

## **EXHIBIT 2-9**

UNITED STATES DISTRICT COURT  
NORTHERN DISTRICT OF CALIFORNIA  
SAN FRANCISCO DIVISION

ORACLE AMERICA, INC.

Plaintiff,

v.

GOOGLE INC.

Defendant.

Case No. CV 10-03561 WHA

**SUMMARY AND REPORT OF EREZ LANDAU**

**SUBMITTED ON BEHALF OF PLAINTIFF  
ORACLE AMERICA, INC.**

## **II. BACKGROUND AND QUALIFICATIONS**

8. I graduated from Technion - Israel Institute of Technology as a computer engineer, and worked as a computer engineer since 1984.

9. I have over 25 years of experience in software engineering and performance optimization. During my career, I have conducted numerous performance measurements, identified performance bottlenecks, and invented techniques to improve performance.

10. Since the Sun Microsystems acquisition, I have worked on developing an embedded Java virtual machine future architecture, and porting the Java virtual machine to new platforms. (I will generally refer to the Java virtual machine as the “JVM”).

11. I am qualified to conduct performance benchmark and testing analysis and analyze the results obtained because I have done performance measurements numerous times before for internal measurements and for customer specific projects.

## **B. Devices and Software Distributions**

18. We selected a single production phone, namely the Android HTC Nexus One.

Here are the specifications and links to the board and operating system platform that we installed and ran on two Android HTC Nexus One devices: <http://www.google.com/phone/detail/nexus-one>.

19. The HTC Nexus One is a good test environment because it is representative of a real Android phone. Memory consumption is not expected to vary between phones and startup time is expected to stay proportional to other timing on the phone. Under these assumptions it is expected that the test performed will produce similar results on other Android phones as well.

## **C. Android Sources**

20. The Android sources that were used to build the Dalvik VM used during these tests were downloaded by following the Google instructions from this Google URL and specifying the froyo tag (version 2.2.1) in the git command.

<http://source.android.com/source/downloading.html>

<http://source.android.com/source/initializing.html>

\$ repo init -u git://android.git.kernel.org/platform/manifest.git -b froyo

The Android sources were built using the instruction located at this URL:

<http://source.android.com/source/building.html>

<http://source.android.com/source/initializing.html>

## **D. Applications Tested**

21. We tested Google standard applications available in the Android repository:

android.process.acore

android.process.media

com.android.bluetooth

com.android.deskclock

com.android.email

com.android.inputmethod.latin

com.android.launcher

com.android.mms  
com.android.music  
com.android.phone  
com.android.protips  
com.android.providers.calendar  
com.android.quicksearchbox  
com.android.settings  
com.android.voicedialer  
com.android.wallpaper  
com.cooliris.media

22. The reason for this selection of applications is these are representative of applications that Google makes available through the public Android source code repository. This will allow Google to validate this test on any platform they choose to.

23. Of note, the zygote process is the initial virtual machine (“master”) and solely exists to be cloned as new virtual machine instances for launching a new application.

24. The system\_server process is also a virtual machine instance but is cloned somewhat differently. I did not make any source code modifications that would impact system\_server.

#### **IV. PERFORMANCE BENCHMARK AND TESTING ANALYSIS OF U.S. PATENT NO. 7,426,720**

##### **A. Overview**

25. The baseline for these experiments was the Froyo release of Android, pulled from the Google git repository with the Google repo commands

```
$ repo init -u git://android.git.kernel.org/platform/manifest.git -b froyo
```

```
$ repo sync
```

to initialize and sync from the Google repository. I made two copies of the repository that contained the Android system: (1) an unmodified copy to serve as a baseline for comparison and (2) a version with the zygote functionality that Oracle accuses of infringing the '720 patent disabled.

26. I disabled the use of the accused functionality in Android as discussed in Appendix A.

27. The modification for experiments were restricted to two files:

- a. frameworks/base/core/java/com/android/internal/os/RuntimeInit.java and
- b. frameworks/base/core/jni/AndroidRuntime.cpp.

In summary, the zygote process which is the initial (master) virtual machine which manages the virtual machine cloning (and thus sharing) was modified to call the new 'exec' method for the new created process (clone), so it re-initializes the process as a fresh new virtual machine (over running whatever was inherited from the master virtual machine) thus disabling the '720 patent. The modifications are detailed and shown in Appendix A.

28. All workspaces were compiled and loaded to the device following the directions described at Google's <http://source.android.com/source/initializing.html>.

29. The benchmarks used to test the performance of the workspaces were all Google standard applications available in the repository, as listed above.

30. I tested the performance of the repositories by running them on the HTC Nexus One with the two workspaces installed on two separate phones: (1) one phone to test the

unmodified Android source code (where the functionality that Oracle accuses of infringing the '720 patent is practiced and not disabled – labeled “Original (copy-on-write)” in Exhibit A) and (2) another phone to test the modified Android source code (disabling the functionality that Oracle accuses of infringing the '720 patent – labeled “No VM Sharing” in Exhibit A).

31. I ran all benchmark applications on both phones.

32. The accompanying charts attached as Exhibit A records the results of runs of a standard phone startup which launch all the default applications.

#### **B. Mtask ClassLoading Test**

33. I combined HelloWorld and MtaskClassLoading tests in a single application with Google HelloActivity which says “hello” and does the class loading. That is to say, the MtaskClassLoading test says “hello” and does class loading of a given class list. The test is actually a modified version of android HelloActivity sample available under folder ‘development/samples/HelloActivity’ adding a call to MtaskClassLoading.main() method to do the class loading.

34. The class list is made of Android relevant preloaded classes. That is, the list of classes to load was taken from the Android preloaded class list.

35. Test code is attached as Exhibit B.

36. Logcat dumps are attached as Exhibits C and D.

#### **V. CONCLUSION**

37. The performance benchmark testing shows as much as 40% memory savings (Android uses 51MB instead of 86MB) and also saves .10 second per application launch time. In other words, disabling the functionality that Oracle accuses of infringing the '720 patent increases the memory consumption by 70% (because 86 is 1.7 times 51).

38. Memory savings were 2MB per each running application. This becomes more significant when executing many small widgets with a small amount of required unique memory consumption. Moreover, the results demonstrate that for applications that need more of the