**INTEL MULTICORE PROCESSORS**

**HISTORY**

**COMPARISON BETWEEN ARCHITECTURES**

**DETAILED STUDY OF THE i7-6700HQ – SKYLAKE ARCHITECTURE**

**Author: Simina Radu-Alexandru**

**Group: 30433 – CTI EN**

**TA: Czako Zoltan**

**SUMMARY**

1. Introduction ….............................................................................................................. 3
2. Objective …...................................................................................................................
3. Methods …....................................................................................................................
4. Results ….......................................................................................................................
5. Conclusions …...............................................................................................................

**INTRODUCTION**

**Intel Corporation** (commonly known as **Intel** and stylized as **intel**) is an American [multinational corporation](https://en.wikipedia.org/wiki/Multinational_corporation) and [technology company](https://en.wikipedia.org/wiki/Technology_company) headquartered in [Santa Clara](https://en.wikipedia.org/wiki/Santa_Clara,_California), [California](https://en.wikipedia.org/wiki/California), in the [Silicon Valley](https://en.wikipedia.org/wiki/Silicon_Valley). It is the world's second largest and second highest valued [semiconductor chip](https://en.wikipedia.org/wiki/Semiconductor_chip) manufacturer based on revenue after being overtaken by [**Samsung Electronics**](https://en.wikipedia.org/wiki/Samsung_Electronics), and is the inventor of the [x86](https://en.wikipedia.org/wiki/X86) series of [microprocessors](https://en.wikipedia.org/wiki/Microprocessor), the processors found in most personal computers (PCs). Intel ranked No. 46 in the 2018 [Fortune 500](https://en.wikipedia.org/wiki/Fortune_500) list of the largest United States corporations by total revenue.

A **multi-core processor** is a [computer processor](https://en.wikipedia.org/wiki/Computer_processor) [integrated circuit](https://en.wikipedia.org/wiki/Integrated_circuit) with two or more separate [processing units](https://en.wikipedia.org/wiki/Central_processing_unit), called **cores**, each of which reads and executes [program instructions](https://en.wikipedia.org/wiki/Instruction_set), as if the computer had several processors. The instructions are ordinary [CPU instructions](https://en.wikipedia.org/wiki/Instruction_set) (such as add, move data, and branch) but the single processor can run instructions on separate cores at the same time, increasing overall speed for programs that support [multithreading](https://en.wikipedia.org/wiki/Multithreading_(computer_architecture)) or other [parallel computing](https://en.wikipedia.org/wiki/Parallel_computing) techniques. Manufacturers typically integrate the cores onto a single integrated circuit [die](https://en.wikipedia.org/wiki/Die_(integrated_circuit)) (known as a chip multiprocessor or CMP) or onto multiple dies in a single [chip package](https://en.wikipedia.org/wiki/Chip_carrier). The microprocessors currently used in almost all personal computers are multi-core. A multi-core processor implements [multiprocessing](https://en.wikipedia.org/wiki/Multiprocessing) in a single physical package. Designers may couple cores in a multi-core device tightly or loosely. For example, cores may or may not share [caches](https://en.wikipedia.org/wiki/CPU_cache), and they may implement [message passing](https://en.wikipedia.org/wiki/Message_passing) or [shared-memory](https://en.wikipedia.org/wiki/Shared_memory) inter-core communication methods. Common [network topologies](https://en.wikipedia.org/wiki/Network_topology) to interconnect cores include [bus](https://en.wikipedia.org/wiki/Bus_network), [ring](https://en.wikipedia.org/wiki/Ring_network), two-dimensional [mesh](https://en.wikipedia.org/wiki/Mesh_networking), and [crossbar](https://en.wikipedia.org/wiki/Crossbar_switch). Homogeneous multi-core systems include only identical cores; [heterogeneous](https://en.wikipedia.org/wiki/Heterogeneous_computing) multi-core systems have cores that are not identical. Just as with single-processor systems, cores in multi-core systems may implement architectures such as [VLIW](https://en.wikipedia.org/wiki/Very_long_instruction_word), [superscalar](https://en.wikipedia.org/wiki/Superscalar_processor), [vector](https://en.wikipedia.org/wiki/Vector_processor), or [multithreading](https://en.wikipedia.org/wiki/Multithreading_(computer_architecture)).

### **Advantages of multi-core processors**

The proximity of multiple CPU cores on the same die allows the[**cache coherency**](https://en.wikipedia.org/wiki/Cache_coherency) circuitry to operate at a much higher clock rate than what is possible if the signals have to travel off-chip. Combining equivalent CPUs on a single die significantly improves the performance of [cache snoop](https://en.wikipedia.org/wiki/Cache_snooping) (alternative: [Bus snooping](https://en.wikipedia.org/wiki/Bus_snooping)) operations. Put simply, this means that [signals](https://en.wikipedia.org/wiki/Discrete_signal) between different CPUs travel shorter distances, and therefore those signals [degrade](https://en.wikipedia.org/wiki/Degradation_(telecommunications)) less. These higher-quality signals allow more data to be sent in a given time period, since individual signals can be shorter and do not need to be repeated as often.

Assuming that the die can physically fit into the package, multi-core CPU designs require much less [printed circuit board](https://en.wikipedia.org/wiki/Printed_circuit_board) (PCB) space than do multi-chip SMP designs. Also, a dual-core processor uses slightly less power than two coupled single-core processors, principally because of the decreased power required to drive signals external to the chip. Furthermore, the cores share some circuitry, like the L2 cache and the interface to the [front-side bus](https://en.wikipedia.org/wiki/Front-side_bus) (FSB). In terms of competing technologies for the available silicon die area, multi-core design can make use of proven CPU core library designs and produce a product with lower risk of design error than devising a new wider-core design.

**Multi-core chips also allow higher performance at lower energy**. This can be a big factor in mobile devices that operate on batteries. Since each core in a multi-core CPU is generally more energy-efficient, the chip becomes more efficient than having a single large monolithic core. This allows higher performance with less energy.

**Disadvantages of multi-core processors**

Maximizing the usage of the computing resources provided by multi-core processors requires adjustments both to the [operating system](https://en.wikipedia.org/wiki/Operating_system) (OS) support and to existing application software. Also, **the ability of multi-core processors to increase application performance depends on the use of multiple threads** within applications.

Integration of a multi-core chip can lower the chip production yields. They are also more difficult to manage thermally than lower-density single-core designs. Intel has partially countered this first problem by creating its quad-core designs by combining two dual-core ones on a single die with a unified cache, hence any two working dual-core dies can be used, as opposed to producing four cores on a single die and requiring all four to work to produce a quad-core CPU. Finally, raw processing power is not the only constraint on system performance. Two processing cores sharing the same system bus and memory bandwidth limits the real-world performance advantage.

**Methods**

For this paper I have decided to follow a comparison-based system between the different architectures released over time by Intel, and specifically, to track the development of the “Intel Core i" series, the i3, i5, i7 and the newer i9, starting from their first generations. These are not the first architectures that use multiple cores on the same die, however they are very well spread, and most PCs and laptops that have Intel chips are equipped with one of these architectures.

After a side-by-side comparison of these processors, I will proceed and cover in depth the i7-6700HQ processor, which runs in my personal laptop, by stating the upgrades that were brought to this specific architecture, why I chose it and what are its limitations.



[Intel 5700HQ](https://www.notebookcheck.net/uploads/tx_nbc2/csm_4th_Generation_Intel___CoreOE_i7_Processor_Badge_f44b2eb209_07.png)

# **Intel Core i7-5700HQ**

The **Intel Core i7-5700HQ** is a fast quad-core processor based on the Broadwell architecture, which has been launched in June 2015. In addition to four CPU cores with Hyper-Threading clocked at 2.7 - 3.5 GHz (2/3/4 Cores: 3.5 GHz as well), the chip also integrates an HD Graphics 5600 GPU and a dual-channel LPDDR3-/DDR3L-1866 memory controller. The Core i7 is manufactured in a 14 nm process with FinFET transistors.

**Architecture**

Broadwell represents the "Tick" in Intel's Tick-Tock model, which means a shrink of its predecessor Haswell. The new 14 nm manufacturing process with three-dimensional FinFET transistors allows not only improvements in energy efficiency, but also a significantly smaller die size fit for more compact packages and devices.

Further advantages result from the revised microarchitecture of Broadwell. Due to its improved branch prediction, bigger buffer sizes (1500 instead of 1000 entries in the L2 TLB) and other tweaks, the performance per clock has been increased by more than 5 percent over its predecessor. There are also some new instruction set extensions designed for cryptographic applications.

**Performance**

Thanks to its improved architecture and the advanced process node (lower power consumption leads to a better utilization of the Turbo Boost), the Core i7-5700HQ offers a performance similar to the Haswell-based [Core i7-4900MQ](https://www.notebookcheck.net/Intel-Core-i7-4900MQ-Notebook-Processor.86104.0.html) (2.8 - 3.8 GHz). Thus, the CPU has sufficient power for demanding applications, modern games and intense multitasking.

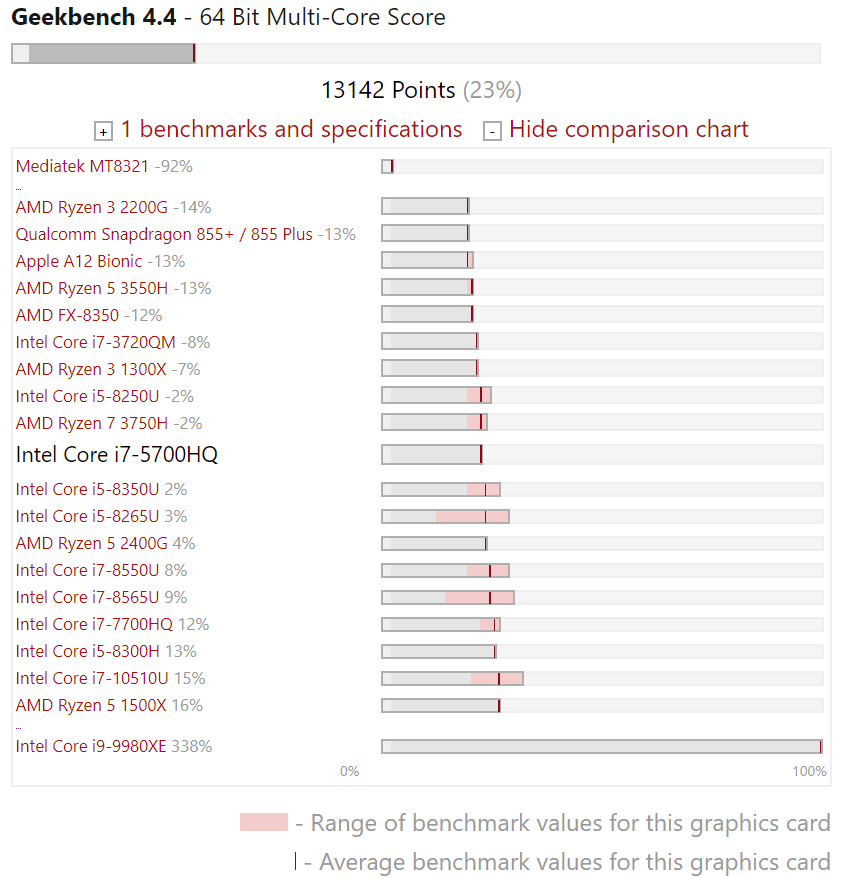
**Graphics**

The integrated [**HD Graphics 5600**](https://www.notebookcheck.net/Intel-HD-Graphics-5600.125595.0.html) offers 24 Execution Units (EUs) clocked at 300 - 1050 MHz. Similar to the CPU core, the GPU architecture (Intel Gen 8) has been thoroughly revised for improved performance-per-clock. The HD Graphics 5600 beats the previous [HD Graphics 4600](https://www.notebookcheck.net/Intel-HD-Graphics-4600.86106.0.html) (20 EUs) by about 20 percent, but is still significantly slower than dedicated low-end GPUs like the [GeForce 920M](https://www.notebookcheck.net/NVIDIA-GeForce-920M.138763.0.html). Most games as of 2015 will be playable only in (very) low settings.

Broadwell is the first Intel chip to fully support DirectX 11.2 as well as OpenCL 1.3/2.0 and OpenGL 4.3. Video will output natively via DisplayPort 1.2 or HDMI 1.4. However, the new HDMI 2.0 standard is not supported.

**Power Consumption**

The i7-5700HQ is rated at a TDP of 47 W including the graphics card, memory controller and VRMs. Thus, the CPU is best suited for laptops 15-inches or greater. If necessary, the TDP can be lowered to 37 W (cTDP down).



[Intel 6700HQ](https://www.notebookcheck.net/uploads/tx_nbc2/corei7_01.png)

# **Intel Core i7-6700HQ**

The **Intel Core i7-6700HQ** is a quad-core processor based on the Skylake architecture, that has been launched in September 2015. In addition to four CPU cores with Hyper-Threading clocked at 2.6 - 3.5 GHz (4 cores: max. 3.1 GHz, 2 cores: max. 3.3 GHz), the chip also integrates an HD Graphics 530 GPU and a dual-channel DDR4-2133/DDR3L-1600 memory controller. The CPU is manufactured using a 14 nm process with FinFET transistors.

**Architecture**

Skylake replaces both Haswell and Broadwell and brings the same microarchitecture in every TDP class from 4.5 to 45 W. The extensive improvements of the Skylake design include increased out-of-order buffers, optimized prefetching and branch prediction as well as additional performance gains through Hyper-Threading. Overall, however, performance per clock has been increased by only 5 to 10 percent (compared to Haswell) respectively under 5 percent (compard to Broadwell), which is quite modest for a new architecture ("Tock").

**Performance**

According to the specified clock rates and the improved architecture, the Core i7-6700HQ performs roughly on par with the former, somewhat higher clocked [Core i7-4900MQ](https://www.notebookcheck.net/Intel-Core-i7-4900MQ-Notebook-Processor.86104.0.html) or [i7-4810MQ](https://www.notebookcheck.net/Intel-Core-i7-4810MQ-Notebook-Processor.109765.0.html) (Haswell). Under long-lasting full load, the Skylake chip will take the lead over Haswell-based predecessors, as the advanced 14 nm process leads to a higher energy efficiency and reduces throttling.

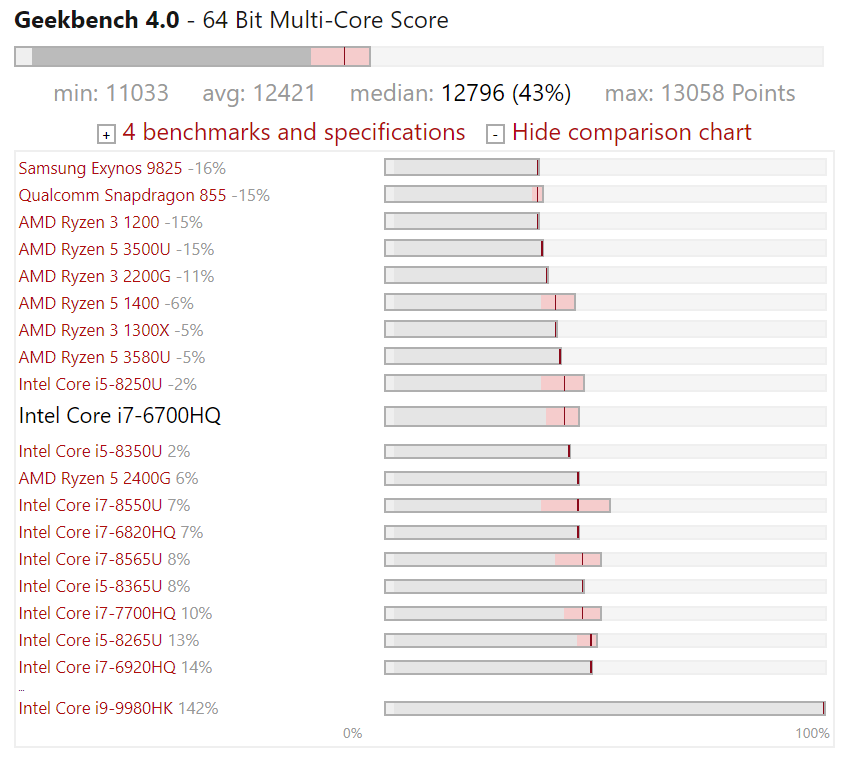
Even the most demanding applications and excessive multitasking are handled easily.

**Graphics**

The integrated graphics unit called [HD Graphics 530](https://www.notebookcheck.net/Intel-HD-Graphics-530.148358.0.html) represents the "GT2" version of the Skylake GPU (Intel Gen. 9). The 24 Execution Units, also called EUs, are clocked at 350 - 1050 MHz and offer a performance about 20 percent above the old [HD Graphics 4600](https://www.notebookcheck.net/Intel-HD-Graphics-4600.86106.0.html). Games of 2015 can thus be played smoothly in low or medium settings. For more information about performance and features, check our page for [HD Graphics 530](https://www.notebookcheck.net/Intel-HD-Graphics-530.148358.0.html).

**Power Consumption**

Specified at a TDP of 45 W (including CPU, GPU and memory controller), the CPU is best suited for bigger notebooks 15-inches in size and above. Optionally, the TDP can be lowered to 35 watts (cTDP down), reducing both heat dissipation and performance.



# [Intel 7700HQ](https://www.notebookcheck.net/uploads/tx_nbc2/corei7_09.png)**Intel Core i7-7700HQ**

The **Intel Core i7-7700HQ** is a fast quad-core processor for notebooks based on the Kaby Lake H architecture (7th generation Core), which was announced in January 2017 at CES. It is the successor to the [Core i7-6700HQ from the Skylake generation](https://www.notebookcheck.net/Intel-Core-i7-6700HQ-Notebook-Processor.149420.0.html) and is manufactured in an improved 14 nm+ process, so the clocks are 200 MHz higher at the same TDP. The architecture was not changed, only the video engine got an update (see our [Kaby Lake article](https://www.notebookcheck.net/Intel-Kaby-Lake-All-Details-and-Information-about-the-Launch-of-the-7th-Processor-Generation.172693.0.html" \t "_self)).

The integrated graphics card is called [Intel HD Graphics 630](https://www.notebookcheck.net/Intel-HD-Graphics-630.187948.0.html), but the architecture does not differ from the 530 GPU from the Skylake generation and only the clocks are slightly higher.

**Performance**

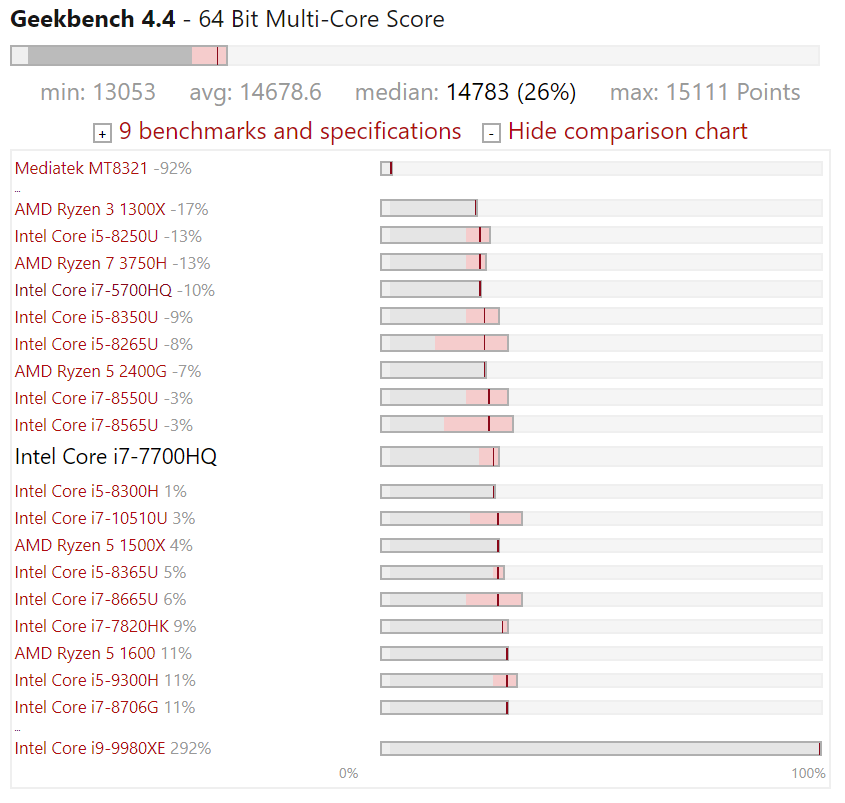
Thanks to the 200 MHz higher clocks (5.5-7.6% depending on the Boost), the CPU performance is increased and roughly on par with the [Core i7-6970HQ (2.8-3.7 GHz but with 128 MB eDRAM)](https://www.notebookcheck.net/Intel-Core-i7-6970HQ-Notebook-Processor.164967.0.html). The TDP can also be reduced to 35 Watts (cTDP down), but this will reduce the performance.

**Graphics**

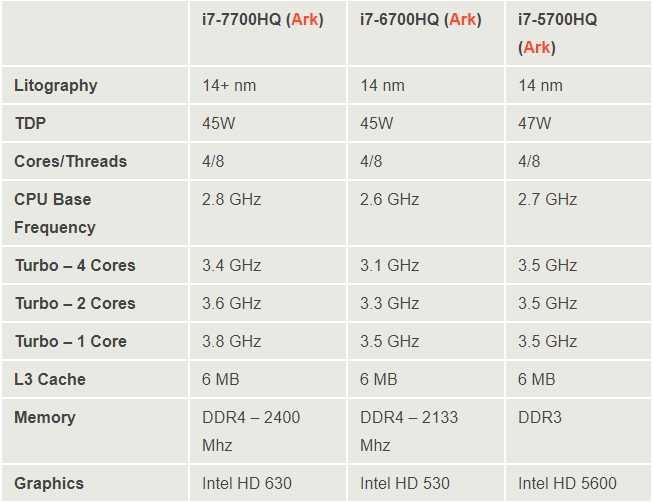
The integrated graphics unit called [HD Graphics 530](https://www.notebookcheck.net/Intel-HD-Graphics-530.148358.0.html) represents the "GT2" version of the Skylake GPU (Intel Gen. 9). The 24 Execution Units, also called EUs, are clocked at 350 - 1050 MHz and offer a performance about 20 percent above the old [HD Graphics 4600](https://www.notebookcheck.net/Intel-HD-Graphics-4600.86106.0.html). Games of 2015 can thus be played smoothly in low or medium settings. For more information about performance and features, check our page for [HD Graphics 530](https://www.notebookcheck.net/Intel-HD-Graphics-530.148358.0.html).

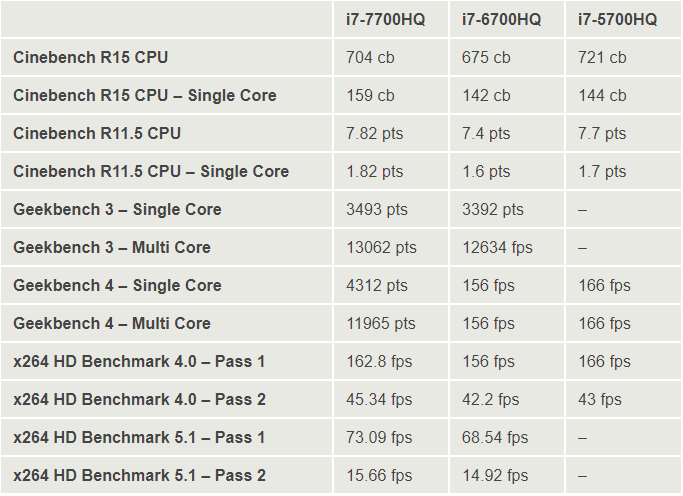
**Power Consumption**

Due to its 45-Watt TDP, the CPU will be used in bigger notebooks with at least 15 inches most of the time.



# **Intel Core i7-7700HQ vs Core i7-6700HQ vs i7-5700HQ**





**Implementation**

**CoreInfo Class**

**●** containing the information about the core in use, its running times, and the number of operations performed on that core.

**●** overrides the **toString()** method for the output of the results using a StringBuilder.

public String name = null;  
public Date startTime = null;  
public Date endTime = null;  
public int numberOfRecords = 0;

**MultiCoreProcessor Class**

**●** the **getListOfSublistsForCores** receives as an argument a list, then devides it in subLists in such a manner that each processor will have the same workload (the minBatchSize). The remained, unassigned work will be spread between the other cores, randomly. The method will return the list of subLists assigned to each separate core.

private static List<List<Object>> getListOfSubListsForCores(List lst) {/\* \*/ }

**●** the **callConsumer** method receives as parameters a list, which is in fact the list of all operations that have to be performed by the entire CPU, and also a Consumer<Object> which will represent the function that has to be performed by the appropriate core. Now, we call the **getListOfSublistsForCores()**  to obtain the operations required to be performed by each core, we turn the **List<List<Object>>** into a parallelStream with the purpose of applying the accept method to each list of operations that have to be performed on the specified core. During the testing, this accept method will be overridden, in order to check specifically what we are interested in.

public static void callConsumer(List lst, Consumer<Object> c) {  
 *getListOfSubListsForCores*(lst).parallelStream()

.forEach(subList -> c.accept(subList));  
}

The smart thing here is the parallelStream(). The Stream API enables developers to create the parallel streams that can take advantage of multi-core architectures and enhance the performance of Java code. In a parallel stream, the operations are executed in parallel and there are two ways to create a parallel stream. There are some scenarios where parallel processing does not necessarily means faster. The allocation of resources to multiple threads and the final step of combining the results will also introduce a degree of overhead. In case of Parallel stream,4 threads are spawned simultaneously and it internally using Fork and Join pool to create and manage threads.Parallel streams create ForkJoinPool instance via static ForkJoinPool.commonPool() method. Parallel Stream takes benefits of all available CPU cores and processes the tasks in parallel. If number of tasks exceeds the number of cores, then remaining tasks wait for currently running task to complete.

**●** the **callUnparallelConsumer()** method does the same exact thing as the previous method, but replaces the parallelStream with a regular stream. This will cause the code to run sequentially, instead of in parallel, on a single core. In our example this will outcome in much worse results, but still, this kind of approach is viable since its overhead is much smaller than the one from the parallel solution. Moreover, there are situations, problems, where one would not benefit from the processing power of multiple cores/threads, so the parallel approach would be just a waste of resources.

public static void callUnparallelConsumer(List lst, Consumer<Object> c) {  
 *getListOfSubListsForCores*(lst).stream()

.forEach(subList -> c.accept(subList));  
}

**TEST CLASSES**

**MultiCore\_Consumer\_AnonymousClassTest**

**●** in this test class we have a map linking the core name with its coreInfo and also a **count** which will be a shared variable between the multiple cores

**●** the **test()** method will firstly generate the list required for spreading the tasks equally between the cores. Then we call the **callConsumer()**  method on the new list and we override the accept method.

MultiCoreProcessor.*callConsumer*(lst, new Consumer<Object>() {//pass list & function to multi core processor..  
 @Override  
 public void accept(Object o) {

/\*

\*/

}

After that each core, with its attributed list of operations will compute the sum of all indexes of the operations performed by it then, add this sum to the synchronized-shared **count** variable.

int sum = 0;  
for (Integer i : lst) {  
 sum += i;  
}  
synchronized (*count*) {  
 *count* += sum;  
}

Then we make a long lasting operation (the tangent of the first 100 million natural numbers.

for (int j = 0; j < 100000000; j++) {  
 Math.*tan*(j);//some operation that takes long time  
}

We get the name of the core executing the current operation, map it to the coreInfo, and we update the start and endTime for the current process.

String core = Thread.*currentThread*().getName();

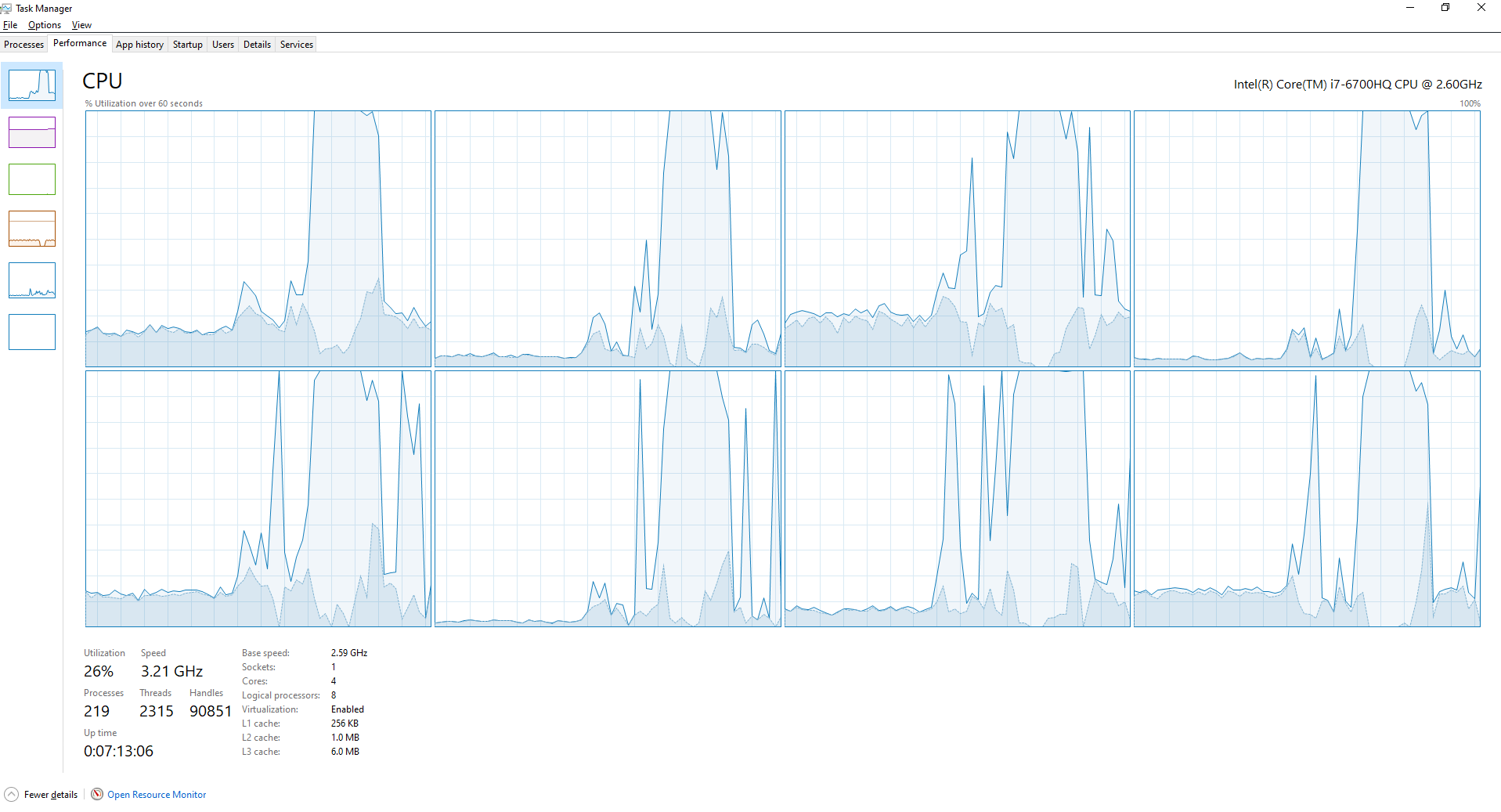
**SingleCore\_Consumer\_AnonymousClassTest**

**●** here we implemented the test case for a single core application, running the workload, previously handled by all the 8 logical cores of the i7-6700HQ.

Aside from that, the code is identcal, but instead of calling the callConsumers method we call the callUnparallelConsumer method, with the purpose of using a simple stream instead of a parallel one, such that the application will be run alternatively on the cores, but never in parallel.

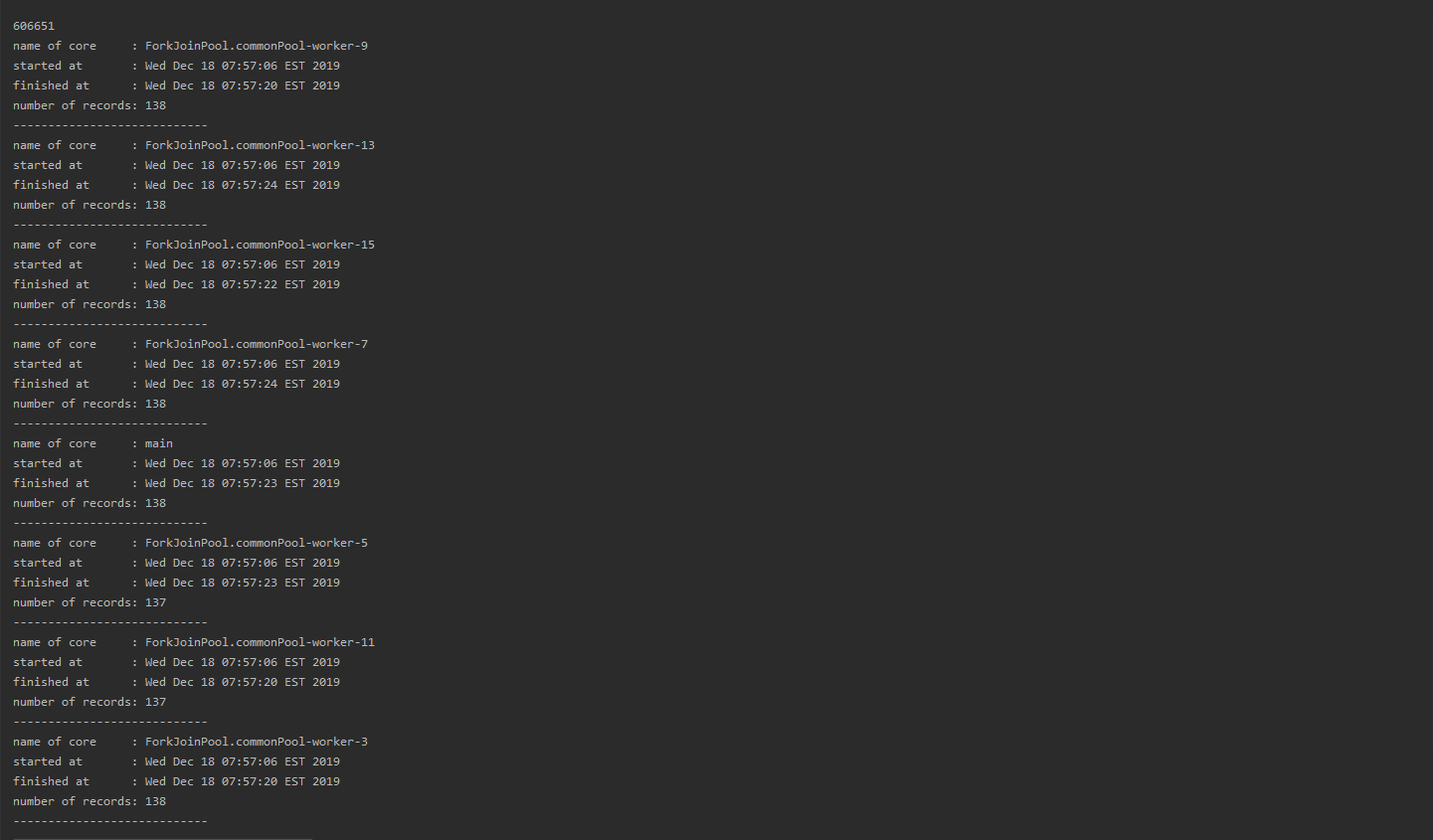
**RESULTS OF THE TESTING IN THE TWO CASES – PARALLEL / SEQUENTIAL EXECUTION**

**Parallel execution:**

****

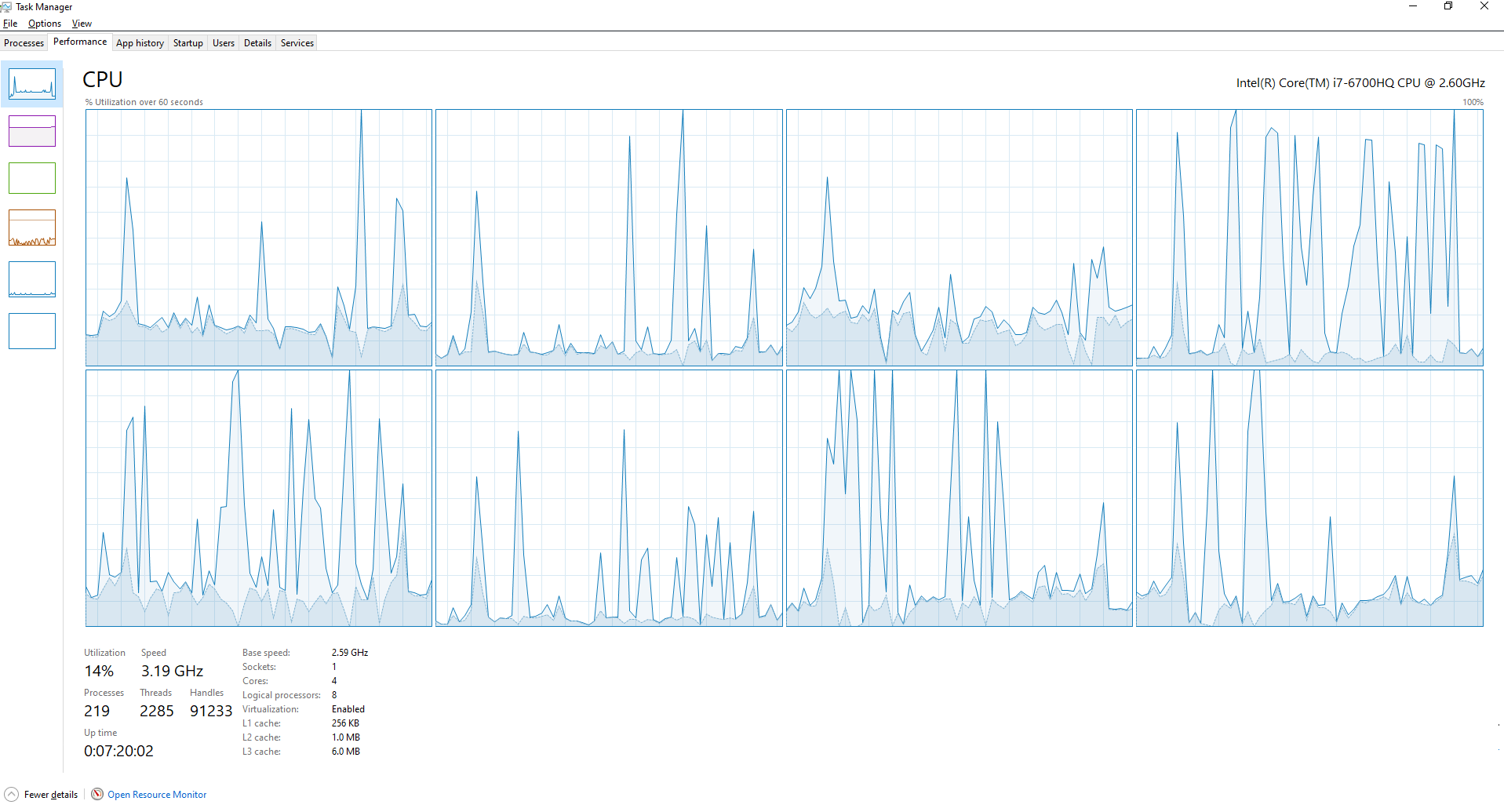
Here it is easy to notice the concurrent run of the 8 logical cores, in the same time, during the execution of our program.

The workload is equally distributed on all cores, such that, the total execution time is reduced, proving the efficiency of multicore processing in this particular case.



We can observe now that the sum of indexes of all the performed operations on the 8 cores is 606651, while each single core has performed roughly the same amount of operations: 137/138. Moreover, one could notice that the active time for each core is reduced, especially compared to the sequential execution of the same operations. Each core has its unique id.

**Sequential Execution:**

****

Now, here it is a little bit harder to notice the shared execution between processors. However, at a closer look one could observe that the spikes in CPU usage appear at different time intervals, which means that at a certain time, only one CPU will perform the process, but, because of OS optimizations, the CPU responsible for handling the execution of our code will be switched between the available one every period of time.

Moreover, it is to notice the fact that, comparing the 2 graphs presented, the execution of the code in a multi threaded manner will take roughly 1/8 of the time it takes for a single threaded process to end its execution.



In the case of sequential execution, all operations are performed on the main core. The sum of all indexes is the same as the previous, proving that the processors perform the same operations. However, the execution time has increased significantly, rising approximately 8 times, which comes in support to the affirmation that 8 cores should un 8 time faster than a single core, but then again this is just a particular case.

Now the idea is that, although, as we can see, the main core is responsible with the execution, it splits its execution on multiple cores, but sequentially, because of the OS. That is why one does not see just one core running continuously ,but rather multiple ones, taking turns in the execution of the program.

**CONCLUSIONS**

Intel is one, if not the biggest player on the market in regards of single and multi-core processors. Together with AMD, they have been dominating the market, proposing very interesting and powerful chips.

The multi core architecture is a big upgrade, providing substantial performance boost while having to deal with big workloads. Although the technology is not perfect yet, it is getting close and the future is bright in this domain. We already have to deal with chips that have up to 64 physical cores and 128 logical cores, due to hyper-threading.

Regarding the architecture that my machine runs on, the i7-6700 HQ is a solid mobile chipset, that posses 4 physical cores and 8 logical, that can boost up to 3.5 GHz, with a base clock of 2.6 GHz. Not that powerful, but somehow power efficient, the CPU performs well under medium-high workloads. During heavy workloads, it has problems dealing with heath dissipation, and for that reason, it may throttle, lowering the chances of maintaining a constant boost speed.

The tests that I have applied to my machine prove these things. While performing pretty well during heavy workload for a short period of time, the temperatures start to rise pretty fast, hence I assume that during longer periods of operating it will start to throttle.

**BIBLIOGRAPHY**

1. Rouse, Margaret (March 27, 2007). [*"Definition: multi-core processor"*](http://searchdatacenter.techtarget.com/sDefinition/0,,sid80_gci1015740,00.html). TechTarget. [*Archived*](https://web.archive.org/web/20100805052158/http:/searchdatacenter.techtarget.com/sDefinition/0,,sid80_gci1015740,00.html) from the original on August 5, 2010*. Retrieved March 6, 2013*.
2. Schauer, Bryan. [*"Multicore Processors – A Necessity"*](https://web.archive.org/web/20111125035151/http:/www.csa.com/discoveryguides/multicore/review.pdf) *(PDF)*. Archived from [*the original*](http://www.csa.com/discoveryguides/multicore/review.pdf) *(PDF)* on 2011-11-25.
3. Suleman, Aater (May 20, 2011). [*"What makes parallel programming hard?"*](https://web.archive.org/web/20110529133159/http:/www.futurechips.org/tips-for-power-coders/parallel-programming.html). FutureChips. Archived from [*the original*](http://www.futurechips.org/tips-for-power-coders/parallel-programming.html)on May 29, 2011*. Retrieved March 6, 2013*.
4. Schor, David. [*"The 2,048-core PEZY-SC2 sets a Green500 record"*](https://fuse.wikichip.org/news/191/the-2048-core-pezy-sc2-sets-a-green500-record/). WikiChip.
5. Vajda, András. [*Programming Many-Core Chips*](https://books.google.com/books?id=pSxa_anfiG0C&pg=PA3&dq=several+tens). Springer. p. 3. [*ISBN*](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [*978-1-4419-9739-5*](https://en.wikipedia.org/wiki/Special:BookSources/978-1-4419-9739-5).
6. Shrout, Ryan (December 2, 2009). [*"Intel Shows 48-core x86 Processor as Single-chip Cloud Computer"*](http://www.pcper.com/reviews/Processors/Intel-Shows-48-core-x86-Processor-Single-chip-Cloud-Computer). [*Archived*](https://web.archive.org/web/20160105080008/http:/www.pcper.com/reviews/Processors/Intel-Shows-48-core-x86-Processor-Single-chip-Cloud-Computer) from the original on January 5, 2016*. Retrieved May 17, 2015*.
7. [*"Intel unveils 48-core cloud computing silicon chip"*](http://news.bbc.co.uk/2/hi/technology/8392392.stm). BBC. December 3, 2009. [*Archived*](https://web.archive.org/web/20121206054225/http:/news.bbc.co.uk/2/hi/technology/8392392.stm) from the Original on December 6, 2012*. Retrieved March 6, 2013*.