); and gathering results from all virtual machines by the Device Agent upon completion of test execution based on the settings in the configuration files (214); and analyzing test results received from all virtual machines and generating test report (216). The step for mapping and attaching physical USB devices to virtual machines in the cloud platform to create multiple independent wireless client device by the Device Agent (210); each wireless client device of the multiple independent wireless client device is attached to one separate physical USB wireless adapter (106).

In another aspect of the invention there is provided that the step for setting up test environment by user further comprises setting vendor ID and product ID in Device Agent's configuration files.

In yet another aspect of the invention is the step for specifying the number of wireless client to be created by user which further comprises the step for providing automated test scripts and base virtual machine image to be used by the user.

In a further aspect of the invention there is provided that the step for injecting test automation scripts into all virtual machines by the Device Agent and initiating test execution in all virtual machines in the cloud platform based on settings in the configuration files further comprises monitoring test execution status in all virtual machines in the cloud platform based on the settings in the configuration files.

In still another aspect of the invention there is provided that said automated test scripts will be tagged with one unique ID and generated test report includes total number of clients SUCCESSFULLY completed GUI automation script; total number of clients FAILED to run GUI automation script; response time on each client; and throughput on each client.

- 5 -

The present invention consists of features and a combination of parts hereinafter fully described and illustrated in the accompanying drawings, it being understood that various changes in the details may be made without departing from the scope of the invention or sacrificing any of the advantages of the present invention.

- 6 -

BRIEF DESCRIPTION OF ACCOMPANYING DRAWINGS

To further clarify various aspects of some embodiments of the present invention, a more particular description of the invention will be rendered by references to specific
5 embodiments thereof, which are illustrated in the appended drawings. It is appreciated that these drawings depict only typical embodiments of the invention and are therefore not to be considered limiting of its scope. The invention will be described and explained with additional specificity and detail through the accompanying drawings in which:

10 FIG. 1 illustrates the general architecture of the present invention.

FIG. 2 is a flowchart illustrating the general methodology of an embodiment of the present invention.

- 7 -

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention relates to a system and method for emulating multiple independent wireless client devices in the cloud. In particular, the invention initiates
5 wireless network performance test and gathers test results from multiple independent wireless client devices which simplifies and assists in the process of analyzing and reporting test results received from multiple independent wireless client devices.

Hereinafter, this specification will describe the present invention according to the
10 preferred embodiments. It is to be understood that limiting the description to the preferred embodiments of the invention is merely to facilitate discussion of the present invention and it is envisioned without departing from the scope of the appended claims.

Referring to FIG. 1.0, the system (100) for emulating multiple independent wireless client
15 devices in the cloud according to the present invention is illustrated. The system (100) comprising a cloud platform (104) consisting a plurality of virtual machines (110a-110n); a plurality of USB wireless adapters (106); and a device agent (102). Servers in a cloud platform which can host virtual machines are known as "node" or "node controller". Device agent (102) will read configuration files sent by users; scans and discover the
20 plurality of USB wireless adapters attached to the cloud platform; launch virtual machines in the cloud platform based on settings in configuration files and number of available USB wireless adapters; map USB wireless adapters to virtual machines by attaching USB wireless adapters to virtual machines in the cloud platform to create multiple independent wireless client devices; injects test automation scripts into all virtual
25 machines; initiates test execution in all virtual machines; gathers test results from all virtual machines; and generates report of test results. The virtual machines (110a-110n) have exclusive access to its own USB wireless adapter (106) and the configuration files sent by user further comprising vendor ID and product ID of the USB wireless adapters to be used in the system; list of base virtual machine images selected to be used in the
30 system; list of test automation scripts selected to be used in the system; mapping of USB wireless adapters to virtual machine images; and number of emulated devices for each combination of virtual machine images and test automation scripts and USB wireless adapters.

- 8 -

The general methodology (200) of an embodiment of the present invention is as illustrated in FIG. 2.0. As illustrated in FIG. 2.0, the test environment is first set up by the user (202). Device Agent will read files (204) wherein device agent will find all nodes in a cloud platform using the cloud platform's API. Device agent will tag all nodes with a unique node ID. Example of node ID ((Node ID="0" – node1, IP=10.1.2.101) and (Node ID="1" – node2, IP=10.1.2.102)). An example of cloud platform API's output (taken from PQRE cloud) is as follows.

```
[qweasd@milennia-fe ~]$
[qweasd@milennia-fe ~]$ nodehost list
```

ID	NAME	CLUSTER	RVM	ALLOCATED CPU	ALLOCATED MEM	STAT
0	alm-n1	-	1	400 / 1024 (39%)	0.53 / 31.40 (21%)	
1	alm-n2	-	3	400 / 1024 (39%)	0.53 / 31.40 (21%)	

```
[qweasd@milennia-fe ~]$
[qweasd@milennia-fe ~]$
```

Device agent will login to all nodes in the cloud platform 1-by-1. Device agent will scan the USB devices attached to a node in the cloud platform and said step is repeated for all nodes in the cloud. Thereafter, device agent will provide a list of USB devices in the cloud platform which can be used (204). An example of USB devices in 1 node is provided as follows.

```
Bus 001 Device 002: ID 0bda:0111 Realtek Semiconductor Corp. Card Reader
Bus 001 Device 003: ID 1a40:0101 Terminus Technology Inc. 4-Port HUB
Bus 002 Device 018: ID 1a40:0101 Terminus Technology Inc. 4-Port HUB
Bus 006 Device 002: ID 6993:b700 Yealink Network Technology Co., Ltd.
Bus 007 Device 011: ID 0a81:0101 Chesen Electronics Corp. Keyboard
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 006 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 009: ID 148f:5370 Ralink Technology, Corp. RT5370 Wireless Adapter
Bus 001 Device 010: ID 148f:5370 Ralink Technology, Corp. RT5370 Wireless Adapter
Bus 001 Device 011: ID 148f:5370 Ralink Technology, Corp. RT5370 Wireless Adapter
Bus 002 Device 019: ID 148f:5370 Ralink Technology, Corp. RT5370 Wireless Adapter
Bus 002 Device 020: ID 148f:5370 Ralink Technology, Corp. RT5370 Wireless Adapter
Bus 002 Device 021: ID 1687:3257 Kingmax Digital Inc.
Bus 002 Device 022: ID 0951:1607 Kingston Technology DataTraveler 100
root@qweasd-MILLENIA-R3:/home/qweasd#
root@qweasd-MILLENIA-R3:/home/qweasd#
```

Based on the results obtained, Device agent will create an array (a list) to represent the USB in which the number of wireless client to be created by the user will be specified (206) by providing automated test scripts and base virtual machine image to be used by the user.

- 9 -

Each element in the created array contains the following data:

arrayElement[n] = {Node ID, USB Bus ID, USB Device ID, USB Vendor ID, USB Product ID}

5

```

ArrayElement[0] = {0, 001, 001, 1d6d, 5370}
ArrayElement[1] = {0, 002, 003, 0951, 1607}
ArrayElement[2] = {1, 002, 001, 1d6d, 5370}

.....
ArrayElement[50] = {6, 002, 004, 1d6d, 5370}

```

(1)

- a. "USB Bus ID + USB Device ID" represents the "address" that **uniquely identify a USB device in a host system**
- b. "USB Vendor ID + USB Product ID" **uniquely identify 1 type of USB device.**
- c. "Node ID + USB Bus ID + USB Device ID" **uniquely identify a USB device attached to a node in the cloud platform.**

10

15

The device agent will read the configuration file and obtain the types of USB adapters selected by the user. Device agent will tag all USB adapter types with a unique ID. For example, if a user intends to use the following USB devices in the cloud, it will be tag as follows.

20

- a. TP-LINK TL-WN727N, tag ID=ID0
- b. TP-LINK TL-WN722N, tag ID=ID1
- c. TP-LINK TL-WN822N, tag ID=ID2

Accordingly, the device agent will generate an array which represents the USB devices in the cloud platform **which matches the Vendor ID and Product ID** set in the configuration file. The length of this array will be less than or equals to the first set of array generated because the USB devices which do not match the Vendor ID and Product ID will be filtered out.

25

```

ArrayElement[0] = {0, 001, 001, 1d6d, 5370}
ArrayElement[1] = {0, 002, 003, 0951, 1607}

```

(2)

- 10 -

```
ArrayElement[2] = {1, 002, 001, 1d6d, 5370}
```

```
.....
```

```
ArrayElement[30] = {6, 002, 004, 1d6d, 5370}
```

- Based on the ~~the~~ second set of array obtained above, device agent will calculate and generate an array which represents the number of "each type of USB device" attached to "each of the nodes" in the cloud. Each element in the array represents the data for 1 node. The length of the array is equals to the number of nodes in the cloud.
- 5 arrayElement[0] = {Node ID 0, number of **USB adapter ID0** attached to Node 0, number of **USB adapter ID1** attached to Node 0, ...}

```
ArrayElement[0] = {0, 6, 6, 0, ...}
```

```
ArrayElement[1] = {1, 4, 10, 0, ...}
```

```
.....
```

```
ArrayElement[6] = {6, 0, 6, 2, ...}
```

(3)

10

Thereafter, device agent will locate the available resources in all nodes in the cloud using the cloud platform's API. Device agent will create an array to represent the available resources of all nodes in the cloud platform. Each element in the array represents the specification of 1 node in the cloud.

- 15 arrayElement[0] = {Node ID, number of CPU Core, amount of RAM in MB, size of harddisk storage in GB}

```
ArrayElement[0] = {0, 8, 16000, 500}
```

```
ArrayElement[1] = {1, 8, 16000, 500}
```

```
.....
```

```
ArrayElement[6] = {6, 8, 16000, 500}
```

(4)

- 11 -

Device agent will combine the array in (3) and (4).

arrayElement[n] = {Node ID, number of CPU Core, amount of RAM in MB, size of harddisk storage in GB, number of **USB adapter ID0** attached to Node n, number of **USB adapter ID1** attached to Node n, ...}

5

```

ArrayElement[0] = {0, 8, 16000, 500, 6, 6, 0, ...}
ArrayElement[1] = {1, 8, 16000, 500, 4, 10, 0, ...}
.....
ArrayElement[6] = {6, 8, 16000, 500, 0, 6, 2, ...}

```

(5)

Device agent will read the configuration file and obtain

- a. all combinations of "base virtual machine image type + USB wireless adapter type" that the user wants to use, and
- b. the number of virtual machine for each combination type.

10

For example:

a. Combination 1:

- i. Base virtual machine image = CentOS-6.4-x86_64_ID1
- ii. USB wireless adapter = TP-LINK TL-WN727N
- iii. Number of virtual machine = 20

15

b. Combination 2:

- iv. Base virtual machine image = ubuntu-13.04-desktop-i386_ID2
- v. USB wireless adapter = TP-LINK TL-WN822N
- vi. Number of virtual machine = 10

20

For each type of virtual machine requested by the user, the device agent will

- a. check the array in (5) and locate a node which has the required resources (USB adapter + CPU + RAM + hard disk)
- b. Launch the selected VM type in the selected node. The VM will be tagged with a unique VM ID.
- c. Device agent will check the array in [2] and create an array with the following format. Each element in the array represents 1 VM launched by the device agent and the mapping to its designated USB adapter.

25

30

- 12 -

arrayElement[n] = {Node ID, VM ID, USB Bus ID, USB Device ID}

```

ArrayElement[0] = {0, 1, 1d6d, 5370}
ArrayElement[1] = {0, 2, 0951, 1607}
ArrayElement[2] = {1, 3, 1d6d, 5370}
.....
ArrayElement[6] = {6, 30, 1d6d, 5370}

```

(6)

- 5 d. Update the array in [5]. Hardware resources (USB adapter + CPU + RAM + harddisk) which have been used in .b] will be removed from the array.

Thereafter, the device agent will launch different virtual machine images in the cloud platform by using virtual machine images provided by the user based on settings in the configuration files (208). The device agent will generate an error message if there are not enough resources in the cloud. The error message will be saved in a variable which will be shown to the user at the end of the system task. After launching all virtual machines, the device agent will perform "USB passthrough" to attach the USB adapters directly to the designated VM in the cloud. The method used to perform "USB passthrough" depends on the types of hypervisor used in the cloud. Different hypervisor will use different methods to perform USB passthrough.

The following is an example to perform USB passthrough in a node which utilizes KVM hypervisor and is managed by libvirt. Device agent will check the array in (6) to locate the USB Bus ID and USB Device ID of a USB device which needs to be attached to a virtual machine. Device agent will prepare a XML file with the following format:

```

<hostdev mode='subsystem' type='usb'>
  <source>
    <address bus="001" device="002"/>
  </source>
</hostdev>

```

(7)

- 13 -

By referring to the array in (6), the device agent will perform "USB passthrough" to attach the USB adapter directly to the designated virtual machine. In this example, device agent will utilize the libvirt tool's API and the XML file in (7).

- 5 Device agent will repeat the step to perform USB passthrough in a node for all virtual machines in array (6). The device agent will login to all virtual machines in array (6) and the device agent will verify that all virtual machines have been attached with one USB adapter. Thereafter, the device agent will read the configuration file to obtain the list of test automation script selected by the users. All test automation scripts will be tagged
10 with one unique ID. For example: Script 1, tag ID=0 and Script 2, tag ID=1.

The device agent will update the array in (6) with the mapping to the designated test script. Each element in the array represents one virtual machine, its designated USB adapter and its designated test automation script.

- 15 The format of the array is as below:
arrayElement[n] = {Node ID, VM ID, USB Bus ID, USB Device ID, Test Script Tag ID}

```

ArrayElement[0] = {0, 1, 1d6d, 5370, 0}
ArrayElement[1] = {0, 2, 0951, 1607, 0}
ArrayElement[2] = {1, 3, 1d6d, 5370, 1}
.....
ArrayElement[6] = {6, 30, 1d6d, 5370, 0}

```

(8)

- 20 Physical USB devices are mapped and attached to virtual machines in the cloud platform to create multiple independent wireless client devices by the device agent (210). By referring to the array in (8), device agent will inject the selected graphical user interface (GUI) test automation script into all virtual machines in the cloud platform based on settings in the configuration files (212).

- 25 The protocol that can be used includes SCP, FTP, TFTP, SFTP, etc. Device agent will repeat said step of injecting test automation scripts for all virtual machines in array (8). Subsequently, device agent will initiate test execution in all virtual machines in array (212). Device agent will check and monitor the test execution status in all virtual

- 14 -

machines and upon completion or occurrence of timeout error in all virtual machines, the device agent will gather the test results from all virtual machines in array (8) based on the settings in the configuration files (214). Subsequently, the device agent will analyze the test results received from all virtual machines and generate a test report (216). The
5 generated test report includes total number of clients SUCCESSFULLY completed GUI automation script; total number of clients FAILED to run GUI automation script; response time on each client; and throughput on each client.

The present invention simplifies and assists the process of running / initiating wireless
10 network performance test and gathers the test results from multiple independent wireless client devices which further simplifies and assists in the process of analyzing and reporting test results received from multiple independent wireless client devices. Each wireless client device is attached to one separate physical USB wireless adapter (106) and the said wireless client devices is provided with exclusive access to its own USB
15 wireless adapter to connect to the System Under Test (SUT) directly and is able to run its own operating system and applications in parallel (i.e. concurrently).

The present invention may be embodied in other specific forms without departing from its essential characteristics. The described embodiments are to be considered in all
20 respects only as illustrative and not restrictive. The scope of the invention is, therefore indicated by the appended claims rather than by the foregoing description. All changes, which come within the meaning and range of equivalency of the claims, are to be embraced within their scope.

- 15 -

CLAIMS

1. A system (100) for emulating multiple independent wireless client devices in the cloud comprising:

5 a cloud platform (104) consisting a plurality of virtual machines (110a-110n);
 a plurality of USB wireless adapters (106); and
 a device agent (102)
 characterized in that
10 the device agent (102) having means for:
 reading configuration files sent by users;
 scanning and discovering the plurality of USB wireless adapters attached to the cloud platform;
 launching virtual machines in the cloud platform based on settings in
15 configuration files and number of available USB wireless adapters;
 mapping USB wireless adapters to virtual machines by attaching USB wireless adapters to virtual machines in the cloud platform to create multiple independent wireless client devices;
 injecting test automation scripts into all virtual machines;
20 initiating test execution in all virtual machines;
 gathering test results from all virtual machines; and
 generating report of test results.

2. A system (100) according to Claim 1, wherein the plurality of virtual machines
25 (110a-110n) have exclusive access to its own USB wireless adapter (106).

3. A system (100) according to Claim 1, wherein configuration files sent by user further comprising:

30 vendor ID and product ID of the USB wireless adapters to be used in the system;
 list of base virtual machine images selected to be used in the system;
 list of test automation scripts selected to be used in the system;
 mapping of USB wireless adapters to virtual machine images; and

- 16 -

number of emulated devices for each combination of virtual machine images and test automation scripts and USB wireless adapters.

4. A method (200) for emulating multiple independent wireless client devices in the cloud comprising steps of:

5 setting up test environment by user (202);
 reading files by Device Agent and providing list of USB devices in cloud
 platform which can be used (204);
 specifying number of wireless client to be created by user (206);
10 launching different virtual machines in the cloud platform by the Device
 Agent by using virtual machine images provided by the user based on
 settings in the configuration files (208);
 mapping and attaching physical USB devices to virtual machines in
 the cloud platform to create multiple independent wireless client device
15 by the Device Agent (210);
 injecting test automation scripts into all virtual machines by the Device
 Agent and initiating test execution in all virtual machine in the cloud
 platform based on settings in the configuration files (212); and
 gathering results from all virtual machines by the Device Agent upon
20 completion of test execution based on the settings in the configuration
 files (214); and
 analyzing test results received from all virtual machines and
 generating test report (216)
 characterized in that
25 mapping and attaching physical USB devices to virtual machines in
 the cloud platform to create multiple independent wireless client device
 by the Device Agent (210); each wireless client device of the
 multiple independent wireless client device is attached to one
 separate physical USB wireless adapter.

- 30 5. A method (200) according to Claim 4, wherein setting up test environment by
 user further comprises setting vendor ID and product ID in Device Agent's
 configuration files.

- 17 -

6. A method (200) according to Claim 4, wherein specifying number of wireless client to be created by user further comprises providing automated test scripts and base virtual machine image to be used by the user.
- 5 7. A method (200) according to Claim 4, wherein injecting test automation scripts into all virtual machines by the Device Agent and initiating test execution in all virtual machines in the cloud platform based on settings in the configuration files further comprises monitoring test execution status in all virtual machines in the cloud platform based on the settings in the configuration files.
- 10 8. A method (200) according to Claim 6, wherein said automated test scripts will be tagged with one unique ID.
- 15 9. A method (200) according to Claim 4, wherein generated test report includes:
 - total number of clients SUCCESSFULLY completed GUI automation script;
 - total number of clients FAILED to run GUI automation script;
 - response time on each client; and
 - throughput on each client.

1/2

100

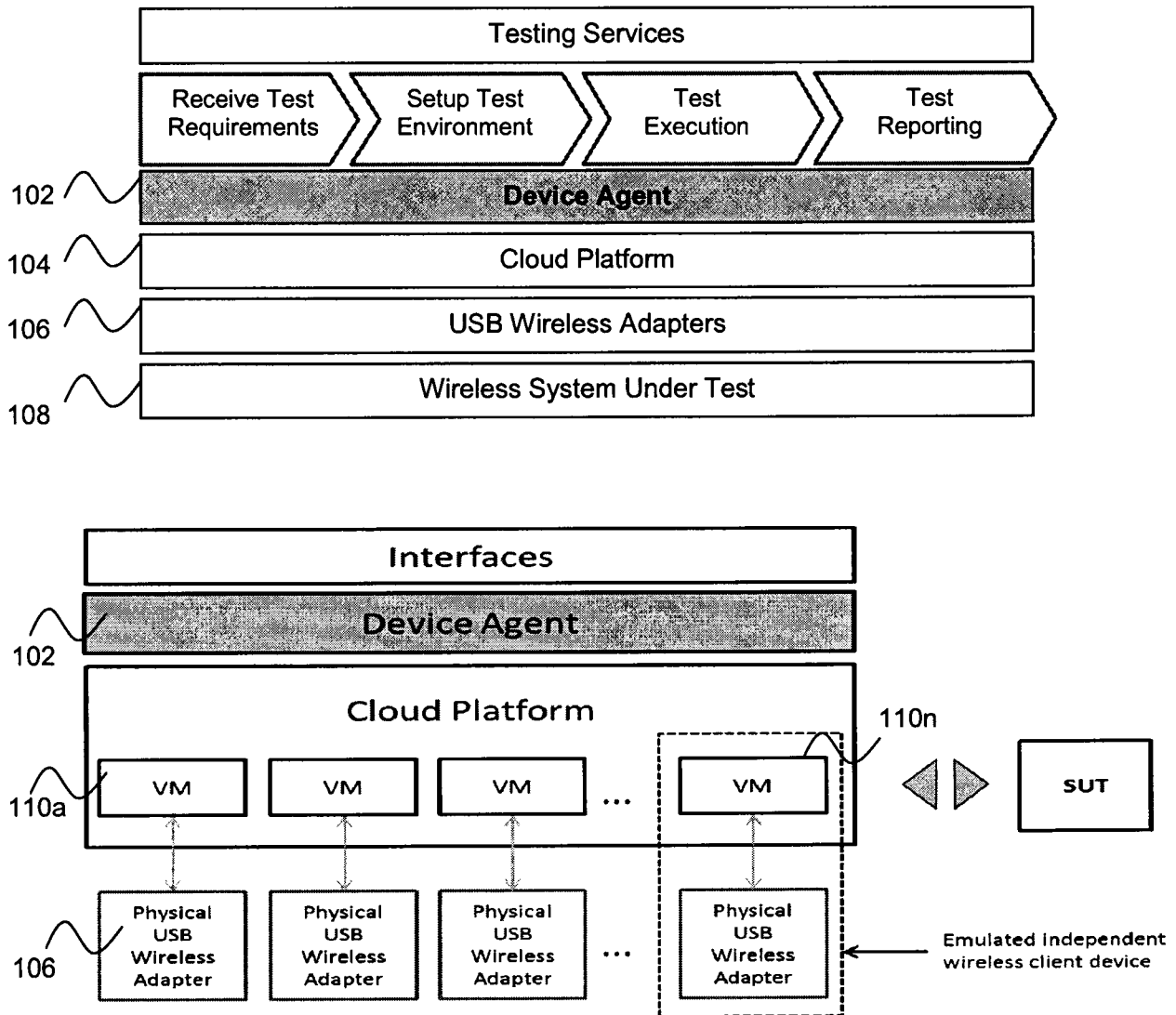


FIG. 1.0

2/2

200

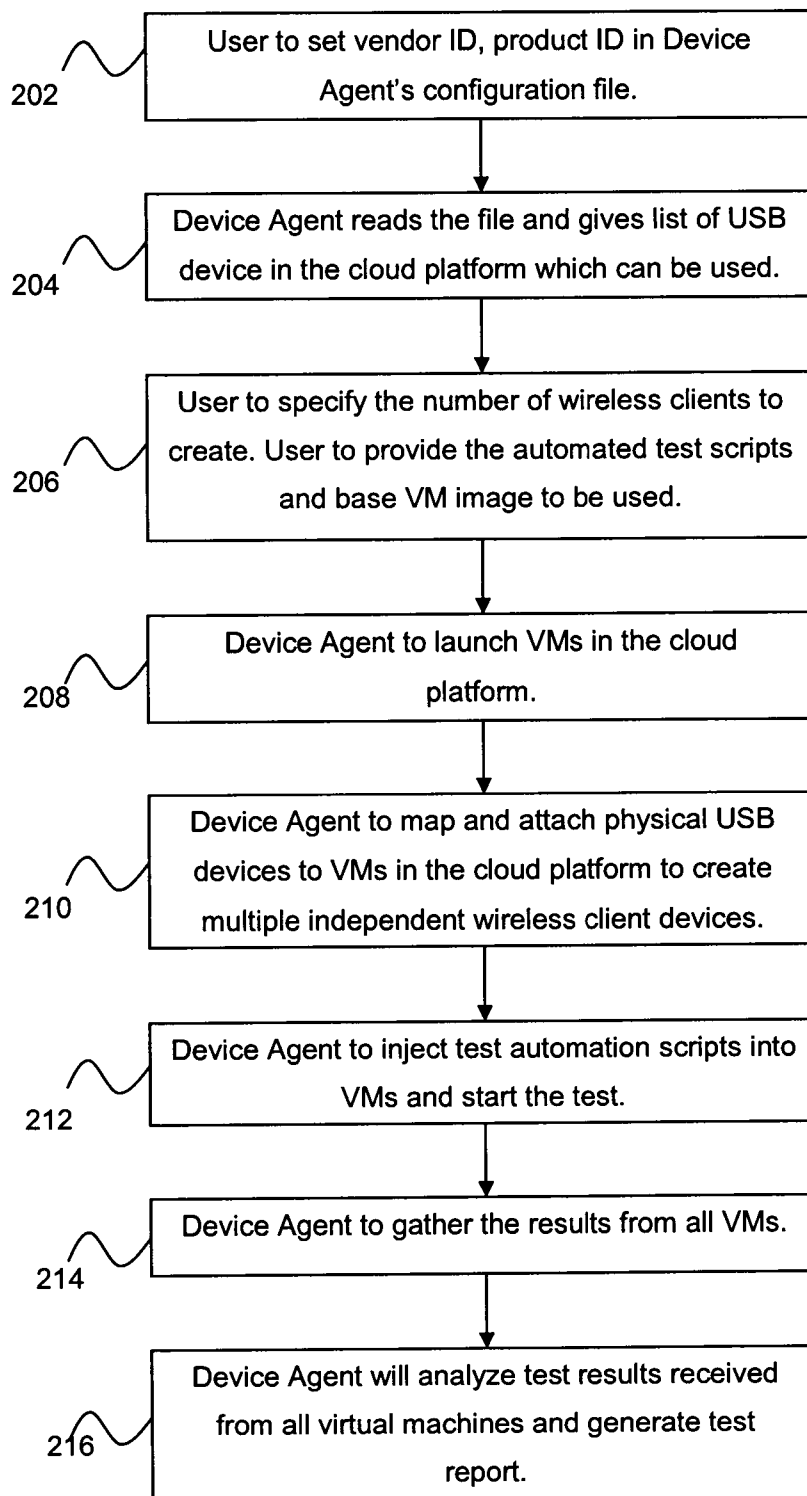


FIG. 2.0

INTERNATIONAL SEARCH REPORT

International application No
PCT/MY2014/000147

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F11/30 G06F11/34 H04W24/00 G06F11/36
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F H04W

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2011/010691 A1 (LU ZIHONG [US] ET AL) 13 January 2011 (2011-01-13) the whole document -----	1-9
A	US 2003/202486 A1 (ANTON FRANCIS M [US] ET AL ANTON JR FRANCIS M [US] ET AL) 30 October 2003 (2003-10-30) cited in the application abstract figures 7-8 -----	1-9
A	US 2013/272162 A1 (LI LI [CN] ET AL) 17 October 2013 (2013-10-17) abstract figures 7-8 paragraph [0007] - paragraph [0008] ----- -/-	1-9



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

8 January 2015

Date of mailing of the international search report

15/01/2015

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Lanchès, Philippe

INTERNATIONAL SEARCH REPORT

International application No

PCT/MY2014/000147

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2012/311128 A1 (PECHANEC JIRI [CZ] ET AL PECHANEC JIRÍ [CZ] ET AL) 6 December 2012 (2012-12-06) paragraph [0014] - paragraph [0022] -----	1-9

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/MY2014/000147

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2011010691	A1	13-01-2011	NONE
US 2003202486	A1	30-10-2003	AU 2003225046 A1 17-11-2003 CA 2426240 A1 29-10-2003 TW I244295 B 21-11-2005 US 2003202486 A1 30-10-2003 WO 03094450 A1 13-11-2003
US 2013272162	A1	17-10-2013	CN 103368910 A 23-10-2013 US 2013272162 A1 17-10-2013
US 2012311128	A1	06-12-2012	NONE