



TIMi suite
integrated data mining solutions

An automated predictive datamining tool

Server/Infrastructure Selection for TIMi v1.09

Creation Date: September 2012

Last Edited: October 2018

Introduction

This document is a guide on how to select a good hardware infrastructure for TIMi.

The optimal hardware infrastructure for TIMi is composed of several PC's (laptops & servers) interconnected together. Section 1 gives advices on which PC to buy to run TIMi efficiently while Section 2 explains the different roles of each PC in the global infrastructure. The next section (Section 3), explain how this infrastructure integrates with other tools such as Hadoop, Ordinary BI tools (Kibella, Tableau, Qlick, etc.) and Jenkins (Jenkins is a scheduler that allows to run TIMi-based batch job automatically each day, week, month). Finally, the last section (Section 4) summarizes this document in a global info-graphic.

1. How to select a good machine to run TIMi?

1.1. Minimum Requirements

Minimum system Requirements for TIMi installation on a server or workstation are:

OS: Windows 2000, XP, Vista, Win7, Windows8, Win10

Strongly advised: No virtual machine: By default, TIMi won't run in a VMWare virtual Machine or equivalent software. If you need to run TIMi in a VMWare virtual Machine, please request a Network License (the Network License is free when you buy a Timi license).

RAM: Minimum 2GB per simultaneous user. Large data-transformations require more RAM.

Video Card: To use StarDust, you need a 3D-hardware-accelerated-graphical card that supports OpenGL2.0 (any computer produced after 2007 will do the trick). The specific OpenGL drivers from you video card manufacturer must be installed (do not use "generic" OpenGL drivers).

For increased processing speed, we strongly suggest a multi-core CPU (TIM is 100% mutli-threaded). A server with an "Intel Core I7 at 3.0 GHz (or above)" CPU is always a good option. To process large volumes of data with Anatella (to do "Big Data" analytics), you need a 64-bit OS and plenty of RAM (16GB or 32GB).

1.1.1. Minimum requirement: Anatella.

Although Anatella (our ETL) is able of achieving 99% of the required data transformations on "simple commodity PC's" (typically we are using the simple laptops from the dataminers), it is nevertheless advisable to use a server with a large quantity of RAM (16 GB or 32GB) for transformations involving tables containing tens of billions of rows.

1.1.2. Minimum requirement: Timi Modeler.

Timi Modeler is the only “Machine Learning tool” that is 100% multi-threaded. In other words: TIMi Modeler performs its computations using all the available CPUs on the server. This means that the computations made by a user who is “alone” on a quad-core server will be four times faster (approximately) than when there are 4 users working simultaneously on the server. In contrast, other datamining softwares perform all their computation using a single CPU. This means that when a user is “alone” on a quad-core server, the computation time (of the software competitors) is the same as when there are 4 users on the server (because 3 of the 4 CPUs remain unused).

This unique feature of Timi Modeler affects the QoS (“Quality of Service”) provided by Timi Modeler: a limited number of users on each server greatly improves the computation speed and therefore the QoS. Therefore, a larger number of TIMi servers (and CPU’s) provide a higher QoS. The purchase of additional “TIMi” servers is nearly always justified (to improve the QoS).

On a given hardware, the computation time of Timi Modeler mainly depends on the size of the analyzed datasets. When handling large datasets, computation time increases. When working on very large datasets, to obtain a satisfactory QoS (i.e. a computation time that is reasonable), it’s necessary to provide more CPU resources to the TIMi users.

Here is a table that summarizes the situation:

Dataset Size	Minimum Required RAM per concurrent user for TIMi modeler	Minimum Required CPU Resources For TIMi Modeler
Large	2 GB	from 0.4 to 8 CPU’s per concurrent user (ideally 1 CPU/ concurrent user)
		On a quadcore server: 1 to 10 concurrent users (for a good QoS: 4 simultaneous users)
Small (less than 1 MB)	100 MB	from 1 to 110 simultaneous users on a quadcore server

The recommendations given the table above reflects the fact that the Timi Modeler users are usually analyzing datasets at the Gigabyte size (common volumes are: ¼, ½, 1, 2, 5, 10, 20 GB), which is a common situation for data mining analysis, yet quite exceptional for “old” statistical packages... To ensure a greater modeling accuracy (and therefore a higher ROI), TIMi practically never does any sampling and always work on the “full” data set (thanks to its unique compression algorithm, TIMi can store in internal RAM datasets of several dozen gigabytes). This approach is more costly in CPU and RAM but consistently deliver superior models and thus a higher ROI.

1.2. Introduction to the Optimal hardware selection for TIMi

The objective of this document is to help you select the best server for an efficient working environment with **The Intelligent Mining Machine (TIMi)**. The TIMi software solution contains 4 tools: Anatella, TIMi Modeler, Stardust and Kibella.

Fortunately, these four tools share the same needs in terms of hardware. More precisely, the main bottleneck (that slows down all computations) when computing some results with Anatella, TIMi, Stardust or Kibella is nearly always the CPU.



The main limiting factor in terms of processing speed for Anatella is usually not the hard drive (as it's the case with other ETL's) but rather the processor (i.e. the CPU) that performs the computations. Since "simple modern laptops" are now usually provided with rather slow hard drive but with good processors (i.e. they are equipped with Intel Core i7 and Intel Core i5 3GHz processors). These "simple laptops" are good candidates to run Anatella.

Thanks to its unique data compression technology, Anatella reduces to the minimum the bandwidth used on your corporate computer network. So there is no reason to let idle the good processors on your dataminer's computers.

Some of our clients have done several benchmarks that demonstrates that Anatella installed on one simple laptop equipped with a CoreI7 CPU and 32GB RAM is approximately three times faster than a "1 million euro" Oracle Exadata solution. Thus, the total available "computing power" that you get when installing Anatella on each&every dataminer laptop is **worth several millions euros when purchased from another vendor**.

To Summarize: The possibility to use Anatella directly on the dataminers laptops allows you to provide to your dataminer team a very comfortable working environment (i.e. it allows you to achieve a very high QoS = "Quality of Service") and a tremendous computing power.

In an attempt to better extract all the power available inside the CPU (since the CPU is the main bottleneck), we created tools that are heavily multithreaded (i.e. under some specific conditions the tools might use several CPU cores simultaneously). Despite using inside our tool the most advanced multithreading techniques (e.g. lock-free code), we still advise our customer to buy CPU that have the best speed on ONE core (i.e. single-core execution) because of the large overhead that multithreading and parallel computations always exhibit.

Let's give an example: 99% of the time, it's more efficient (i.e. faster) to run all the computations on one core on a (fast) "Intel Core I7 at 4GHz" (4GHz is the speed of the CPU) than to run the same computation using 3 cores/CPU on a (slower) "Intel Core I7 at 2GHz" (because of the multithreading overhead).

To summarize, when buying some hardware to run TIMi & Anatella, you should buy a machine with a fast CPU (more precisely, you should pay attention to the "single-core execution" speed of the CPU): The objective of this section is to help you on this process.

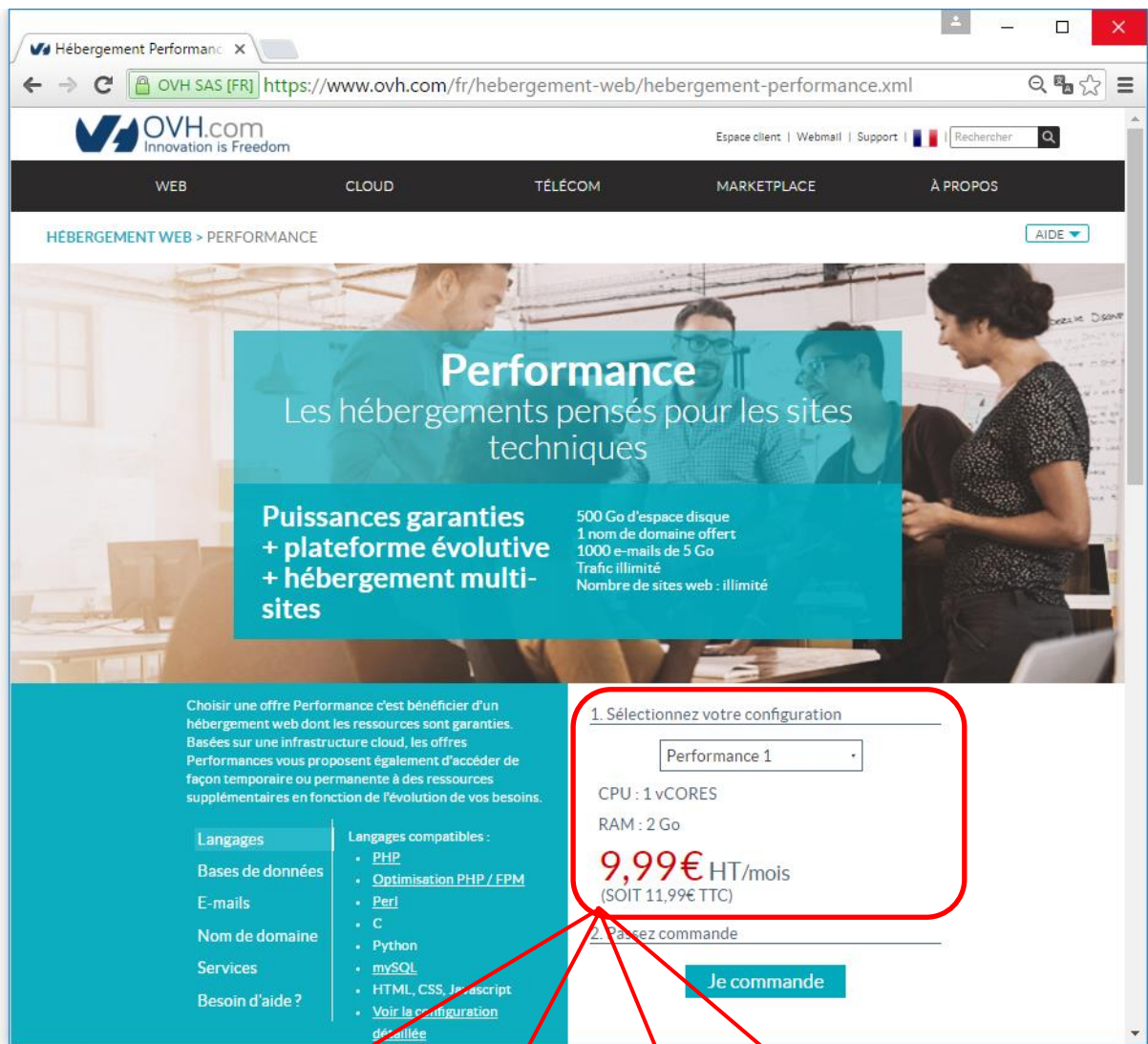
Most of the time, one of the best solution is simply to give to each of the analysts good "Core I7" laptops. In this way, each analyst has the usage of 100% of its own powerful Core 17 CPU: for more information about this subject/infrastructure, see the section 2.1 here below.

To help you select the right CPU for your server, we advise you to use the "geek benchmark" website that list the ("single-core") performances of all current CPU's.

1.3. About common “Big” Servers in Data Centers

Unfortunately, currently, most “big servers” in large data centers are optimized to be “web servers”. Such type of “big servers” possess very efficient hard drives but always some VERY BAD cpu. Thus, if you decided to install TIMi inside a data center, it most certainly means that you selected a VERY BAD machine for TIMi.

Basically, it's always the same thing: The owners of the "data centers" want to tell their customers that they will have their own "core/CPU" dedicated for themselves only. Here is a good example:



The screenshot shows the OVH.com website for Performance hosting. A red box highlights the configuration selection area for 'Performance 1'.

Configuration	CPU	RAM	Price (HT)	Price (TTC)
Performance 1	1 vCORES	2 Go	9,99€	11,99€
Performance 2	2 vCORES	4 Go	18,99€	22,79€
Performance 3	3 vCORES	6 Go	26,99€	32,39€
Performance 4	4 vCORES	8 Go	33,99€	40,79€

1. Sélectionnez votre configuration

Performance 1

CPU : 1 vCORES

RAM : 2 Go

9,99€ HT/mois
(SOIT 11,99€ TTC)

1. Sélectionnez votre configuration

Performance 2

CPU : 2 vCORES

RAM : 4 Go

18,99€ HT/mois
(SOIT 22,79€ TTC)

1. Sélectionnez votre configuration

Performance 3

CPU : 3 vCORES

RAM : 6 Go

26,99€ HT/mois
(SOIT 32,39€ TTC)

1. Sélectionnez votre configuration

Performance 4

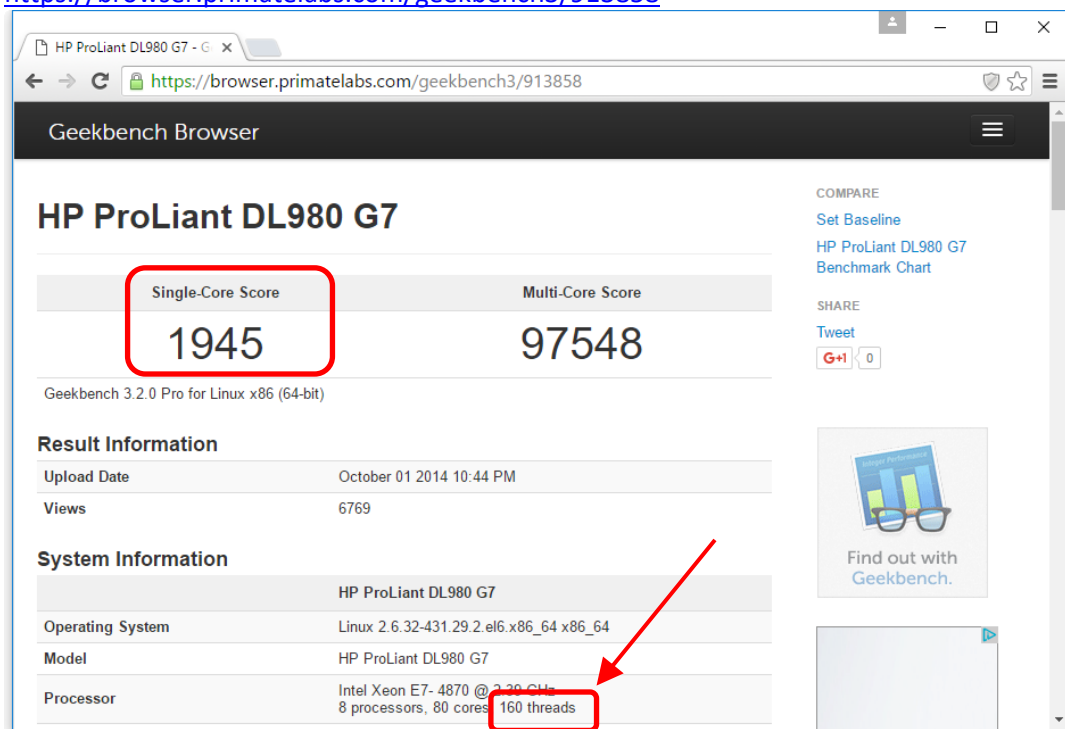
CPU : 4 vCORES

RAM : 8 Go

33,99€ HT/mois
(SOIT 40,79€ TTC)

To be able to provide to all their customers with this really large number of cores, the data center’s owners are buying (or building) servers such as this one:

<https://browser.primatelabs.com/geekbench3/913858>



Please note in the above screenshot the very low “single-core” Score: 1945 (less than 2000: pitiful!).

This means that the speed of each of the (160) vcores/threads inside this server is really slow (but there are many of them!). To summarize, the data center shows you an advertisement that guarantees to you your own, private vcore but nobody said that this vcore will be fast! 🤪 You must also realize that, very often, the situation is even worse than the above screenshot: The CPU’s inside common servers in data centers are, most of the time, far far worse.

What’s happening if you install Anatella/TIMi on a server such as the “HP ProLiant” described here above? Everything will be very, very slow compared to a small, inexpensive “stupid” Core I7 laptop. The only way to get this (extremely expensive) “HP ProLiant” running a little bit faster than the “stupid” Core I7 laptop would be to use many cores (i.e. more than 8) at the same time (since this server has 160 vcores/threads compared to a simple “Core I7” laptop that still has already 8 vcores/threads). For example, you could try to run many graphs at the same time (at least more than 8). Unfortunately, in reality, this situation mostly never happens: 99% of the time, the order in which the graphs are executed is just Sequential: one graph than another graph and so on... (i.e. most of the time, you run one graph at-a-time).

1.4. About XEON processors

The “professional, high-grade” CPU’s for data centers are, most of time, XEON processors (and not Core I7 processors). What to think about XEON processors in general? Let’s look at the GeekBench website to have an answer (the table below has been extracted on the 2018/4/30 from the page <https://browser.primatelabs.com/processor-benchmarks>) :



The best CPU on the 2018/4/30 for Anatella/TiMi

#	CPU	Frequency	64-bit Single Core Perf.
1	Intel Core i7-8700K	3.7 GHz (6 cores)	5928
2	Intel Core i7-7700K	4.2 GHz (4 cores)	5703
3	Intel Core i5-8600K	3.6 GHz (6 cores)	5659
4	Intel Core i3-7350K	4.2 GHz (2 cores)	5650
5	Intel Core i3-8350K	4.0 GHz (4 cores)	5481
6	Intel Core i9-7960X	2.8 GHz (16 cores)	5429
7	Intel Core i5-7600K	3.8 GHz (4 cores)	5371
8	Intel Core i9-7920X	2.9 GHz (12 cores)	5370
9	Intel Core i7-7820X	3.6 GHz (8 cores)	5369
10	Intel Core i9-7900X	3.3 GHz (10 cores)	5352
11	Intel Core i7-6700K	4.0 GHz (4 cores)	5338
12	Intel Core i7-8700	3.2 GHz (6 cores)	5338
13	Intel Core i7-7800X	3.5 GHz (6 cores)	5274
14	Intel Xeon W-2150B	3.0 GHz (10 cores)	5206
15	Intel Core i9-7980XE	2.6 GHz (18 cores)	5203
16	Intel Core i9-7940X	3.1 GHz (14 cores)	5193
17	Intel Core i5-6600K	3.5 GHz (4 cores)	5101
18	Intel Xeon E3-1275 v6	3.8 GHz (4 cores)	4966
19	Intel Core i7-4790K	4.0 GHz (4 cores)	4900
20	Intel Core i5-8400	2.8 GHz (6 cores)	4882
21	Intel Core i7-5775C	3.3 GHz (4 cores)	4869
22	AMD Ryzen 7 2700X	3.7 GHz (8 cores)	4867
23	Intel Core i5-7600	3.5 GHz (4 cores)	4855
24	Intel Xeon E3-1535M v6	3.1 GHz (4 cores)	4853
25	Intel Core i7-7700	3.6 GHz (4 cores)	4848
26	Intel Core i5-5675C	3.1 GHz (4 cores)	4841
27	Intel Xeon W-2140B	3.2 GHz (8 cores)	4796
28	Intel Core i3-7300	4.0 GHz (2 cores)	4789
29	AMD Ryzen 5 2600X	3.6 GHz (6 cores)	4758
30	Intel Core i7-7820HK	2.9 GHz (4 cores)	4694
31	Intel Xeon E3-1245 v6	3.7 GHz (4 cores)	4651
32	Intel Xeon E3-1270 v6	3.8 GHz (4 cores)	4631
33	Intel Xeon E3-1505M v6	3.0 GHz (4 cores)	4615
34	Intel Core i5-4690K	3.5 GHz (4 cores)	4582
35	Intel Core i3-6320	3.9 GHz (2 cores)	4581
36	Intel Core i7-6850K	3.6 GHz (6 cores)	4580
37	Intel Core i7-6900K	3.2 GHz (8 cores)	4572
38	Intel Core i7-7567U	3.5 GHz (2 cores)	4564
39	Intel Xeon E3-1270 v5	3.6 GHz (4 cores)	4553
40	Intel Core i7-4770K	3.5 GHz (4 cores)	4534
41	Intel Core i7-6950X	3.0 GHz (10 cores)	4524
42	Intel Core i7-6700	3.4 GHz (4 cores)	4517
43	Intel Xeon E3-1275 v5	3.6 GHz (4 cores)	4502
44	Intel Core i7-5960X	3.0 GHz (8 cores)	4500
45	Intel Core i7-5930K	3.5 GHz (6 cores)	4496
46	Intel Core i5-4670K	3.4 GHz (4 cores)	4483
47	Intel Core i7-5775R	3.3 GHz (4 cores)	4476
48	Intel Core i7-8650U	1.9 GHz (4 cores)	4473
49	Intel Xeon E3-1575M v5	3.0 GHz (4 cores)	4472
50	Intel Xeon E3-1260L v5	2.9 GHz (4 cores)	4462
51	Intel Xeon E3-1545M v5	2.9 GHz (4 cores)	4433
52	Intel Core i5-7500	3.4 GHz (4 cores)	4427
53	Intel Core i7-7700T	2.9 GHz (4 cores)	4427
54	Intel Xeon E3-1275 v3	3.5 GHz (4 cores)	4425
55	Intel Core i3-6300	3.8 GHz (2 cores)	4422
56	Intel Core i7-6800K	3.4 GHz (6 cores)	4420
57	Intel Xeon E3-1245 v5	3.5 GHz (4 cores)	4420
58	Intel Core i3-7100	3.9 GHz (2 cores)	4412
59	Intel Xeon E3-1240 v5	3.5 GHz (4 cores)	4405
60	Intel Core i3-8100	3.6 GHz (4 cores)	4403
61	Intel Core i5-6600	3.3 GHz (4 cores)	4397
62	AMD Ryzen Threadripper 1950X	3.4 GHz (16 cores)	4386
63	Intel Core i7-4980HQ	2.8 GHz (4 cores)	4373
64	Intel Core i7-6820HK	2.7 GHz (4 cores)	4366
65	Intel Xeon E3-1285L v3	3.1 GHz (4 cores)	4355
66	Intel Core i7-5820K	3.3 GHz (6 cores)	4354
67	Intel Xeon E3-1230 v5	3.4 GHz (4 cores)	4326
68	Intel Xeon E3-1276 v3	3.6 GHz (4 cores)	4320
69	Intel Core i7-6920HQ	2.9 GHz (4 cores)	4314
70	AMD Ryzen Threadripper 1900X	3.8 GHz (8 cores)	4303
71	Intel Xeon E3-1246 v3	3.5 GHz (4 cores)	4283
72	Intel Core i7-7660U	2.5 GHz (2 cores)	4283
73	Intel Core i7-4940MX	3.1 GHz (4 cores)	4281
74	AMD Ryzen 7 1800X	3.6 GHz (8 cores)	4275
75	Intel Core i7-7920HQ	3.1 GHz (4 cores)	4273
76	Intel Core i7-4790	3.6 GHz (4 cores)	4267
77	Intel Core i3-6100	3.7 GHz (2 cores)	4266
78	AMD Ryzen Threadripper 1920X	3.5 GHz (12 cores)	4256
79	Intel Xeon E3-1241 v3	3.5 GHz (4 cores)	4252
80	Intel Core i7-7820HQ	2.9 GHz (4 cores)	4229
81	Intel Xeon E3-1271 v3	3.6 GHz (4 cores)	4226
82	AMD Ryzen 5 1600X	3.6 GHz (6 cores)	4218
83	Intel Core i7-8550U	1.8 GHz (4 cores)	4213
84	Intel Core i7-4770R	3.2 GHz (4 cores)	4206
85	Intel Xeon E5-2637 v4	3.5 GHz (4 cores)	4191
86	Intel Core i7-7560U	2.4 GHz (2 cores)	4176
87	Intel Core i7-4771	3.5 GHz (4 cores)	4174
88	Intel Core i7-6700T	2.8 GHz (4 cores)	4173
89	Intel Core i5-6500	3.2 GHz (4 cores)	4170
90	Intel Core i3-4370	3.8 GHz (2 cores)	4159
91	Intel Xeon E3-1535M v5	2.9 GHz (4 cores)	4156
92	Intel Core i5-5675R	3.1 GHz (4 cores)	4146
93	Intel Xeon E5-1650 v4	3.6 GHz (6 cores)	4146
94	Intel Core i5-7360U	2.3 GHz (2 cores)	4145
95	Intel Core i5-4690	3.5 GHz (4 cores)	4140
96	Intel Core i7-4770	3.4 GHz (4 cores)	4131

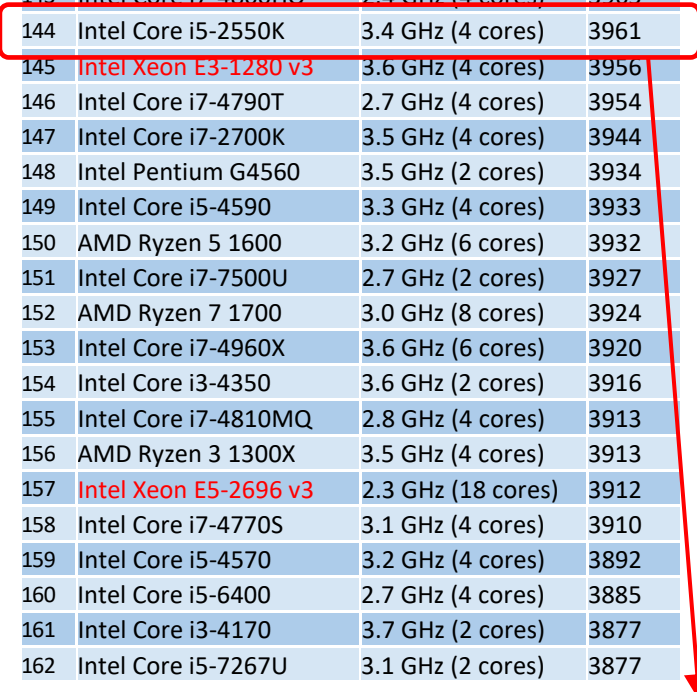
Low grade "budget & cheap" CPU with a price lower than \$170 on the 2018/4/30



97	Intel Core i7-4930MX	3.0 GHz (4 cores)	4129
98	Intel Core i5-7287U	3.3 GHz (2 cores)	4123
99	Intel Core i7-5950HQ	2.9 GHz (4 cores)	4116
100	Intel Core i7-3770K	3.5 GHz (4 cores)	4108
101	Intel Xeon E3-1240 v3	3.4 GHz (4 cores)	4108
102	Intel Core i7-4790S	3.2 GHz (4 cores)	4108
103	Intel Xeon E5-1650 v3	3.5 GHz (6 cores)	4105
104	Intel Core i7-6770HQ	2.6 GHz (4 cores)	4098
105	Intel Xeon E3-1265L v3	2.5 GHz (4 cores)	4093
106	Intel Core i7-7700HQ	2.8 GHz (4 cores)	4093
107	Intel Core i7-7600U	2.8 GHz (2 cores)	4091
108	Intel Xeon E5-1620 v4	3.5 GHz (4 cores)	4083
109	AMD Ryzen 7 1700X	3.4 GHz (8 cores)	4075
110	Intel Xeon E3-1231 v3	3.4 GHz (4 cores)	4074
111	Intel Xeon E3-1270 v3	3.5 GHz (4 cores)	4070
112	Intel Xeon E3-1245 v3	3.4 GHz (4 cores)	4059
113	Intel Core i5-6287U	3.1 GHz (2 cores)	4055
114	Intel Xeon E3-1230 v6	3.5 GHz (4 cores)	4053
115	Intel Core i5-7400	3.0 GHz (4 cores)	4050
116	Intel Core i5-6600T	2.7 GHz (4 cores)	4046
117	Intel Core i5-7260U	2.2 GHz (2 cores)	4046
118	Intel Core i3-4340	3.6 GHz (2 cores)	4043
119	Intel Core i5-4670	3.4 GHz (4 cores)	4035
120	Intel Xeon E3-1505M v5	2.8 GHz (4 cores)	4030
121	Intel Xeon E3-1230 v3	3.3 GHz (4 cores)	4026
122	Intel Core i7-4910MQ	2.9 GHz (4 cores)	4016
123	Intel Core i7-6567U	3.3 GHz (2 cores)	4016
124	AMD Ryzen 5 2400G	3.6 GHz (4 cores)	4016
125	AMD Ryzen 5 1500X	3.5 GHz (4 cores)	4012
126	Intel Core i3-7100T	3.4 GHz (2 cores)	4012
127	Intel Core i7-4770T	2.5 GHz (4 cores)	4010
128	Intel Core i5-3570K	3.4 GHz (4 cores)	4008
129	Intel Core i3-4360	3.7 GHz (2 cores)	4002

130	Intel Pentium G4500	3.5 GHz (2 cores)	3998
131	Intel Core i5-4690S	3.2 GHz (4 cores)	3997
132	Intel Xeon E5-1620 v3	3.5 GHz (4 cores)	3994
133	Intel Core i7-6820HQ	2.7 GHz (4 cores)	3990
134	Intel Core i7-4870HQ	2.5 GHz (4 cores)	3985
135	Intel Core i3-6098P	3.6 GHz (2 cores)	3983
136	Intel Xeon E3-1240 v6	3.7 GHz (4 cores)	3981
137	Intel Xeon E3-1225 v5	3.3 GHz (4 cores)	3979
138	Intel Pentium G4600	3.6 GHz (2 cores)	3978
139	Intel Core i5-6402P	2.8 GHz (4 cores)	3971
140	Intel Core i5-5575R	2.8 GHz (4 cores)	3967
141	Intel Xeon E3-1220 v5	3.0 GHz (4 cores)	3967
142	Intel Core i7-4960HQ	2.6 GHz (4 cores)	3965
143	Intel Core i7-4860HQ	2.4 GHz (4 cores)	3965
144	Intel Core i5-2550K	3.4 GHz (4 cores)	3961
145	Intel Xeon E3-1280 v3	3.6 GHz (4 cores)	3956
146	Intel Core i7-4790T	2.7 GHz (4 cores)	3954
147	Intel Core i7-2700K	3.5 GHz (4 cores)	3944
148	Intel Pentium G4560	3.5 GHz (2 cores)	3934
149	Intel Core i5-4590	3.3 GHz (4 cores)	3933
150	AMD Ryzen 5 1600	3.2 GHz (6 cores)	3932
151	Intel Core i7-7500U	2.7 GHz (2 cores)	3927
152	AMD Ryzen 7 1700	3.0 GHz (8 cores)	3924
153	Intel Core i7-4960X	3.6 GHz (6 cores)	3920
154	Intel Core i3-4350	3.6 GHz (2 cores)	3916
155	Intel Core i7-4810MQ	2.8 GHz (4 cores)	3913
156	AMD Ryzen 3 1300X	3.5 GHz (4 cores)	3913
157	Intel Xeon E5-2696 v3	2.3 GHz (18 cores)	3912
158	Intel Core i7-4770S	3.1 GHz (4 cores)	3910
159	Intel Core i5-4570	3.2 GHz (4 cores)	3892
160	Intel Core i5-6400	2.7 GHz (4 cores)	3885
161	Intel Core i3-4170	3.7 GHz (2 cores)	3877
162	Intel Core i5-7267U	3.1 GHz (2 cores)	3877

163	Intel Xeon E5-1680 v2	3.0 GHz (8 cores)	3871
164	Intel Xeon E3-1280 V2	3.6 GHz (4 cores)	3866
165	Intel Core i7-3960X	3.3 GHz (6 cores)	3857
166	Intel Core i7-2600K	3.4 GHz (4 cores)	3840
167	Intel Core i5-4590S	3.0 GHz (4 cores)	3839
168	Intel Core i7-4610M	3.0 GHz (2 cores)	3839
169	Intel Xeon E5-1660 v2	3.7 GHz (6 cores)	3836
170	Intel Core i5-7300U	2.6 GHz (2 cores)	3830
171	Intel Core i5-2500K	3.3 GHz (4 cores)	3829
172	Intel Core i5-6440HQ	2.6 GHz (4 cores)	3828
173	Intel Pentium G3258	3.2 GHz (2 cores)	3821
174	Intel Core i7-5700HQ	2.7 GHz (4 cores)	3808
175	Intel Core i5-4570S	2.9 GHz (4 cores)	3804
176	Intel Core i7-4930K	3.4 GHz (6 cores)	3797
177	Intel Core i7-4900MQ	2.8 GHz (4 cores)	3797
178	Intel Core i7-3970X	3.5 GHz (6 cores)	3788
179	Intel Xeon E3-1220 v6	3.0 GHz (4 cores)	3782
180	AMD Ryzen 3 2200G	3.5 GHz (4 cores)	3778
181	Intel Core i5-7300HQ	2.5 GHz (4 cores)	3776
182	Intel Core i5-8250U	1.6 GHz (4 cores)	3775
183	Intel Core i7-3770	3.4 GHz (4 cores)	3766
184	Intel Core i5-6267U	2.9 GHz (2 cores)	3766
185	Intel Xeon E5-2687W v4	3.0 GHz (12 cores)	3764
186	Intel Core i3-4160	3.6 GHz (2 cores)	3760
187	Intel Core i7-4820K	3.7 GHz (4 cores)	3745
188	Intel Xeon E5-2667 v4	3.2 GHz (8 cores)	3742
189	Intel Core i3-4330	3.5 GHz (2 cores)	3734
190	Intel Core i7-6700HQ	2.6 GHz (4 cores)	3732
191	Intel Core i7-5557U	3.1 GHz (2 cores)	3727
192	Intel Xeon E3-1245 V2	3.4 GHz (4 cores)	3718
193	Intel Core i7-3930K	3.2 GHz (6 cores)	3714
194	Intel Pentium G4400	3.3 GHz (2 cores)	3712
195	Intel Core i7-4800MQ	2.7 GHz (4 cores)	3709



This CPU was released in Q1 2012 (this is a 6-years-old CPU!)



196	Intel Xeon E3-1270 V2	3.5 GHz (4 cores)	3703
197	Intel Xeon E3-1275 v2	3.5 GHz (4 cores)	3697
198	Intel Xeon E3-1225 v3	3.2 GHz (4 cores)	3680
199	Intel Core i7-4770HQ	2.2 GHz (4 cores)	3680
200	Intel Xeon E3-1220 v3	3.1 GHz (4 cores)	3676
201	Intel Core i5-4460	3.2 GHz (4 cores)	3665
202	Intel Xeon E5-2690 v4	2.6 GHz (14 cores)	3664
203	Intel Core i7-6660U	2.4 GHz (2 cores)	3662
204	Intel Xeon E5-2697 v3	2.6 GHz (14 cores)	3657
205	Intel Xeon E3-1240 v2	3.4 GHz (4 cores)	3654
206	Intel Core i3-6100T	3.2 GHz (2 cores)	3649
207	Intel Core i5-3570	3.4 GHz (4 cores)	3648
208	Intel Core i7-4850HQ	2.3 GHz (4 cores)	3648
209	Intel Core i3-4150	3.5 GHz (2 cores)	3648
210	Intel Core i7-3770S	3.1 GHz (4 cores)	3646
211	Intel Core i7-6700TE	2.4 GHz (4 cores)	3644
212	AMD Ryzen 3 1200	3.1 GHz (4 cores)	3636
213	Intel Xeon E5-2667 v2	3.3 GHz (8 cores)	3629
214	Intel Core i7-4720HQ	2.6 GHz (4 cores)	3629
215	Intel Core i5-6500T	2.5 GHz (4 cores)	3628
216	Intel Core i7-6650U	2.2 GHz (2 cores)	3626
217	Intel Xeon E5-2696 v4	2.2 GHz (22 cores)	3618
218	Intel Core i5-4308U	2.8 GHz (2 cores)	3617
219	AMD Ryzen 5 1400	3.2 GHz (4 cores)	3608
220	Intel Core i5-3550	3.3 GHz (4 cores)	3604
221	Intel Core i7-3840QM	2.8 GHz (4 cores)	3596
222	Intel Core i5-5287U	2.9 GHz (2 cores)	3586
223	Intel Xeon E3-1230V2	3.3 GHz (4 cores)	3578
224	Intel Xeon E5-1650 v2	3.5 GHz (6 cores)	3573
225	Intel Xeon E5-2643 v4	3.4 GHz (6 cores)	3573
226	Intel Xeon E5-2690 v3	2.6 GHz (12 cores)	3572
227	Intel Core i5-4570T	2.9 GHz (2 cores)	3571
228	Intel Xeon E5-1620 v2	3.7 GHz (4 cores)	3569

229	Intel Core i5-4440	3.1 GHz (4 cores)	3566
230	Intel Core i7-6560U	2.2 GHz (2 cores)	3561
231	Intel Core i3-4130	3.4 GHz (2 cores)	3553
232	Intel Core i7-3820	3.6 GHz (4 cores)	3549
233	Intel Xeon E3-1225 V2	3.2 GHz (4 cores)	3546
234	Intel Xeon E5-1660	3.3 GHz (6 cores)	3543
235	Intel Xeon E3-1280	3.5 GHz (4 cores)	3542
236	Intel Core i7-4710MQ	2.5 GHz (4 cores)	3537
237	Intel Core i5-6360U	2.0 GHz (2 cores)	3535
238	Intel Core i7-4710HQ	2.5 GHz (4 cores)	3524
239	Intel Xeon E3-1225 v6	3.3 GHz (4 cores)	3521
240	Intel Core i5-3550S	3.0 GHz (4 cores)	3518
241	Intel Core i7-4578U	3.0 GHz (2 cores)	3514
242	Intel Core i7-3770T	2.5 GHz (4 cores)	3505
243	Intel Core i5-3570S	3.1 GHz (4 cores)	3505
244	Intel Xeon E5-2687W v3	3.1 GHz (10 cores)	3504
245	Intel Core i5-4460S	2.9 GHz (4 cores)	3502
246	Intel Core i7-4700EQ	2.4 GHz (4 cores)	3502
247	Intel Core i5-4430	3.0 GHz (4 cores)	3501
248	Intel Core i3-4330T	3.0 GHz (2 cores)	3495
249	Intel Core i7-3940XM	3.0 GHz (4 cores)	3489
250	Intel Xeon E5-2680 v4	2.4 GHz (14 cores)	3485
251	Intel Core i7-4700HQ	2.4 GHz (4 cores)	3476
252	Intel Core i5-4570R	2.7 GHz (4 cores)	3470
253	AMD Ryzen 7 2700U	2.2 GHz (4 cores)	3470
254	Intel Core i7-6600U	2.6 GHz (2 cores)	3469
255	Intel Xeon E5-2686 v3	2.0 GHz (18 cores)	3466
256	Intel Core i7-4600M	2.9 GHz (2 cores)	3465
257	Intel Core i5-3470	3.2 GHz (4 cores)	3457
258	Intel Core i7-4750HQ	2.0 GHz (4 cores)	3457
259	Intel Pentium G3460	3.5 GHz (2 cores)	3450
260	Intel Xeon E5-1650	3.2 GHz (6 cores)	3447
261	Intel Core i5-6300HQ	2.3 GHz (4 cores)	3440

262	Intel Core i3-4170T	3.2 GHz (2 cores)	3433
263	Intel Core i5-3450S	2.8 GHz (4 cores)	3429
264	Intel Xeon E5-2660 v3	2.6 GHz (10 cores)	3426
265	Intel Core i7-2600	3.4 GHz (4 cores)	3424
266	Intel Core i5-3475S	2.9 GHz (4 cores)	3414
267	Intel Core i5-4670T	2.3 GHz (4 cores)	3414
268	Intel Core i7-4712MQ	2.3 GHz (4 cores)	3409
269	Intel Core i7-4700MQ	2.4 GHz (4 cores)	3402
270	Intel Xeon E5-1620	3.6 GHz (4 cores)	3397
271	Intel Core i5-4200H	2.8 GHz (2 cores)	3396
272	Intel Core i5-3450	3.1 GHz (4 cores)	3393
273	Intel Xeon E5-2697 v4	2.3 GHz (18 cores)	3392
274	Intel Core i7-4785T	2.2 GHz (4 cores)	3387

As you can see in the table here above, there exists only very few correct XEON processors (the XEON CPU's are written with a **red** font). Nearly all the best processors are "Core I7".

At the position 5 of the above table, we find a low-grade, cheap CPU (the "Intel Core i3-8350K" that costs less than \$170 on the 2018/4/30). All XEON processors are always much more expensive and, furthermore, all of them are much slower than this entry-grade, low cost, "Core I3" CPU. One more reason to just avoid Xeon processors.

At position 144 of the above table, we find a 6 years old CPU (the "Intel Core i5-2550K" that was released in Q1 2012). All the CPU's slower than this very, very old CPU (i.e. all the CPU's located at a ranking number above 144) should be avoided at all costs. For example, one of the most common XEON processor is the "Xeon E5-2660 v3" processor (because it offers a large core-count and it's "relatively cheap" compared to other Xeon processors). Unfortunately, this "Xeon E5-2660 v3" processor has also a catastrophic ranking position of 264 (it's much, much slower than a 6 years-old CPU!). So, my advice regarding XEON processors in general is: **Take extreme caution or just avoid them.**

You must realize that the biggest buyers of large "professional" servers are the (web) data centers. This means that, overtime, the offer (from the PC manufacturers) has adapted to the demand (from the web data centers) and nowadays, most large "professional" servers have quite good hard-drives BUT VERY BAD CPU's (i.e. they have low-grade XEON servers such as the E5-2660 v3) because this is what's most commonly required in standard web-data-centers.

1.5. The Best CPU and Motherboard for a TIMi Server (as of March 2018)

As you can see in the above table, the best CPU on the 2018/4/30 for Anatella/TIMi is the Intel Core i7-7800K at 3700MHz. When selecting a server, you must also pay attention to the motherboard: A good CPU installed on a bad motherboard also gives a poor result (i.e. low speed).

If your IT department allows it, you can buy an assembled, tested & configured server (with a good motherboard) for a good price here (as of 2018/4/30):

<https://www.ldlc-pro.com/fiche/PB00248584.html>

The components of the " LDLC PC10 RealT Coffee Edition" desktop computer are:

- CPU: 6-Core Intel Core i7-8700K (3.7 GHz)
- Mother board: Gaming ASUS ROG STRIX Z370E GAMING
- RAM : 16 Go DDR4 2400 MHz
- System disk: SSD M.2 PCIe NVMe 240 Go
- Mass Storage Disc: Seagate Barracuda 7200.14 SATA 6Gb/s 2 To
- GPU: NVIDIA GeForce GTX 1080
- Network Card : Gigabit LAN
- 8 high-definition audio canals
- Power Supply: LDLC US-650G Quality Select 80PLUS Gold - 650 Watt
- CPU cooler: Watercooling All-in-one
- Box : IN WIN 303C black
- Microsoft Windows 10 Family 64 bits

LDLC PC10 ReaIT Coffee Edition

Groupes LDLC SA [FR] | https://www.ldlc-pro.com/fiche/PB00248584.html

France | Groupe LDLC | Besoin d'aide ?

LDLC HIGH-TECH PARTNER

COMPTE Se connecter | PANIER 0 article 0€00


ORDINATEURS | PIÈCES | PÉRIPHÉRIQUES | IMAGE & SON | MOBILITÉ | RÉSEAUX | CONSOMMABLES | LOGICIELS | PAPETERIE | CONNECTIQUE

Rechercher... OK Packs Garantie & services

Ordinateurs > Ordinateur de bureau > LDLC PC10 ReaIT Coffee Edition

LDLC PC10 REALT COFFEE EDITION

Intel Core i7-8700K (3.7 GHz) 16 Go SSD M.2 NVMe PCIe 240 Go + SSHD 2 To NVIDIA GeForce GTX 1080 8Go Windows 10 Famille 64 bits (monté) (ref: PC10 REALT COFFEE EDITION)



zoom

Soyez le premier à donner votre avis

- Partager cette fiche
- Imprimer la page
- Être informé d'une baisse de prix
- Ajouter à mes préférés

> Découvrez tous les produits LDLC

LDLC

> Ordinateur de bureau LDLC

1 791€63 HT
dont éco-participation 1€20

Quantité: - 1 +

COMMANDER

Livraison partout dans le monde | EN STOCK | chronopost EXPRESS

NOS EXPERTS À VOTRE SERVICE | 04 27 46 60 05 | Je souhaite être rappelé par un conseiller expert

Nouveau !

The above PC makes a solid Anatella/TIMi server. It's one of the best server you can buy for Anatella/TIMi (maybe the best one).

If you are a little short on the budget, you might also consider this server (because it also has 6 "real" cores and each core is really fast):

LDLC PC10 ReaIT Free Coffee Edition

Groupes LDLC SA [FR] | https://www.ldlc-pro.com/fiche/PB00248581.html

France | Groupe LDLC | Besoin d'aide ?

LDLC HIGH-TECH PARTNER

COMPTE Se connecter | PANIER 0 article 0€00


ORDINATEURS | PIÈCES | PÉRIPHÉRIQUES | IMAGE & SON | MOBILITÉ | RÉSEAUX | CONSOMMABLES | LOGICIELS | PAPETERIE | CONNECTIQUE

Rechercher... OK Packs Garantie & services

Ordinateurs > Ordinateur de bureau > LDLC PC10 ReaIT Free Coffee Edition

LDLC PC10 REALT FREE COFFEE EDITION

Intel Core i5-8600K (3.6 GHz) 16 Go SSD 240 Go + HDD 2 To NVIDIA GeForce GTX 1060 6Go Windows 10 Famille 64 bits (monté) (ref: PC10 REALT FREE COFFEE EDITION)



zoom

Soyez le premier à donner votre avis

- Partager cette fiche
- Imprimer la page
- Être informé d'une baisse de prix
- Ajouter à mes préférés

> Découvrez tous les produits LDLC

LDLC

> Ordinateur de bureau LDLC

1 291€63 HT
dont éco-participation 1€20

Quantité: - 1 +

COMMANDER

Livraison partout dans le monde | EN STOCK | chronopost EXPRESS

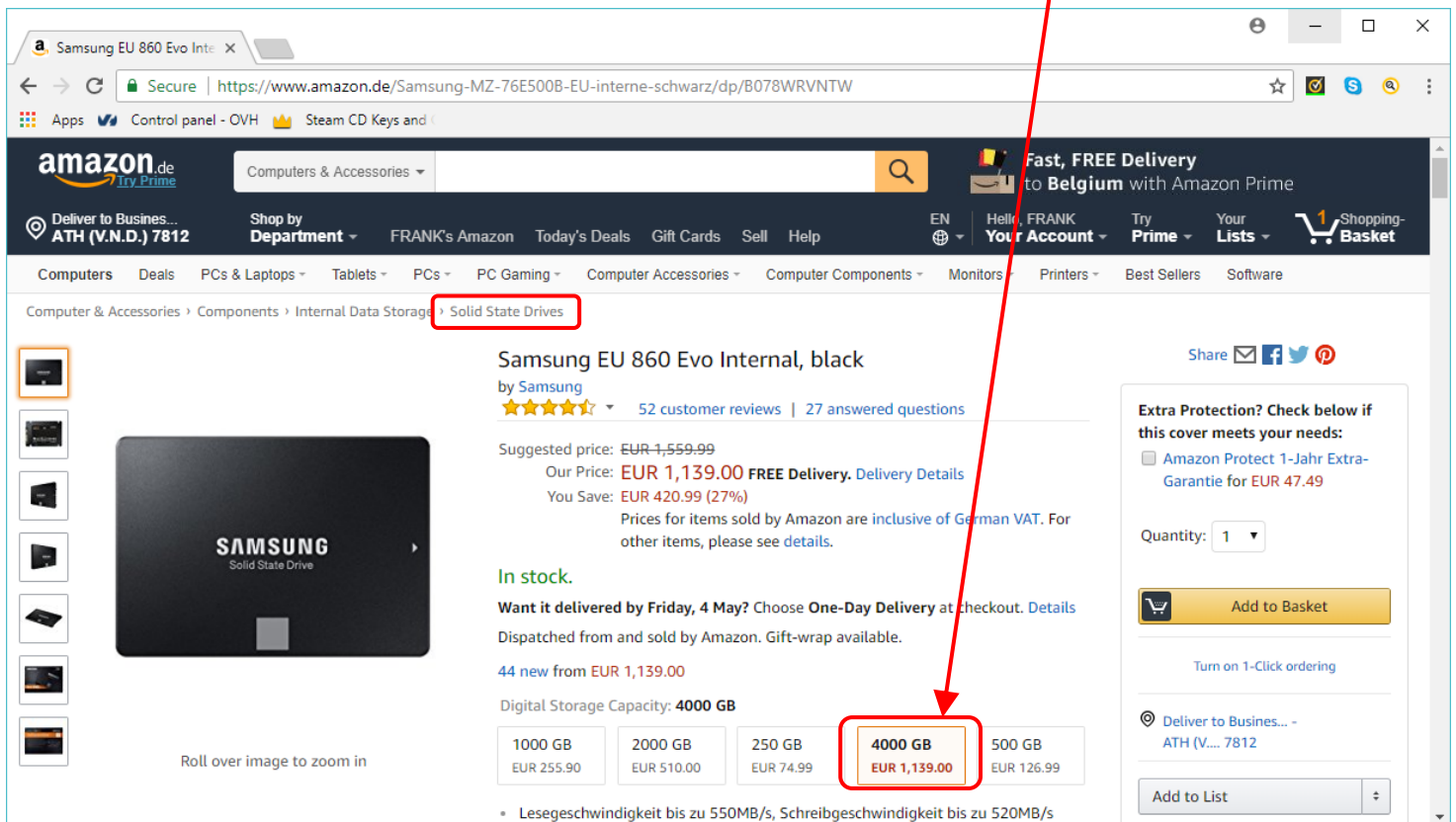
NOS EXPERTS À VOTRE SERVICE | 04 27 46 60 05 | Je souhaite être rappelé par un conseiller expert

Nouveau !

If you want the best performances, you should also:

- Upgrade your windows home edition to a professional edition (this is a one-click purchase inside the Windows Store. It costs \$99).
- Add some good SSD drives (SSD drives are a ***lot more reliable*** than classical, old magnetic HD drives, so it make sense to use an SSD drive if you don't trust too much your backup solution – or when you have no NAS and many simultaneous Anatella users on the server). SSD drives are cheap nowadays, so I would suggest you this one (4TB SSD for 1140€):

<https://www.amazon.de/Samsung-MZ-76E500B-EU-interne-schwarz/dp/B078WRVNTW>



The screenshot shows the Amazon.de product page for the Samsung EU 860 Evo Internal SSD. The product is listed as 'Samsung EU 860 Evo Internal, black' with a suggested price of EUR 1,559.99 and a current price of EUR 1,139.00. The page includes a navigation menu, a search bar, and a 'Solid State Drives' breadcrumb. The product image shows the SSD with the Samsung logo. The price and '4000 GB' options are highlighted with red boxes. A red arrow points from the URL above to the '4000 GB' option. The performance specifications at the bottom are also highlighted with a red box.

Read-Speed: 550 MByte/sec ; Write-speed: 530 MByte/sec

2. Common infrastructures to run TIMi/Anatella

First a little bit of terminology: Any “Advanced Analytics” architecture/infrastructure must take into account that an advanced analytic project has always two phases:

1. **Phase 1: The “Exploration phase”**

What are the characteristics of the “Exploration Phase”?

- The Analysts/Data Scientists are developing a new KPI, a new predictive Model or, in general, creating new results through the analysis of data.
- The Analysts/Data Scientists typically run very heavy data transformations, very heavy computations, searching for the “golden egg”. On a “standard” infrastructure where all the computations are “centralized” on a central database or a central hadoop cluster, these heavy data transformations might disrupt the work of other analyst or, even worse, jeopardize the global stability of the whole IT infrastructure of the company (This is why, in most companies, the Data Scientists are not the “friends” of the IT people).
- It doesn’t matter so much if one “heavy” computation fails (e.g. because of a bad parameterization).
- The duration of the “Exploration phase” is, typically, from a few hours to a few weeks.

2. **Phase 2: The “Production phase”**

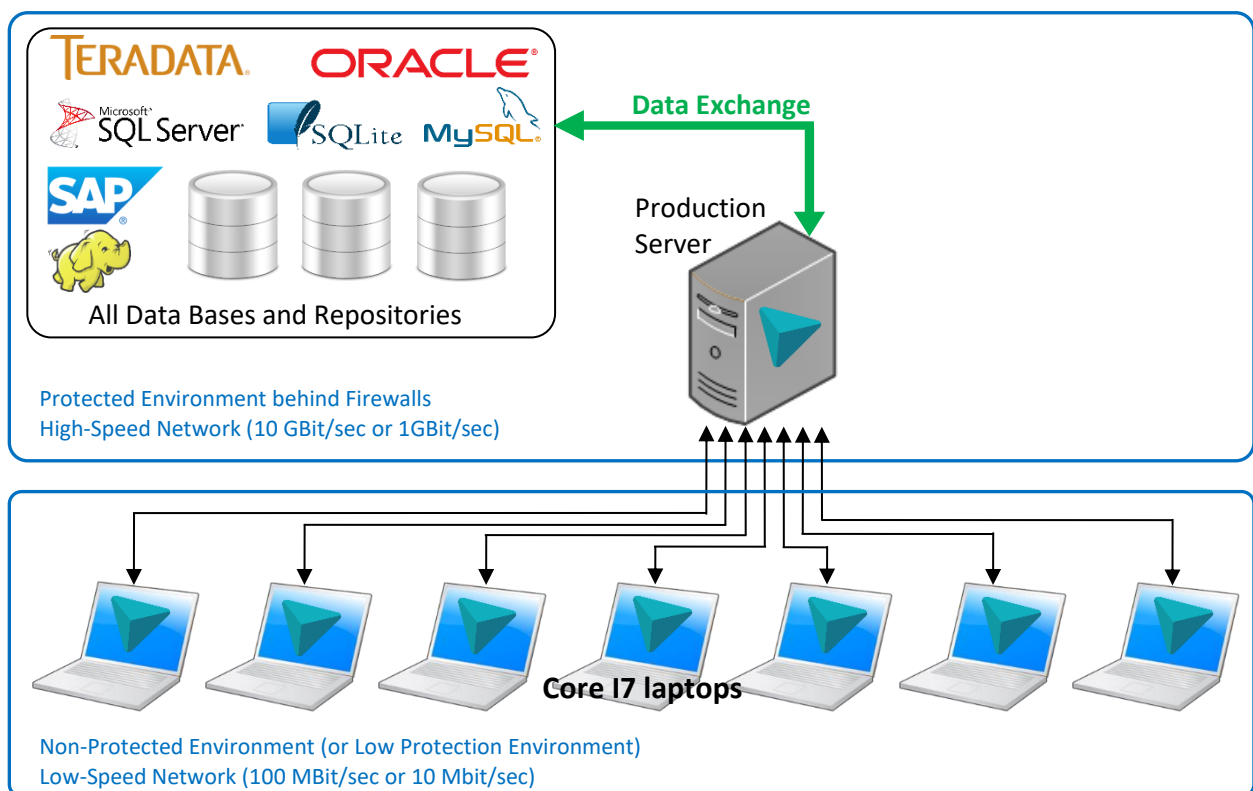
The “Production phase” comes after the “Exploration phase” and usually lasts for years. There are usually no really “heavy” computations during the “Production phase”.

The main concerns of the “Production phase” is the stability: All processes must run smoothly, without ever failing.

We’ll now review 4 different infrastructures, from the cheapest one to the most expensive one. There are no “best” infrastructure: It all depends on your budget and your particular business needs.

2.1. A first (cheap) infrastructure

Let's start with the most common, cheapest infrastructure:



What are the general principles behind such an infrastructure?

- The datasets on which the Analyst are working are centralized on the “Production server” (they are stored inside .gel_anatella files or .cgel_anatella files). Since these file formats are (heavily) compressed, the storage space is practically ***never*** an issue.
- The “Production Server” runs automatically, on a regular basis (typically: every day or week), the different Anatella-data-transformation-graphs and the different timi-models to create all the KPI's, dashboards or scores required for the normal operation of the company.

The computations on the “Production Server” are always based on “fresh data” originating from the various data centers, data warehouse and, or any other data sources.

- During the “exploration phase”, all the Analysts & all the Data Scientists are working on their own (Core I7) laptops. The above picture illustrates the situation when there are 7 Data Scientists inside the company (since there are 7 laptops inside the picture). Since each laptop has roughly three times the computing power of an Oracle Exadata machine, each analyst has, basically, enough computing power to compute any data transformations. The only limiting factor is the access & the storage of the datasets on which the Analyst are working.

- During the “exploration phase”, an Analyst/Data Scientist needs some data to work with (obviously). This data is typically originally stored on the “Production Server” (although, it can change: see some alternative solutions below).

Typically (for performance reasons), the Analyst/Data Scientist will make a copy of the required data on its laptop and work on the copy to find new results. If the “exploration phase” lasts for a long period of time, the Analyst might need to refresh its local copy of the datasets, but 99% of the time, the Analyst can work with slightly outdated data to produce the required analysis results.



In general, as an analyst, you should avoid using data stored on a distant machine or accessing data through a slow network interface (That’s ok if the network interface that is connected to your laptop is a 10Gbit/sec network interface but I guess that it won’t be the case).

For efficiency reasons, 99% of the time, it’s better to copy locally **one time** your data on an encrypted partition on your local PC and then work with the local copy.

To encrypt a partition, you can use the free “bitlocker” application included inside MS-Windows or the famous & free TrueCrypt application.

Once the analysis is complete (i.e. once the new KPI looks good, once the new predictive models are ok, once the new reports are ok, etc.), the Analysts “moves” all his graphs & models to the “Production Server”. In technical terms, we’ll say that the “exploration phase” is finished and the “production phase” starts. Once the graphs & the models are on the “Production Server”, they will be applied on the “fresh” data, to always get “fresh” results (i.e. the “production phase” is always on “fresh” data).

Moving a data-transformation process from one computer to another (i.e. from the Analyst’s laptop to the “production server”) is usually an error-prone procedure (e.g. it’s usually a real nightmare with SAS). Contrary to all the other solutions, Anatella possesses some unique functionalities (e.g. relative path to the datasets, self-contained .anatella files, etc.) that allow you to migrate effortlessly all your work (graphs & models) from one machine to another.

Here are the Pro & Con of the above infrastructure:

- Pro:
 - **Cheap:** No hardware investment required outside the purchase of a “Production Server”
 - **Scalable:** If you have more analysts, simply add more laptops.
 - **Secure:** Why? Because:
 - The Analysts/Data Scientists cannot delete any critical data from your operational systems because all they are allowed to do is to copy, from time-to-time, some .gel_anatella files or some .cgel_anatella files on their local hard drive. They can’t even damage the .gel_anatella files (or .cgel_anatella files) that other analysts might require because the .gel_anatella files (or .cgel_anatella files) are read-only files.
 - During the “exploration phase”, the Analysts/Data Scientists typically run very heavy data transformations, searching for the “golden egg” in your data. On a standard, centralized infrastructure, these *heavy* data transformations

might disrupt the work of other analysts or, even worse, jeopardize the global stability of the whole IT infrastructure of the company.

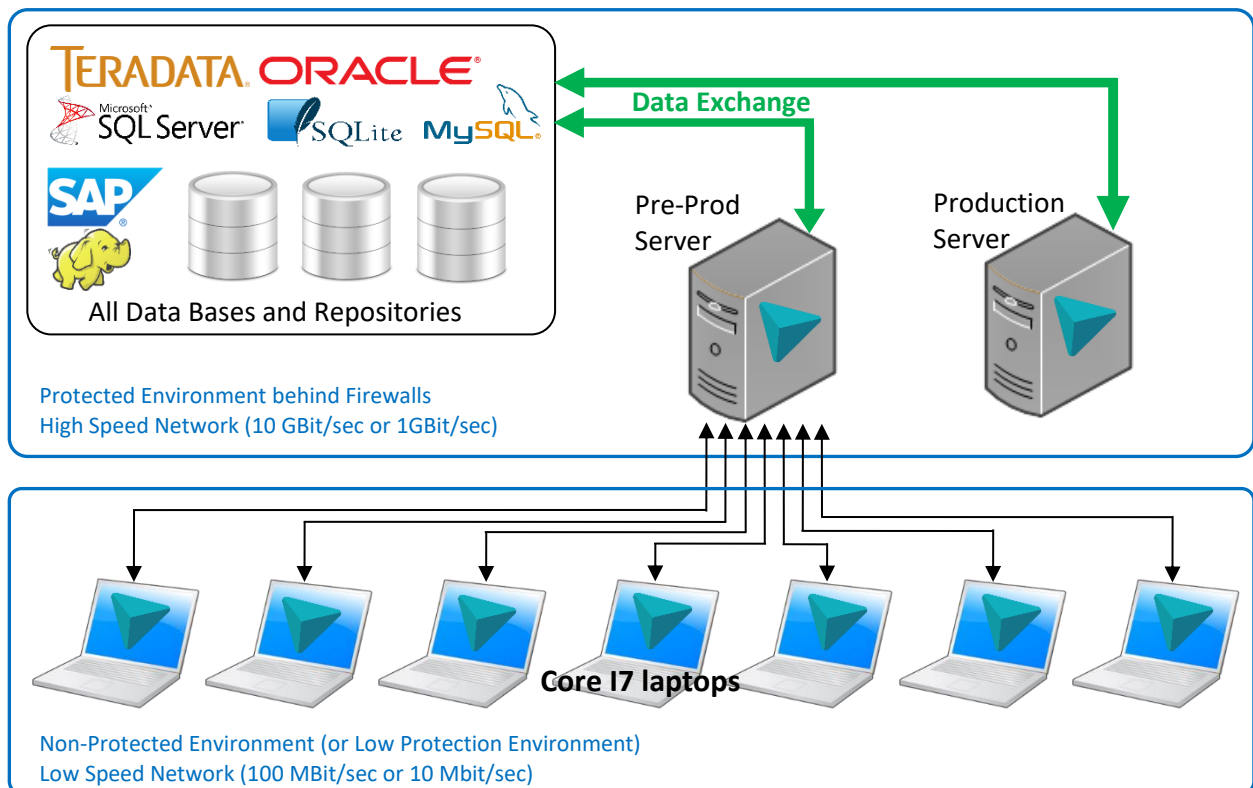
This disastrous situation will never happen with the proposed solution here above: Indeed, each of the Data Scientists is using its own CPU without consuming any resource from the production servers (or from other Analysts).

- Con:
 - Once an analysis is complete (i.e. once the “exploration phase” is complete) and a new process is developed, it’s directly “published” to the “Production Server” without any testing-period first. This means that a bad, untested, process could consume so much processing power that it could “bring to its knees” the production server (which is a bad thing). We should add a pre-production server to avoid such situation, to improve reliability.
 - All the datasets are stored on the “production server” (i.e. the .gel_anatella and the .cgel_anatella files). If many Analysts/Data Scientists decide to simultaneously copy some large datasets on their laptop, the “production server” might slow-down briefly (due to the many simultaneous “copy operations”).
 - During the “exploration phase”, some datasets are copied on the laptops from the Analysts/Data Scientists for a brief period of time (i.e. for the time required for them to produce new results: i.e. to produce new graphs and new models). This might be a concern for very sensitive data (especially for banks, insurance, etc.). To alleviate this problem, the data is usually stored on the laptops on an encrypted partition (the partition is encrypted with bitlocker or truecrypt) but this might not be a solution that is “secure enough”.

In the next sections, we’ll review each of these “con” arguments and give different solutions to each of them.

2.2. A second infrastructure

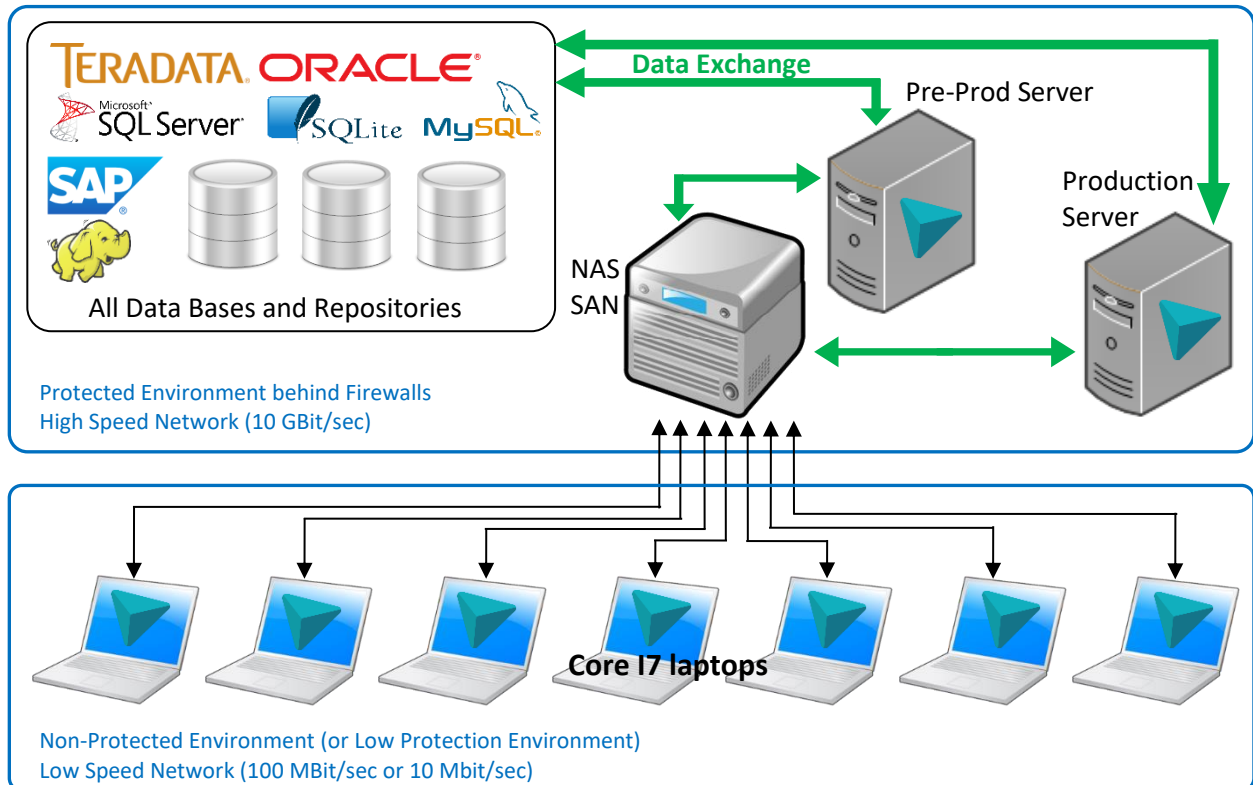
To increase reliability, use a “pre-production” server:



Once an analysis is complete (i.e. once the “exploration phase” is complete) and a new process is developed, it’s “published” to the “Pre-Production Server” for a few weeks. If the “Pre-Production” server remains stable & responsive, you can thereafter “publish” your new graphs&models from the “Pre-Production” environment to the final “Production Server” environment.

2.3. A Third infrastructure

To increase throughput when many laptops are accessing the datasets (stored in the .gel_anatella file and the .cgel_anatella files), use a NAS (Network Attached Storage):



Please note that, since all your datasets are now on a NAS, you need some fast network interface between the “Production Server”, the “Pre-Prod server” and the “NAS”. Ideally, you should use a 10 GBit/sec network infrastructure. If you are using a more standard (i.e. less expensive) infrastructure based on standard 1Gbit/sec network cards, you might occasionally experience some slower disk access.

Please refer to the following table to know more about this subject:

Physical Location of the data	Maximum I/O Speed for processes running inside the “Production Server”
RAID6-Drive inside the “Production Server” (or a SAN inside the “Production Server”)	2000 MByte/sec (e.g. using 4 standard SSD drives)
One new-generation NVMe SSD drive inside the “Production Server” (this is a bargain price!)	2000 MByte/sec (but NVMe SSD drives are usually small: less than 1GB capacity)
NAS within a 10Gbit/sec network	1000 MByte/sec on a “BIG” NAS 500 MByte/sec on a “small” NAS
One standard SSD drive inside the “Production Server”	500 MByte/sec
NAS within a 1Gbit/sec network	100 MByte/sec This might increase up to 800 MByte/sec if you use “port trunking” but this is uncommon. For more information, see here: https://en.wikipedia.org/wiki/Link_aggregation
One Magnetic drive inside the “Production Server”	80 MByte/sec

Recommended solutions (depending on your budget)

It must be said that this it's usually useless to be able to read your data at 2GB/sec because reading data is rarely the "bottleneck" when running an Anatella graph (i.e. inside Anatella, the CPU is usually the bottleneck, not the I/O's). This is why we will practically never advise you to buy the fastest (and more expensive) storage solution: i.e. Most of the time, a server equipped with a standard SSD drive, already delivers optimal I/O performances (see section 1.5 for some advices about SSD drives).

2.3.1. A small Note about I/O speed

The objective of this section is to explain why the I/O's are not usually a bottleneck when manipulating data with Anatella (i.e. the CPU is usually the bottleneck and not the I/O's).

Anatella works in "streaming" (in opposition to R or python that works, by default, in "batch"). This means that, inside Anatella, there exists a data flow that is going "through" all the operations inside the "Anatella data transformation graph" (abbreviated to "graph"). For example, if you want to join two tables, you must (of course) read the data from both tables (you can read each table at 80 MB/sec "compressed data" or 800 MB/sec "uncompressed data") and, at the same time, compute the join, line-by-line (in streaming). However, the calculation of the join in itself is very expensive: It can only be done at 100 MB/sec on average (on a standard telecom table). So, there's a "bottleneck" at 100 MB/sec: i.e. it's useless to extract/read "data lines" out of the hard drive at a speed of 800 MB/sec if, right after the lecture, the data flow can only be processed at a maximum speed of 100 MB/sec.



In the above example, the execution time is 100% governed by the speed of the join (and not at all by the speed of the hard drive or the speed of the I/O accesses).

It's a little in opposition to codes in R/Python, where the execution times of the different components add up: With Anatella, the execution time of a graph is (usually) proportional to the slowest element of the graph.

This is why it is possible to tell Anatella to allocate a larger number of CPU's to a particular operation (i.e. to a particular "box"), to avoid/reduce this "bottleneck" effect (for more information about this subject, see section 5.3.2 of the "AnatellaQuickGuide.pdf"). For example, Anatella could be told to use 7 CPU's to calculate the join (instead of using one CPU by default), to get a throughput of $7 \times 100 \text{ MB/sec} = 700 \text{ MB/sec}$ at the end (and, therefore, removing the "Bottleneck" of the join).

In practice, one quickly realizes that the hard disk (or the I/O speed) is practically NEVER the "bottleneck element" that decides of the overall execution time of an Anatella-data-transformation-graph. That's why we are now putting most of our development efforts in improving the speed of all other components (join, sort, filter, scoring) inside Anatella. For example, I think no one can beat the "sort routines" included in Anatella.

Currently, 99% of developers that are working in the R, Python, Hadoop ecosystems, etc. did not yet arrived to the same conclusion as us, and thus they are still (and quite stupidly) trying to get better I/O's. These developers did not manage to get the same conclusions as us because:

- They have a different architecture (that is not based on "data streaming", as in Anatella).
- They are using the Java language (that has such terrible I/O performances that it's always blocking everything).

- They have different “workloads”: Anatella is built for analytics and predictive analytics tasks in mind. Such type of workloads typically requires complex, CPU-intensive computations (to create refined KPI’s or to do “feature engineering”) that dominates the computation time: i.e. these CPU-intensive operations usually represent 95% of the computation time (i.e. inside Anatella, the CPU is usually the bottleneck). So, if we can read the data faster, we will (maybe!) just gain a few percent out of the 5% of time that Anatella devotes to reading the data. On the other hand, it’s true that, for a very simple “Anatella-data-transformation-graphs” (e.g. for example, a graph that contains only one aggregate to compute), it’s worth reading the data faster.



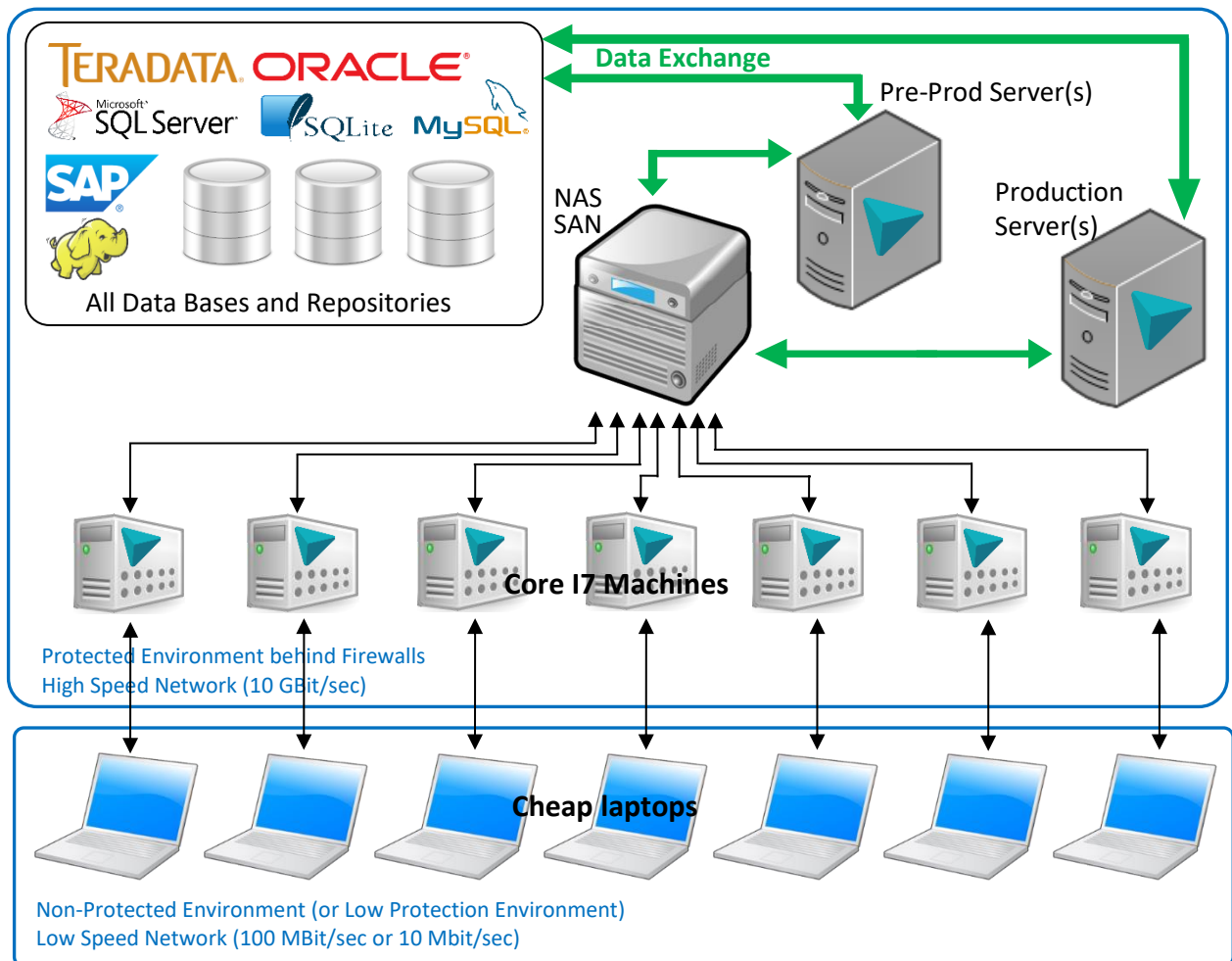
Anatella can also run little “boxes” coded in R or Python. What’s happening for such little box? It’s true that the “normal” R and Python engines do not work “in streaming” (such as Anatella) but we managed to “transform & upgrade” these engines so that they can also work (most of the time) in streaming like the rest of the “boxes” inside Anatella. So, no worries! 😊

An amusing side-effect of this “upgrade” is the possibility to easily run in “parallel” (i.e. on several CPU’s) R and Python codes (i.e. there is an almost automatic parallelization of the R/Python code).

A good example is the “R_ApplyModel” box in Anatella that runs 100% in streaming mode and on multiple CPU’s (inside a N-Way multithreaded section: See the section 4.8.3.2. of the “AnatellaQuickGuide.pdf” for more information about this subject).

2.4. A fourth infrastructure

If you have very sensitive data, it might be better that your data always stays only inside your “Protected Environment behind Firewalls” and you’ll have something like:



Each Analyst/Data Scientist is accessing its own “Core i7” machine using the standard Remote Desktop Protocol (The analysts only use their laptop as a simple terminal, so the laptops can be “cheap”). This architecture is slightly more expensive because you need to buy two machines (a good “Core i7” machine and a “cheap” laptop) for each new Data Scientist (instead of only one previously). The big advantage is, obviously, that your confidential datasets won’t leave your “Protected Environment behind Firewalls” and you have a very secure solution.

2.4.1. Hadoop Integration

This fourth infrastructure has another advantage: i.e. We can use the additional “Core i7” machines as Hadoop “data nodes” to create a large HDFS drive for a low price.



What’s an HDFS drive? An HDFS drive is a logical drive that is built “on top” of several real hard drives that are physically placed in several (usually cheap) servers. In Hadoop terminology, these servers (that contains your data) are named “data nodes”. An HDFS drive is composed of several “data nodes” (that contain data) and one “name node” (that is indexing the data contained in the HDFS drive: i.e. the name node contains an index that allows to know on which data node is stored each chunk of data).

The advantages of HDFS are:

- * It's a really cheap storage solution compared to a traditional database system because:
 - # It only uses common grade hardware (i.e. low costs PC's) that is much cheaper than the dedicated/specialized hardware (such as the "InfiniBand network cards") found in Teradata, Exadata, etc. databases.
 - # it's open source and free software: There are no licensing costs.
- * It's a solution that is easily extensible: If you need more storage, simply add more servers (These servers are named "data nodes" in Hadoop terminology).
- * An HDFS drive can store files that are larger than one physical drive. For example, an HDFS drive can store a 2TB file although it's composed of only 1TB physical hard drives.

The disadvantages of HDFS are:

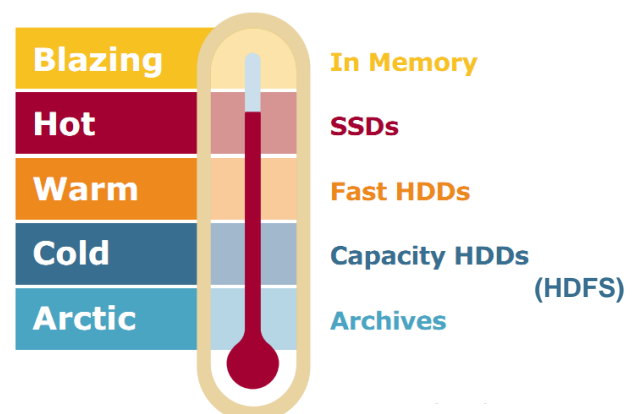
- * It's an expensive storage solution compared to a traditional SSD/NAS/SAN/RAID6 drive. It's more expensive because you need to hire specialized staff for maintenance and support of all the tools inside the Hadoop ecosystem (including, of course, your HDFS drive). The tools in the Hadoop ecosystem are known to lead to large maintenance and support costs to keep them running; i.e. you'll need a specialized staff that is able to keep your Hadoop environment "Up & Running". Luckily, the HDFS drive is an Hadoop component that is amongst the easiest to maintain.
- * It's inefficient in terms of storage consumption. Let's take a simple example: Let's assume that you want that your storage system stays operational (without any data loss) even if there are some catastrophic failures inside 2 physical disks (i.e. the storage system must be resilient to 2 failures). In such condition, to store a 2GB file, we'll have:
 - # in a RAID6 drive: 3GB of disk space is used.
 - # in a HDFS drive: 6GB of disk space is used.
 This means that the storage cost is (at least) two times higher for HDFS than for RAID6 (because you need to buy two times more physical disks to have the same capacity and the same resilience).
- * The data access is quite slow, especially compared to a local SSD/RAID6 drive (that runs between 500 MByte/sec and 2000Mbyte/sec). With HDFS, you can expect a read speed between 5 Mbyte/sec and 50 MByte/sec (it mainly depends on your network cards).

The two major drawbacks of an HDFS drive are (1) its heavy price (compared to a local SSD/RAID6 drive) and (2) its low speed. Still, it's one of the only solution that offers (theoretically) an unlimited data capacity and it can thus make sense to use an HDFS drive if you need to store really large volumes of data (that do not fit inside a SSD/RAID6 drive, despite the efficient data compression algorithms used in Anatella).

Furthermore, the slow speed of HDFS is usually not really a problem when you manipulate your data using Anatella because, when running an Anatella-Data-Transformation-Graph, the I/O speed is usually not the "bottleneck" (i.e. the bottleneck is usually the CPU): See the section 2.3.1. from this document to know more about this subject.

To alleviate the slow speed of HDFS, we can also choose how the data is stored using a methodology based on "data temperature". We use the term "temperature" as a metaphor for "frequency of access":

- Seldom-used data is "cold data" that should be kept on low-cost, high-capacity disk storage, such as HDFS.

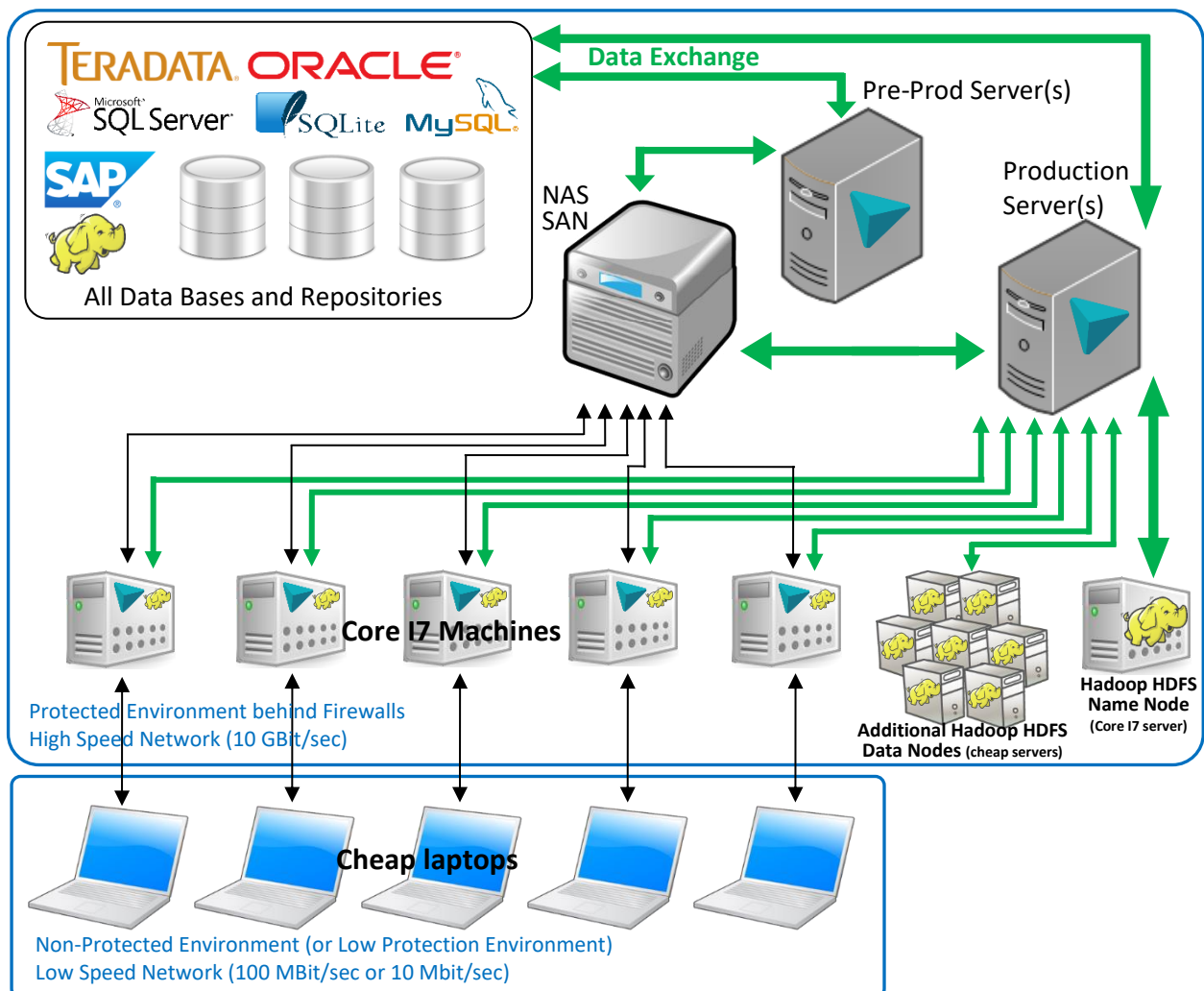


Cold data may not be the most popular, but it is used daily to support vital strategic analytics and decision-making. The cold data is often last year's financials, government risk and compliance data, or consumer behaviours that is used by the marketing department for loyalty or churn analysis.

- High-use data is “hot data” and it should be placed on high-performance SSD, SAN/NAS or RAID6 drives.

Hot data typically includes all recently collected data tables. It also includes all the “heavy used” datasets that are automatically prepared every night and used during the day by the data scientists to complete their daily tasks.

To summarize, when using an Hadoop HDFS drive, we'll have:



For stability reasons, we strongly advise you to use a (good) dedicated machine for the “Hadoop HDFS Name Node” that runs no other process than the “HDFS Name Node” process.

3. Integration with third party tools

3.1. Integration with Hadoop HDFS

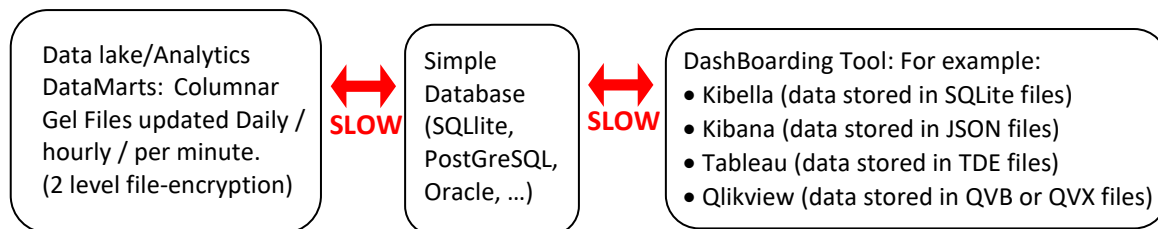
See the previous section: Section 2.4.1.

3.2. Integration with “simple” BI tools

All the proposed infrastructures allow an easy&efficient integration with “simple” BI/Reporting/Dashboarding tools.

Typically, these BI tools are used to display some reports or dashboards inside a browser. There are many different techniques to give to the BI tools the datasets required to compute the reports and the dashboards.

One first solution is the following:



The advantages of the above solutions are:

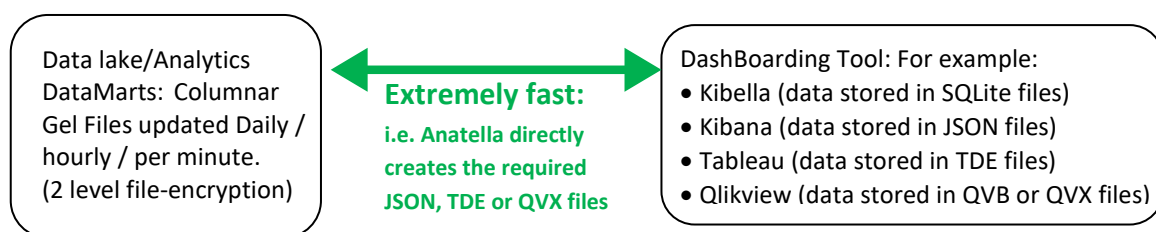
- It works with all DashBoarding tools because (nearly) all the DashBoarding tools can get their data from “common” SQL-based data sources (one exception is Kibana that requires to get its data from JSON files).

The dis-advantages of the above solutions are:

- The display-speed is “sluggish”. More precisely: Qlickview, Tableau and Kibana are much more efficient when the data to display is stored inside their own proprietary format (i.e. when it’s stored inside TDE files for Tableau or inside QVB/QVX files for Qlickview, or inside JSON files for Kibana). The time required to update a webpage containing an interactive dashboard is a lot longer when the dataset is stored in a remote database. At the end, the end-user experience when using a DashBoarding that runs many extraction out of a remote database is the following: it “feels sluggish”: i.e. the user is enduring (intolerable) delays in all webpage-refresh (this is why Qlickview promotes so much “in-memory” analytics). In particular, you should really avoid “slowly reacting” databases (such as Hive or PostgreSQL).
- Some advanced functionalities are only available when the data is properly saved in the proprietary format of the tool (see the qlickview documentation: e.g. in Qlickview, some aggregates are only available when the data source is a QVB/QVX file ; The same exists in Tableau: i.e. Some aggregates are only available when the data source is a TDE file).
- It’s inefficient in terms of hard-drive-space-consumption because there exists several copies of the datasets required for display. The first copy is stored into the DashBoarding tool in itself (i.e. it’s stored inside TDE files for Tableau, or inside QVB/QVX files for Qlickview, or inside JSON files for Kibana). ...and the second copy of the data is stored inside the “Simple Database” (located in the central position of the above chart).

- It's inefficient in terms of refresh-speed, when refreshing/updating the data source:
 - i.e. copying data from the data lake and into the database “in-the middle” (these are “insert”-type of operation) is ***extremely* slow** (“insert” operations in databases are slow). You can (partially) avoid this very slow “copy operation” by using a special database type: a SQLite database. One unique particularity of the SQLite databases is the extremely high speed of the “bulk-insert-operation” (i.e. writing data inside a SQLite database is nearly as fast as writing the same data inside a simple flat text file). Any other database will lead to a very slow running time.
 - i.e. copying data from the database “in-the middle” into the Dashboarding tool (typically, using an ODBC connection). We are actually talking about executing “*database-extraction procedures*”. Such kind of procedures are **always slow**. The Tableau and the Qlikview documentation both agree that, getting data from a file (i.e. from a TDE file for Tableau or from a QVX file for Qlikview) is the most efficient way of getting data-access: The Tableau & Qlickview documentation states that, getting data through a TDE file (for Tableau) or a QVX file (for Qlickview) is usually around 100 times faster, compared to a database-extraction.

To solve all the above problems, we propose the following:



Anatella creates the required dataset files directly inside the proprietary format of the Dashboarding tools. i.e. Anatella creates QVX files for Qlikview, Anatella creates TDE files for Tableau, Anatella creates JSON files for Kibana. The “bottleneck-in-the-middle” (i.e. the database) has disappeared: it has been replaced by a very fast Anatella procedure that generates (at a high speed, since it’s Anatella that is running !) all the required files.



If you intend to use still another Dashboarding tool (i.e. not Tableau, Qlikview or Kibana), you can still get a decent speed by using a SQLite database as the “database-in-the-middle”. Please refer to the section 4.8.2.4. and 4.8.17.6 of the “AnatellaQuickGuide.pdf” for more information about SQLite databases (e.g. why they are so great for interacting with such kind of Reporting/Dashboarding tools)



If you're using Qlikview as a BI tools, there's a great article on the subject of QVX files inside Qlickview here:

<https://community.qlik.com/blogs/qlikviewdesignblog/2014/02/10/odbc-confusion>

Here is an excerpt out of this webpage:

Qlik Design Blog

The Great ODBC Confusion

Posted by **Henric Cronström** in Qlik Design Blog on Feb 11, 2014 2:46:31 AM

Before the ODBC interface to databases was developed in the late 80's and early 90's, it was difficult to connect to a database and import data. But thanks to Microsoft and some other DB vendors, we got an open interface with which we still today can load data from almost any database.

But, some aspects of the Windows ODBC implementation are confusing...

When ODBC was developed, computers were running DOS or Windows 3.1, i.e. 16-bit programs, and as a consequence, ODBC was also 16-bit. Then came 32-bit programs and it got messy: *You could not use the 16-bit ODBC with your 32-bit programs* – you had to use the 32-bit ODBC. But at least there were two icons for the two different ODBC:s in the control panel, so it was clear what you needed to do to configure the right driver.

.....

Finally, a couple of words on how it is that QlikView can use both 32- and 64-bit ODBC drivers: The QV.exe itself never connects to ODBC. Instead it launches a separate process. Depending on whether CONNECT32 or CONNECT64 is used, QVConnect32.exe or QVConnect64.exe is launched, which connects to ODBC and streams the data in QVX format to the QV.exe.

```

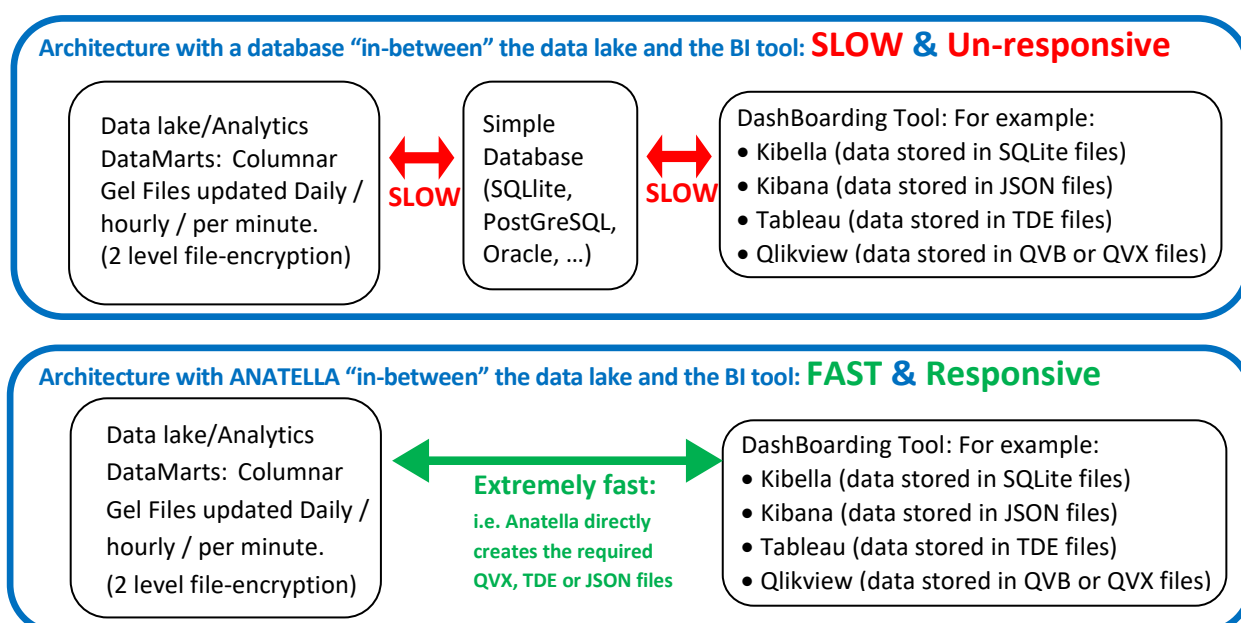
    graph LR
      QV32[QlikView 32] --- QVX((QVX))
      QV32 --- QVConnect32[QVConnect32]
      QV32 --- QVConnect64[QVConnect64]
      QVConnect32 --- ODBC32[32-bit ODBC]
      QVConnect64 --- ODBC64[64-bit ODBC]
  
```

With this solution it is possible to use 32-bit ODBC drivers together with your 64-bit QlikView.

18066 Views Tags: odbc, connection_wizard, 64-bit, 32-bit, force_32_bits, connect_string, qvx, qvconnect, oledb

In particular, you can read in the above article: "*QlikView itself never connects to ODBC (i.e. the QlikView process never connects directly to the SQL database), instead it launches a separate process ... that connects to ODBC and streams the data in QVX format to the QlikView process*". We now understand better the great interest of creating directly, and at very high speed, the famous QVX files rather than using ODBC to make a slow and unreliable data extraction out of your database.

This chart summarizes the interaction between a “data lake” and a “BI tool” inside two different infrastructures: i.e. inside an architecture based on a SQL database “in-the-middle” and inside the proposed optimal architecture based on Anatella:



3.3. Scheduler: Jenkins

At one point, you might have so many jobs (so many data-transformations and scoring) running on your “Production Server” every night that you might need to add a second (or even a third) “Production Server” to still be able to compute everything during the short time-span of the night. To manage all the jobs running on the several different “Production Servers”, one easy solution is to use “Jenkins”: See the section 4.8.7.2. of the “AnatellaQuickGuide.pdf” to have more information about the integration between “Jenkins” and Anatella. Here is an extract of this section:



Jenkins can transparently manage a fleet of many computers (i.e. it manages many “nodes” in technical terms). When Jenkins needs to run a job, Jenkins can easily connect to an “idle” node and run the required job there (in technical term, this is called “distributed computation”). This gives to the final user/company a tremendous computing power: There are actually no limits to the delivered computing power: if you need more computing, simply adds some more “nodes”.

4. Summary on the optimal infrastructure

You will find on the next page a chart that summarizes the proposed architecture.

