

Vue 6 Infinite Performance Analysis

By Frédéric Louguet

V1.04 (July 1st, 2007)

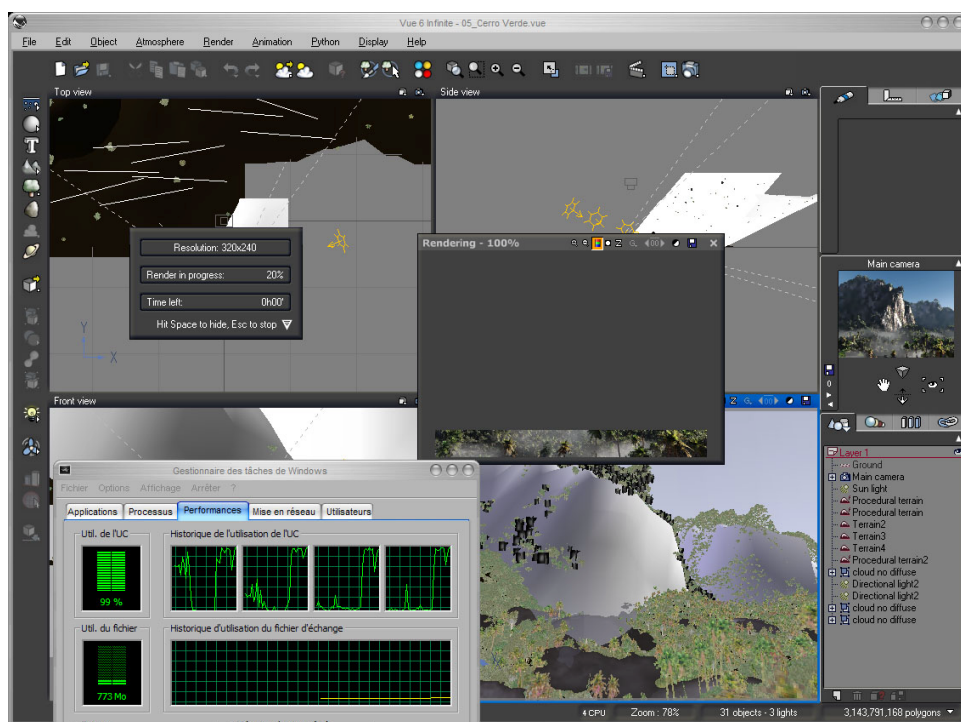
Abstract

The purpose of this document is to provide a brief technical analysis of Vue 6 Infinite (V6I) performance. You will find a detailed analysis of the differences between the 32-bit and 64-bit versions of V6I, a discussion of the different factors which can affect performance, a description of the recent changes in the render presets default parameters... This document should also provide some answers to the following questions:

- what are the benefits and drawbacks of using the 64-bit version?
- what performance can be expected from multiCPUs / multicores machines with V6I?
- how do the render times increase with resolution?

Table of contents:

1) 64-bit vs 32-bit	page 2
2) Multi-CPU's and multi-cores support	page 5
3) Other performance factors	page 7
4) Recent changes in default parameters (for the preset modes)	page 8
5) Non-linear render time increase with procedural terrains / clouds	page 9
6) VueMark: the standard benchmark for Vue Infinite	page 10
7) History	page 11



1) 64-bit vs 32-bit

Vue 6 Infinite is the first version of Vue to be available in 32-bit and 64-bit builds. It is a significant step in the development of the software. In this section we will discuss the advantages and issues associated with the 64-bit build. Both versions (32 & 64-bit) are included with V6I, and the correct version is automatically installed depending on your OS.

1.1) What are the benefits of running the 64-bit version of V6I?

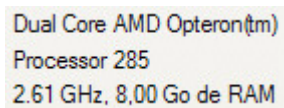
1.1.a) The main benefit is memory addressing.

In Windows XP / Vista 32-bit, each application is limited to a 2 GB address space. Even on a PC with 4 or 8 GB RAM, the 32-bit applications are not able to address more than 2 GB each. Also, Windows itself is not able to see more than 2.75 or 3.00 GB RAM. On the example below, the machine is a dual Opteron 285 with 8 GB RAM, but Windows 32-bit only sees 3 GB RAM:



Dual Core AMD Opteron(tm)
Processor 285
2.61 GHz, 3,00 Go de RAM

In Windows 64-bit, the correct amount is displayed and fully used:

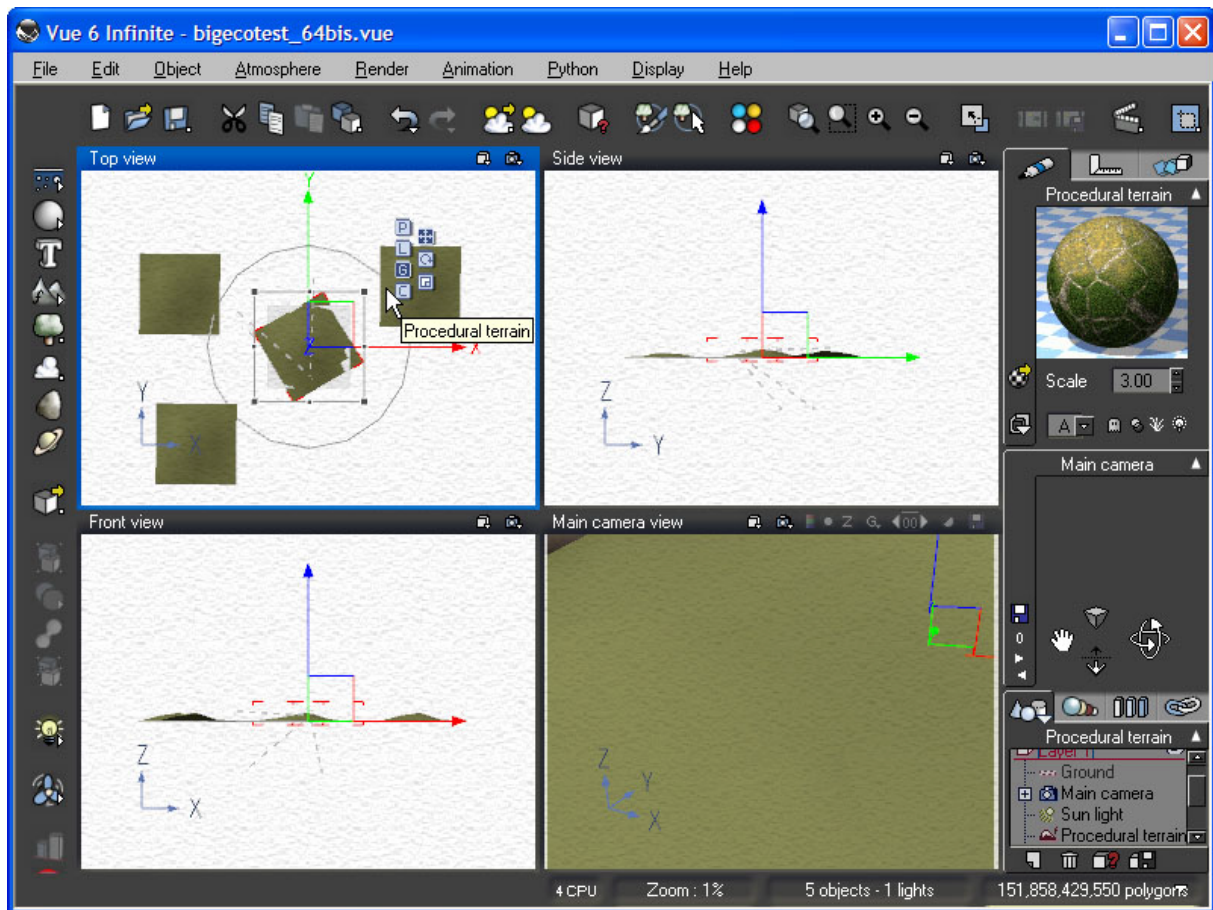


Dual Core AMD Opteron(tm)
Processor 285
2.61 GHz, 8,00 Go de RAM

With the 64-bit version of V6I, it is possible to create more complex scenes, with 3D models of higher geometric complexity, and high resolution textures, and not be bothered by the dreaded “*low resources*” messages. The OOM (*Out Of Memory*) errors become very rare. In fact, the ability to create very complex scenes is limited by the amount of memory installed in the computer, rather than by the operating system.

Of course, there is no benefit in installing Windows XP (or Vista) 64-bit on a 2 GB, or less, system. The advantages of 64-bit computing only show themselves with 4 GB RAM or more.

As an example, it is possible to create ecosystems with a massive amount of polygons (more than 150 billion below), something that is totally impossible with the 32-bit version:



Such scenes can now be saved on disk because Vue 6 does not have the 2 GB file size limit of Vue 5. For instance, a 120 billion-polygon, “Garden” preset ecosystem weighs more than 6 gigabytes! But when loaded in memory on a 8 GB system, you still have 50% resources.

Obviously such scenes will drag even the most powerful machines down, because the computer speed does not increase in the same proportions... This example is just here to show that it is possible to create unbelievably complex scenes with the 64-bit version.

1.1.b) Second benefit: a small speed increase

The 64-bit version of V6I renders most scenes about 5 to 15% faster than the 32-bit version. It is maybe not spectacular, but it is far from negligible. Generally, 3D applications do not gain a lot of rendering speed when going to 64-bit, because they do not use a lot of 64-bit variables / operations. Vue is not alone, Lightwave 3D or Cinema 4D show the same behavior.

Of course, if Vue 6 (32-bit) has difficulties to render, or even load, a heavy scene, Vue 6 (64-bit) will provide a much better performance, simply because it can address more memory. But for scenes which do not depend on the quantity of RAM installed, the performance will only be slightly better with the 64-bit version: you should not expect more than 10 to 15%, at best.

1.2) The drawbacks of running the 64-bit version of Vue

At the moment there is only one big problem: there is currently no 64-bit version of the wxPython library.

It means that Python scripts which use this library do not work in a 64-bit OS. That includes the popular SkinVue application (www.skinvue.net). A possible workaround:

- install a 32-bit and a 64-bit version of Windows in a dual-boot configuration.
- work on Poser characters with SkinVue in the 32-bit version.
- import them afterward in a big scene with the 64-bit version.

1.3) What do I need to run Vue 6 Infinite in 64-bit mode?

- 1) a PC with a 64-bit capable CPU. That means an AMD Athlon 64 / X2 / Opteron, or an Intel Pentium D / Xeon / Core 2 Duo / Quad.
- 2) a 64-bit Windows operating system, such as Windows XP Professional x64 edition, Windows Server 2003 64-bit, or Windows Vista 64-bit.
- 3) the 64-bit build of Vue 6 Infinite.

And it is highly recommended to have:

- 4 GB of system memory, or more. With 2 GB RAM or less memory, Windows XP64 does not have any memory addressing advantage over Windows XP32.

1.4) How do I recognize / install the right version?

The installation program detects on which OS you are running and installs the appropriate version. To determine which version of Vue is running, you can select the “*Help / About Vue 6 Infinite...*” menu option: it will show a “**64-bit**” logo, or nothing for the 32-bit version (see also the splash screen displayed when launching Vue). For the rendercow, right-click on the RC icon in the taskbar and choose “*About...*” to see which version is running.

2) Multi-CPUs and multi-cores support

Parallel processing and multithreading are important features, and particularly suited to 3D rendering. By using the combined power of several CPUs, or cores, render times - which by definition are always too long - can be spectacularly decreased.

2.1) Does Vue 6 Infinite support multithreading and multiCPUs / cores?

Yes. It has been the case since Vue 4 Professional. The Infinite and xStream versions are not core-limited nor CPU-limited, but the entry-level and mid-level versions of Vue (Easel, Esprit, Pro Studio) are limited to two simultaneous rendering threads. So from a practical point of view they are limited to two cores.

However, even with Vue 6 Infinite, you must be aware of the following limitations:

- panoramic still image rendering is NOT multithreaded
- preparation phases (e.g. for radiosity) are NOT multithreaded

2.2) Are there other possible limitations?

Even if V6I itself is not limited, the operating system can be (and the hardware obviously).

- Windows XP Family Edition, and Vista Starter, Home Basic and Premium are limited to ONE CPU (one socket), but it can be MULTI-CORES. Typical example: AMD Athlon 64 X2 (1 cpu x 2 cores), Intel Core 2 Quad (1 cpu x 4 cores).
- Windows XP Professional, Vista Business, Enterprise and Ultimate are limited to TWO CPUs (two sockets), each can be MULTI-CORES. Typical example: dual Opteron Dual Core (2 cpus x 2 cores), dual Xeon 53xx (2 cpus x 4 cores).
- For more than two physical CPUs, the Windows Server operating system is required.

2.3) What performance can I expect when going from one to two / four cores?

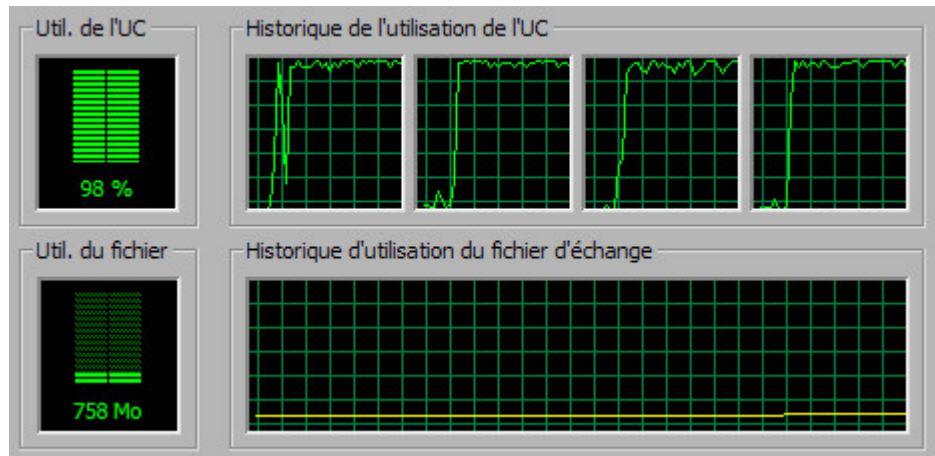
To answer that question, here are some render times with a well-known scene (Cerro Verde from the Vue 6 Infinite Extras CD), rendered in 640 x 480 pixels, Final mode, on a four cores machine (Dual Opteron 285) and with the 64-bit release version of V6I. The Vue 6 rendering engine was artificially limited (by controlling the affinity of the process) to use only one rendering thread, then two, three and finally four.

Here are the results: **[V6I 6.10 - build 291050]**

- 1 core : 3'14" (*it was 3'40" in the first V6I official build 289768*)
- 2 cores: 1'42" (x 1.902 times faster than 1 thread – 95.1% efficiency)
- 3 cores: 1'09" (x 2.812 times faster than 1 thread – 93.7% efficiency)
- 4 cores: 0'54" (x 3.593 times faster than 1 thread – 89.8% efficiency)

We see that the average gain tends to diminish with the number of cores, this is a perfectly normal behavior. To have near 90% efficiency with 4 cores is already an impressive performance.

Here is below a graph of CPU activity on a 4-cores machine shortly after the beginning of the render:



3) Other performance factors

3.1) Memory

Memory capacity has already been discussed. Memory speed has a minimal impact on render times. Expensive, low latency memory will not provide more than a 1-3% gain, at best. Knowing that such memory can cost twice the price of regular RAM, it is hardly worth it. Remember: CPU speed is the key for 3D rendering.

To demonstrate the small influence of memory bandwidth on Vue render times with fast computers, I used a Core 2 Extreme QX6700 machine running at 2933 MHz, and ran the VueMark (see section 6) with memory speeds of 667 MHz and 533 MHz (DDR II). Here are the results with different memory timings (tests were done with build 288741):

DDR II 667 MHz, timings 5-5-5-15: 633 VueMarks
DDR II 533 MHz, timings 4-4-4-12: 628 VueMarks
DDR II 533 MHz, timings 5-5-5-15: 626 VueMarks

So the performance delta is only 1.1 %, compared to the 25% difference in memory bandwidth between the DDR2-667 and DDR2-533 at the same timings.

3.2) Hard disk

The hard disk speed will have no influence on rendering whatsoever. A fast hard disk will only help to load and save scenes a bit faster. RAID 0 configurations will not have any impact on render speed either.

3.3) Graphics card

Buying a professional Quadro card for Vue is not recommended from a price / performance point of view. Quados often have weaker hardware than their GeForce equivalent, and are always much more expensive. The only real advantage that Quados have over more mainstream cards is an optimized and certified OpenGL driver which can give some benefits with *some* applications. It can be useful when doing heavy modeling with some high-end packages, like 3D Studio Max or Maya. In the case of Vue, the benefits would be negligible at best. The same analysis can be applied to ATI FireGL cards compared to the Radeon series.

A good mainstream GeForce 7900 / Radeon X1900 card will give a great performance for viewport manipulation in Vue (provided the operations involved are GPU-limited and not CPU-limited). If you want the fastest card available right now in the mainstream market, go for a GeForce 8800 GTX. However, it is important to know that the performance gaps between the various mainstream graphics cards are not as obvious in 3D creation applications as they are in games. You can easily work with a 7300 GT in Vue, although its performance is terrible with recent games.

Important: Vue does NOT use the GPU power for rendering at all, so the best graphics card in the world will not have any effect on render times (hopefully that will change in the future, but no one knows when). Also, a SLI (multi-GPUs) configuration will not have any impact on Vue's OpenGL performance, although it can be very useful in high resolution gaming.

4) Recent changes in default parameters (for the preset modes)

The Ultra, Superior and Broadcast render preset parameters have changed in post-290396 builds of Vue 6 Infinite, including the new 6.10 version (build 291050). The Ultra mode has been considerably toned down, and all in all the new settings are more balanced.

Here is a quick comparison between build 290396 and build 291050:

OAA=Object AntiAliasing

TAA=Texture AntiAliasing

*** Broadcast**

290396 - OAA: 4-16 (60%) TAA 4-12 (80%)

291050 - OAA: 4-16 (60%) TAA 4-12 (**60%**)

*** Superior**

290396 - OAA: 4-24 (70%) TAA 4-24 (90%)

291050 - OAA: 4-24 (70%) TAA **4-16 (70%)**

*** Ultra**

290396 - OAA: 4-32 (85%) TAA 4-28 (95%)

291050 - OAA: **4-28 (80%)** TAA **4-20 (80%)**

And most of all, the 'Recompute Subrays' option is now unchecked by default in Ultra mode.

All this means that builds > 290396 will be considerably faster in Ultra mode, and slightly faster in Superior mode (for Broadcast it won't really be noticeable) when rendering scenes with heavy reflections / refractions. The difference in image quality will be very small for most scenes, but the speed gain can be massive.

There may be some cases - for example scenes with blurred / glossy reflections, some water scenes - where you might want to reactivate the 'Recompute subrays' option in User settings mode to get this extra quality, but of course it will have a strong negative impact on render times.

Of course it also means that render times between pre-291050 builds and post-291050 builds are not directly comparable when using these preset modes.

5) Non-linear render time increase with procedural terrains / clouds

Generally, the render time increases linearly with the resolution. For instance, a 1600 x 1200 pixels image takes about 4 times longer to render than the same image in 800 x 600 (preparation time excluded). However, it is not true for scenes with procedural objects like terrains and clouds.

Example: the 'Drakensberg' scene on the V6I Extras CD.



Here are some results (on a Dual Opteron 285) in Superior mode:

- 200 x 91 : 7s
- 400 x 182 : 29s (x 4.14)
- 800 x 365 : 146s (x 5.03)
- 1600 x 730 : 757s (x 5.18)

Here, we see that going from 400 x 182 to 1600 x 730 has multiplied the number of pixels to render by 16, but the actual render time has been multiplied by 26!

In a few words, the render time increases more than the resolution, and the higher the resolution, the bigger the difference. The explanation is: when the resolution gets higher, more and more details per pixel must be computed at render time, because of the procedural nature of the terrain and clouds (smaller details become visible and must be computed when the resolution is higher). So there is more pixels, and there is also more to do for each pixel.

It is a drawback of using procedural terrain / clouds, so one must be cautious when rendering high resolution images with such objects: it is not possible to simply extrapolate linearly the render time of a high res image from the render time of a low res version: the complexity factor must also be taken into account.

Note: this test has been done with Vue 6 Infinite 6.10 - build 291050.

6) VueMark: the standard benchmark for Vue Infinite

The VueMark (current version 1.2) is a benchmark introduced by the author of this document, in the hope of providing a standard way to measure Vue 6 Infinite rendering performance across a wide range of hardware configurations. It is composed of 3 tests, based on scenes provided on the Extras CD of V6I:

- 'Peaking Through' (outdoor, procedural terrain, metacLOUDs)
- 'RadiosityRoom' (indoor, radiosity)
- 'Cerro Verde' (outdoor, big vegetation ecosystem)



The VueMark 1.2 is much better than the 1.0 version, for the following reasons:

- It is much easier to use. The user just has to run a Python script and wait for the benchmark to finish. No need to load each scene and adjust render settings manually anymore.
- It is more accurate. It measures the total render time, including the preparation phase for radiosity (which can be quite long but is never displayed by Vue...).
- The script eliminates possible user errors (selecting the wrong resolution or render quality, etc.)

To run the VueMark, make sure that the 'Post Render Options' dialog is deactivated (it appears by default after a render), or it would stop the benchmark. Then, follow these steps:

- 1 - Download the VueMark python script -> <http://renderfred.free.fr/utis/VueMark.zip> and unzip the script.
- 2 - Create a 'VueMark' subdirectory under the '\Vue 6 Infinite\Python' directory. Put the VueMark.py script and the three benchmark scenes (01_Peaking Through.vue, 08_RadiosityRoom.vue and 05_Cerro Verde.vue) in this '\Vue 6 Infinite\Python\VueMark' directory.
- 3 - Load Vue 6 Infinite
- 4 - Run the VueMark.py script with the menu command 'Python/Run Python script...' (and browse to locate the script on your hard disk)
- 5 - Wait until the three benchmarks are finished. In the same directory, you will find a 'VueMark<32/64>_<name>.txt' file which contains the results.
- 6 - If you want to report the result, copy the content of this text file and paste it in the <http://www.cornucopia3d.com/forum/viewtopic.php?t=3344> thread at Cornucopia 3D. You will find a complete database for comparison purposes in this thread.

Note that the VueMark score is 100 for a Pentium 4C @ 3.0 GHz. A computer with a VueMark score of 200 is twice as fast as this reference computer.

7) History

1.04: July 1st, 2007

- Various sections have been updated to take into account new V6I versions, new CPUs, new OS, etc.

1.03: February 23th, 2007.

- New VueMark 1.1 (#6)
- Multithreading test (#2.3) redone with the release build of Vue 6 Infinite.

1.02: December 17th, 2006.

- Section 3.1: added a paragraph showing the minimal influence of memory speed / bandwidth on render times.
- New section 6: the VueMark benchmark.

1.01: Some style corrections. November 15th, 2006.

1.00: First version of this document. November 15th, 2006.