Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

Data intensive ATLAS workflows in the Cloud

To cite this article: G F Rzehorz *et al* 2017 *J. Phys.: Conf. Ser.* **898** 062008

IOP Conf. Series: Journal of Physics: Conf. Series **898** (2017) 062008 doi:10.1088/1742-6596/898/6/062008

Data intensive ATLAS workflows in the Cloud

G F Rzehorz[1,2] on behalf of the ATLAS Collaboration, G Kawamura[1], O Keeble[2] and A Quadt[1]

[1] II. Physikalisches Institut, Friedrich-Hund-Platz 1, 37077 Göttingen, Germany
[2] IT Department, CERN, CH-1211, Geneva 23, Switzerland

E-mail: `g.rzehorz@cern.ch`

Abstract. This contribution reports on the feasibility of executing data intensive workflows on Cloud infrastructures. In order to assess this, the metric ETC = Events/Time/Cost is formed, which quantifies the different workflow and infrastructure configurations that are tested against each other. In these tests ATLAS reconstruction Jobs are run, examining the effects of overcommitting (more parallel processes running than CPU cores available), scheduling (staggered execution) and scaling (number of cores). The desirability of commissioning storage in the Cloud is evaluated, in conjunction with a simple analytical model of the system, and correlated with questions about the network bandwidth, caches and what kind of storage to utilise. In the end a cost/benefit evaluation of different infrastructure configurations and workflows is undertaken, with the goal to find the maximum of the ETC value.

1. Introduction

In the context of this paper, Cloud computing only covers the usage of Infrastructure as a Service (IaaS) from public/commercial providers. The potential benefits are difficult to quantify, since the Cloud's impact on a workflow's performance is not well understood. An additional difficulty is that there are many different workflows that are run simultaneously on each of the Worldwide LHC Computing Grid (WLCG) sites and therefore on the Cloud (as a site extension, or site of its own). The different workflows include Monte-Carlo simulations, which are not data intensive and running them on the Cloud is mostly understood. Another type of workflow is physics analysis, since these consist of user generated code, the performance is difficult to evaluate. This paper therefore focuses on raw data reconstruction workflows. The findings can be easily translated into any other data intensive workflow, as long as its resource usage and requirements are known. Section 2 takes a look into a reconstruction job, highlighting the required information. Once the workflow is fully understood, there might be optimisations that can be applied in order to increase the performance. One such optimisation is the CPU overcommittment which is detailed in section 3. Overcommittment means to have more processes running simultaneously than there are CPU cores available. In order to predict how a Cloud site will perform, the model in section 4 is being created. The idea is that the model will represent a graspable output metric, answering many questions in an understandable fashion. This can provide answers about choosing the best Cloud configuration, how to apply optimisations, which Cloud provider to choose, or which workflows to run on an infrastructure. Combining the concepts of the previous sections, section 5 gives some example applications of the Model. The paper concludes with sections 6 and 7.

2. ATLAS Job profile

As already mentioned, this paper focuses on ATLAS experiment [1] raw data reconstruction. In this case Athena(MP) version 20.7.6.7 was used (the exact command can be found in the supplementary data). The job was run on a Virtual Machine (VM) as it would be the case on a Cloud site. The VM was on a hypervisor that was not used by other users, so there was no interference from other jobs. The VM had 8 virtual cores, 32 GB of RAM and a spinning disk. The input data was downloaded from a remote location, before the job was executed.

Figure 1 shows the utilisation of the network, CPU, disk and memory. The plotted data stems from the sar command (part of the sysstat package) which was executed and its output recorded every five seconds.

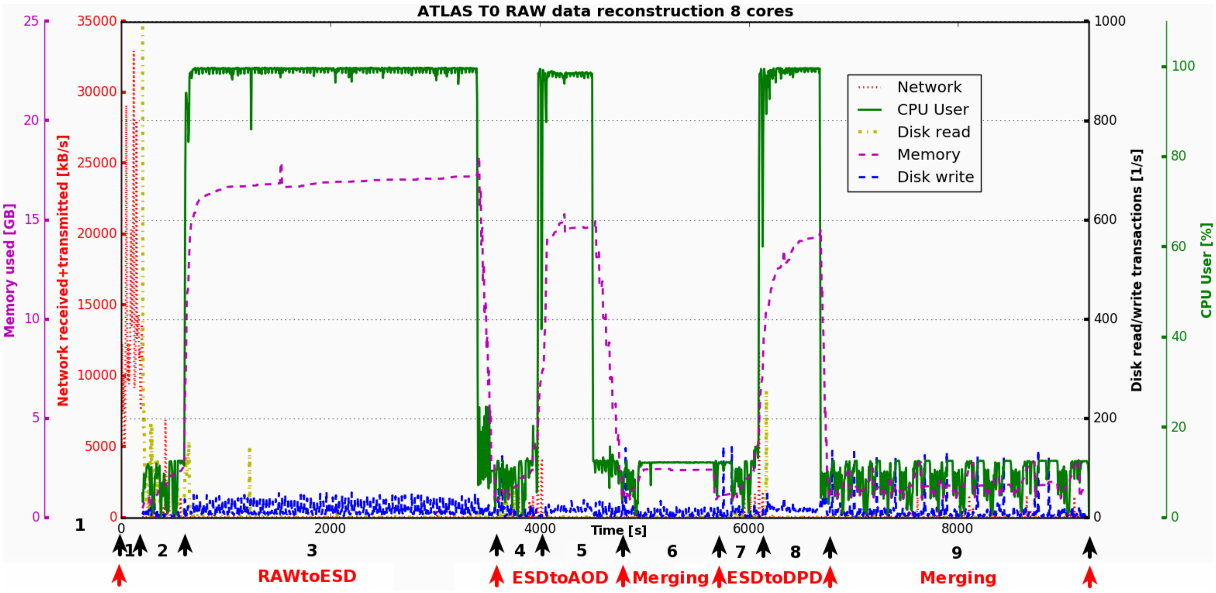


Figure 1. ATLAS Reconstruction Job profile

The plot can be split into several parts, according to different processes happening within the job. The first separation is into transformations (red letters below plot). Each transformation can be split into several substeps according to their resource usage. The (black) numbered areas show distinct resource usages.

The first area depicts the input data download, it shows high network activity (as the job has not started, the other values are not plotted here). The very short second area shows the setup period, where the code and conditions are loaded - not much CPU is used, therefore disk and network activity can be observed. In the third area the first data processing takes place and all eight available CPU cores are fully used. Some disk writing activity is going on throughout this step. The RAWtoESD transformation consists of areas one, two and three. The memory footprint is shown and the 32 GB are more than enough to accommodate the job. In the fourth area, the ESDtoAOD transformation is set up, which is processed in the fifth area. These results are then merged in the sixth area. Another output is produced in the ESDtoDPD transformation. In the end all the final mergings take place. Merging has a special significance, because it uses only a single CPU core but uses a large portion of the overall time. In this case, merging means only 12.5 % of the computing power of the machine is used.

The AthenaMP (MultiProcess) framework that is used to run on multicore VMs, was introduced to save a substantial amount of memory compared to running multiple Athena jobs in parallel. The memory requirement of the processes is well below the total available memory. The processes need even less RAM than can be seen in the plot, which can be easily demonstrated by reducing the available RAM below the peak memory usage from the plot. It can be observed that the job starts to swap out pages, meaning the system transfers some data which is stored in RAM to the disk (disks are much slower than RAM). These pages are not read back in (light swapping) so the job is barely slowed down. A significant slowdown can be observed only after lowering the available RAM even further. Then the swapped out pages are actually needed and heavy swapping in and out of pages slows down the job significantly. This threshold is generally reached at around 1.3 GB RAM per core.

3. CPU overcommittment

CPU overcommittment is a resource optimisation technique. It means to send more work (parallel running jobs) to a computing resource than there are cores. Currently, CPUs are not used 100% during a workflow execution, due to IO-wait (especially for a Cloud site without local storage) and the ATLAS multicore workflow concept AthenaMP (serial merging steps, see job profile). Putting additional workload onto the VMs could make use of the time the CPUs are idle, while at the same time increasing the memory footprint. This trade-off of higher CPU utilisation vs. lower RAM requirements can be hard to implement on a static infrastructure. In the context of Cloud computing, the option to add or remove RAM exists by design, but affects the cost.

Figure 2 shows tabular results from tests performed on the same VM as in section 2. The VM has eight cores and an LHCONE [3] network connection. The available RAM is reduced by having a background application locking 16 GB. The different scenarios that have been tested against each other are the variation of processes (not overcommitted vs. overcommitted), the available amount of RAM (16 or 32 GB) and the different locations of the data (BNL or local). The overcommitting factor of two simplifies the comparison due to AthenaMP restrictions. The data from BNL was read event by event during job execution (events are independent snapshots from the detector containing the results of particle collisions). Local data means it is already in the storage of the VM. Since it takes very little time to download the data from the local site to the storage of the VM (wrt. the job's duration), there is no differentiation between data on VM or local storage. Therefore this test depicts a best vs. worst case comparison. The results in Figure 2 show that overcommitting is very good in latency and/or serialisation hiding, meaning it reduces the overhead when reading data on the fly. Since more processes need more RAM, overcommitting is RAM dependent. Given enough RAM, even the local data scenario benefits from overcommitting (due to the profile of the job). After demonstrating that there are benefits, the ideal RAM-to-core ratio is of interest. It can be obtained by testing many possible scenarios, or by applying the Model from section 4. The Model automatically gives the cost/benefit ratio, whereas it would be tedious to include e.g. hardware cost or Cloud market prices by hand.

4. Workflow and Infrastructure Model

The Model was created in order to answer questions like: For this next Cloud procurement, how much bandwidth is required between the Cloud provider and the data centre?

In order to answer this, many parameters have to be considered, for example the type of workflow (data intensive?), the speed of the CPU and many more. These parameters were either related to the infrastructure or to the workflow and were kept independent of each other. Soon more questions came up that could be answered by the model, or an improved version thereof. In particular the overall performance impact of Cloud site configurations, as well as workflow modifications, was of interest. The Model takes the plethora of workflow and infrastructure parameters as input and generates one graspable output metric. This metric that best describes the performance is ETC = Events/Time/Cost, where Cost depends on the Cloud provider.

ATLAS Real Data Reconstruction					
Number of processes	RAM [GB]	Data location	Overall node throughput [s/event]	Overcommit improvement [%]	Duration improvement to standard [%]
8	32	BNL	4,19 ± 0,05		-32
2x8	32	BNL	2,55 ± 0,01	39	19
8	16	BNL	4,31 ± 0,08		-36
2x8	16	BNL	3,51 ± 0,08	19	-11
8	32	local	3,07 ± 0,04		3
2x8	32	local	2,24 ± 0,01	27	29
8	16	local	3,17 ± 0,09		0
2x8	16	local	3,33 ± 0,01	-5	-5

Figure 2. Overcommittment results summary.

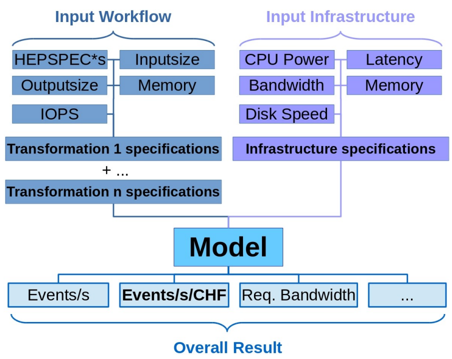


Figure 3. Model outline, depicting the different inputs and possible outputs.

In Figure 3, some of the different in- and output parameters are depicted. On the top left the workflow input parameters are depicted. The field "HEPSPEC*s" makes it evident that they were chosen infrastructure independent, as HEPSPEC06 (HS) [2] is a universal benchmark rating particular to High Energy Physics (HEP).

In Figure 1, the job was split into several transformations, according to the different areas. The Model splits each workflow the same way, whereas multiple jobs in a workflow are seen as series of their consisting transformations. These transformations can differ significantly from one another, as can be seen from the profile. An additional benefit of this modular approach is that the workflow can consist of multiple jobs and the Model accommodates them in the same way. Even switching between different job/workflow compositions is straightforward by adding and removing the necessary transformations.

For different use cases, the overall result can be more than Events/s/CHF. When searching for optimisations on an existing infrastructure, cost does not play a role and the result can be Events/s, which is a measure for the physics throughput (favours fast, usually more expensive hardware). If there is no time pressure, the infrastructure should be optimised for events/CHF, producing physics as cheap as possible (favours cheap, usually slower hardware). In addition, any input metric can become the result, which is useful for cases where infrastructure requirements are unknown, e.g. bandwidth (see section 5.2). This provides answers to questions like: There is a fixed budget, which Cloud infrastructure should be acquired in order to maximise the processed events? Which combination of bandwidth, caches and Cloud storage is the most beneficial? Similarly: What combination of workflows (combination of Simulation, Reconstruction and Analysis) is the best to run on this Cloud?

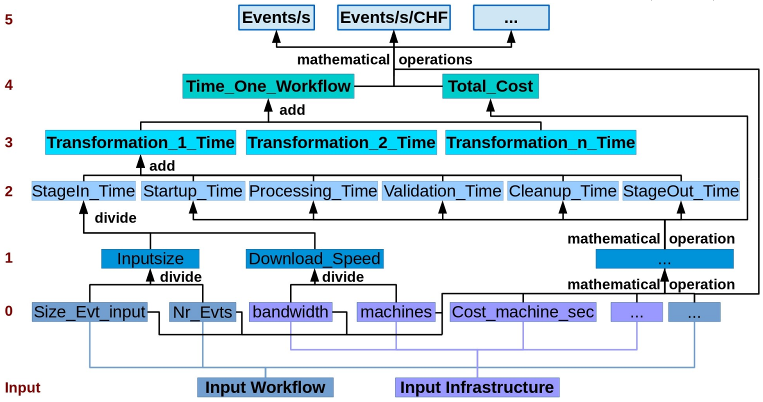
The model has been designed in a very general way, so that all ATLAS workflows, all the other experiments' workflows and even non-physics workflows can be described by it.

4.1. Model description

In Figure 3 the Model concept is sketched. The Model takes and combines the different workflow and infrastructure parameters in order to calculate the desired output metric. A better understanding of the Model can be gained when looking at a more detailed view in Figure 4. The Model consists of a five layer structure, where each layer is combined mathematically into the next layer. In the end all results have to be determined from the workflow duration $Time\_One\_Workflow$ (layer 4), which is a linear combination of the substeps (layer 2) of each transformation (layer 3): $Time\_One\_Workflow = \sum_{Transformations} \sum_{Substeps} Duration$. The substeps correspond to the splitting that has been done in Figure 1.

The example in Figure 4 showcases how the stage-in duration ($StageIn\_Time$) is determined and how it contributes to the overall result: $StageIn\_Time = Nr\_Evts * Size\_Evt\_input/bandwidth/machines$, where Nr_Evts: Number of events being processed per workflow; $Size\_Evt\_input$: Input size of one event [b]; $bandwidth$: Bandwidth connecting a remote storage to the Cloud site [b/s] and $machines$: Number of VMs. This value is added to the transformation and therefore workflow duration. Additional considerations: Instead of downloading the data, it could be read event by event during job execution, which is negatively affected by high latencies (add "event access time"). Hereby, the same bandwidth constraints apply, but the network usage should look relatively flat, whereas the download scenario could look more spiky. This means the prediction is less precise for the download scenario (especially in the beginning, when all jobs download data at the same time - until the downloads are spread apart).

Figure 4. Detailed Model description, schematically visualising its logic and mathematics in six layers (0 - 5).



In the same manner as for the $StageIn\_Time$ all values from the second layer are determined, the detailed description of each item follows.

A similar calculation as for the stage-in duration is done for the stage-out duration ($StageOut\_Time$), replacing input with output and download with upload.

The startup duration ($Startup\_Time$) consists mainly of loading the code into memory, some small checks on the input data and retrieving the ATLAS metadata. This substep has a short duration compared to the other parts and is not influenced heavily by the infrastructure, therefore it is considered to be constant.

The generally most time consuming substep is the processing: $Processing\_Time = (CPU\_time\_overall + CPU\_wait\_time + Idle\_Time)/Nr\_Cores$, where CPU_wait_time: Time [s] the CPUs are waiting for input; $Idle\_Time$: CPU idle time [s] due to a job using less cores than available, e.g. during merging; Nr_Cores: Number of Cores per VM and $CPU\_time\_overall = (CPU\_time * Nr\_Evts)/CPU\_Power$: Overall CPU consumption time (CPU_Power: CPU Power [HEPSPEC]; CPU_time: Time the CPUs are spending on instructions (normalised by CPU Power) per event [HEPSPEC*s]). The time the CPUs are waiting (CPU_wait_time) is complex, it can consist of swapping ($Swap\_Time$) and I/O wait time ($IOWait\_Time$).

Swapping happens when a transformation requires more RAM than is currently available. Additional complexity enters, because a differentiation between light and heavy swapping has to be made (see section 2). The RAM discrepancy increases the transformation's duration (not necessarily in a linear fashion). Another transformation might not be affected by the same RAM limitations, because of a smaller memory footprint. There has been some effort in describing the exact behaviour mathematically, but since the goal is to find the maximum event throughput which will never be in this region, it is not included in the Model. The heavy kind of swapping is penalised severely by setting the $Swap\_Time$ to a high value (penalty) as the infrastructure cannot accommodate the workflow optimally, if $RAM\_machine < nr\_processes * RAM\_per\_processes$, where $RAM\_machine$: RAM per VM [b]; $nr\_processes$: Number of parallel processes (AthenaMP option); $RAM\_per\_processes$: RAM per process

required by the workflow [b].
The time the CPUs are waiting for I/O operations to be performed, is called I/O wait time. Different kinds of disks (SSD, HDD) are better(worse) suited to handle swapping and I/O operations as they can handle more(less) read/write operations per second. The exact impact of the disk speed on the I/O wait time is under investigation and only the HDD scenario is implemented (as being constant) so far. Since the major difference in speed is not between different disks of the same type, but between SSD and HDD, the implementation will probably choose between two scenarios (fast, slow).
Validation and clean-up are not infrastructure dependent and even shorter than the start-up substep and they are also considered constant. All the durations of the substeps (layer 2) of all transformations (layer 3) sum up to the overall duration (layer 4).
Before continuing to the final result (layer 5), as mentioned in section 3 one purpose of the model is to investigate and test optimisation scenarios. In order to get further results for the overcommittment scenario, the Model has to be modified. The memory limitation discussed for the processing is a crucial part, because overcommitting increases the overall memory footprint. In case overcommittment happens, part of the additional workload can make use of the time the CPUs are idle, resulting in a higher CPU utilisation. The major change is the aggregation of all CPU utilisation and idle times, to consider them for the whole workflow (instead of for each substep). The overall processing time for overcommittment $Processing\_Time_{OC}$ is determined by: $Processing\_Time_{OC} = \sum_{Transformations}(CPU\_time\_overall + CPU\_wait\_time + Idle\_Time - OC\_Factor * (nr\_processes - Nr\_Cores)/nr\_processes * (CPU\_time\_overall + CPU\_wait\_time))/(Nr\_Cores)$. The overcommittment factor OC_Factor is a measure of which fraction of the overcommitted processes can be computed in parallel with the non-overcommitted processes (making use of the idle and CPU wait time). It is percentage based and ranges from 0 to 1. A value of one means it is possible to use all of the CPU wait and idle time to compute the additional processes. If there is no overcommittment ($nr\_processes \leq Nr\_Cores$), the overcommittment factor $OC\_Factor = 0$. The overall CPU time ($CPU\_time\_overall$) as well as the CPU wait time (CPU_wait_time) have to include the overcommitted processes. The first part of the equation is the same as before. What changes is that the time fraction that is gained from overcommitting through parallel processing is subtracted. The difficulty is to determine the overcommittment factor. Work is under way to describe it in dependence of the input parameters.

In order to get to the overall duration of the workflow (layer 4), the processing time is added to the sum of all the other substeps (layer 2). The ETC result (layer 5) is calculated in the following way: $Events\_time\_cost = (Nr\_Evts * machines)/(Time\_One\_Workflow * Cost\_machine\_sec)$. This metric is especially useful when considering Cloud providers. Some Cloud providers charge for data transfer. Including this cost reduces the events/cost ratio for all infrastructure configurations by the same amount.

The Model is kept as simple as possible. For most parameters it is possible to go further into detail. This may be done once the Model is validated and has error estimations. There is a trade-off between keeping it simple and making it accurate. The benefits of a simpler model are that it is applicable without expert knowledge of the workflow and infrastructure and that it is more accessible to other experiments/users. Especially for Cloud computing, where some infrastructure aspects are unknown, a less complex model may be the only option. Disadvantages are that it might not be applicable to some special cases/configurations.

5. Model application

5.1. Overcommittment

One possible Model application is to find the best optimisation parameters. In the case of overcommittment, the parameter space that has to be explored is the amount of RAM against the number of processes. The variation of RAM is necessary, because the memory footprint changes according to the number of processes. Additional RAM comes at a cost, which is modelled the following way: There is a flat budget and fixed amount of cores per VM. Additional RAM therefore means less budget for other parts of the infrastructure (CPUs), which means fewer VMs. The Model has been adapted accordingly, whereas the RAM-to-CPU pricing ratio has been taken from the pricing scheme of a Cloud provider. This can be adapted to a particular provider, or be exercised over several providers (hardware types) to get a comparison.

Figure 5 shows the result of the Model. The maximum ETC value represents the configuration of processes/RAM that should be taken in order to maximise throughput and minimise duration and cost. This plot gives a continuous depiction of all possible scenarios, something that would have taken a long time to fill with hundreds of measurements.

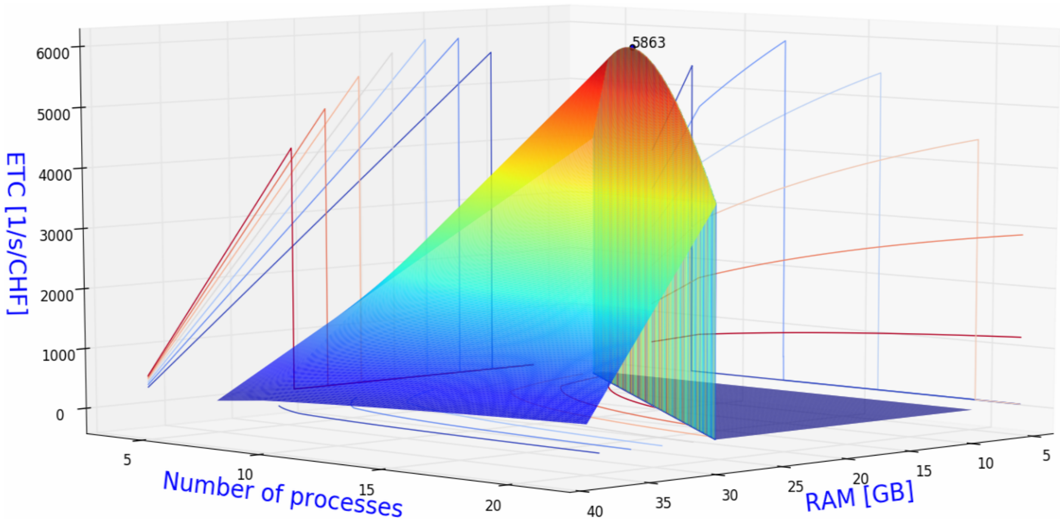


Figure 5. Graphical display of the Model output for 8 core VMs. The maximum is highlighted.

In the estimation (of the unfinished Model) in figure 5, the current 2-to-1 ratio of RAM [GB] to CPU [core] would not be the optimum. The maximum of ~ 5863 Events/s/CHF lies at 14 GB RAM per machine with an overcommittment of 11 processes/machine.

5.2. Bandwidth estimation

Another application is to predict the overall bandwidth requirement of a Cloud site. In principle any of the input parameters can be modelled. In this example, the concern was that the bandwidth between the Cloud provider and external storage would not be enough, meaning CPUs would be constantly idle, because they are waiting for the slowly downloading input data. The question to be answered was, whether a Cloud site consisting of 4000 CPU cores can run ATLAS raw data reconstruction efficiently, while being connected by a 10 Gb WAN link. The Model showed that the link would be enough for the expected specification (1000 x 4 core VMs, 116 HS*s/evt CPU time, 850 kB/evt input, 2701 Evts per Job and 1417 kB/s instantaneous network read). However, it was not a hundred percent clear what kind of infrastructure the Cloud provider would supply. In addition, it has happened before that the ATLAS experiment changed their software or job configuration or the data itself changed. In order to understand these scenarios, some parameters were varied and thereby their impact evaluated. This could help to prepare for future or worst-case scenarios. Figure 6 shows the result.

The black horizontal line depicts the 10 Gb/s bandwidth limit (Bandwidth_Limit). Various parameters of the Raw data reconstruction workflow were varied in order to include future

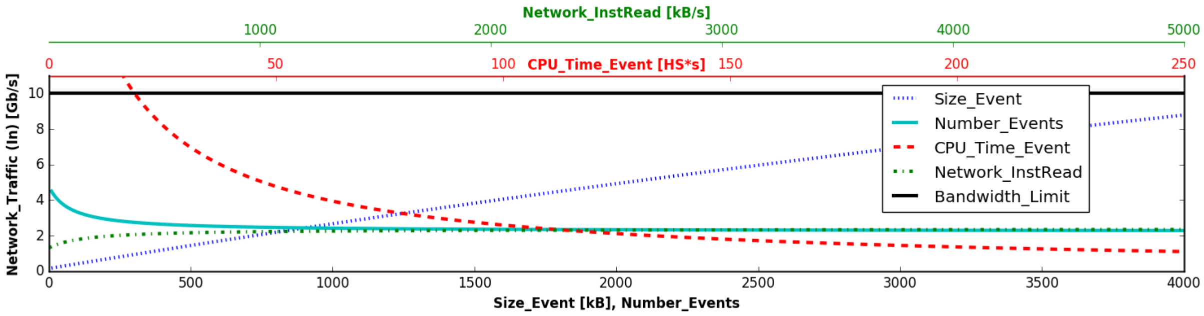


Figure 6. Infrastructure limit and Model output.

changes to the workflow or the input data. The plot shows that within the chosen range, the size of the events (Size_Event), the number of events per job (Number_Events) and the instantaneous bandwidth (Network_InstRead) do not have such a high impact on the bandwidth requirement (Network_Traffic) as to make it exceed its limit. The instantaneous bandwidth is important when reading the input file event by event.

The variation of the CPU time per event (CPU_Time_Event) on the other hand, could become a problem if it goes below ~ 30 HS*s. This could happen either if the events become less complex, which is unlikely as the complexity is rising along with the pileup. The second factor which could reduce the processing time is to have faster CPUs. Regarding the technological evolution of the last years, the scale of this progression is too small for this to have that high of an impact.

6. Future Work

The Model is in the process of being validated. This will be achieved by modelling and then comparing the data gained from recent CERN Cloud procurements with data from the CERN computing centre and personal VMs (controlled environment) at CERN and Göttingen. Furthermore there will be error estimations, which will point to the largest error sources, that may be eliminated by a more in-depth description. In addition more optimisations will be investigated, especially scheduling and caching.

7. Conclusion

The concept of Cloud computing brings challenges but also opportunities. Flexible hardware allows hardware adaptations to the workflows, like overcommitting. Understanding the possible gains of using Cloud computing or different optimisation techniques can be difficult. Therefore it is important to have a deeper knowledge of the workflow itself. The Model can help to describe and choose different workflow and infrastructure combinations, as well as the "best" commercial Cloud provider. The Model depicts correlations between parameters and finds the impact they have on each other. It assists when planning for future changes or worst case scenarios.

References

[1] ATLAS Collaboration 2008 The ATLAS Experiment at the CERN Large Hadron Collider *JINST* **3** S08003
[2] Michelotto M *et al* 2010 A comparison of hep code with spec 1 benchmarks on multi-core worker nodes *J. Phys.: Conf. Ser.* **219** 052009
[3] Martelli E and Stancu S 2015 Lhcopn and lhcone: status and future evolution *J. Phys.: Conf. Ser.* **664** 052025

Acknowledgements

Work sponsored by the Wolfgang Gentner Programme of the Federal Ministry of Education and Research.