First results from a combined analysis of CERN computing infrastructure metrics

Dirk Duellmann\textsuperscript{1} and Christian Nieke\textsuperscript{2}
\textsuperscript{1}CERN, Switzerland
\textsuperscript{2}Technical University Braunschweig, Germany
E-mail: dirk.duellmann@cern.ch

Abstract. The IT Analysis Working Group (AWG) has been formed at CERN across individual computing units and the experiments to attempt a cross cutting analysis of computing infrastructure and application metrics. In this presentation we will describe the first results obtained using medium/long term data (1 months — 1 year) correlating box level metrics, job level metrics from LSF and HTCondor, IO metrics from the physics analysis disk pools (EOS) and networking and application level metrics from the experiment dashboards. We will cover in particular the measurement of hardware performance and prediction of job duration, the latency sensitivity of different job types and a search for bottlenecks with the production job mix in the current infrastructure. The presentation will conclude with the proposal of a small set of metrics to simplify drawing conclusions also in the more constrained environment of public cloud deployments.

1. Introduction
Extracting science results from raw LHC measurements and simulations depends today crucially on complex, distributed computing infrastructures and efficient data management and resource allocation. Over the first LHC run periods significant optimisations have be implemented to this system in close collaboration between the LHC experiments and WLCG, which have allowed to keep up with the increasing demands on the physics side and to fully exploit the excellent LHC machine performance. During the upcoming LHC run periods including the High-Luminosity program an even larger efficiency increase will be required as both Moore’s and Kryder’s “law” are showing strong signs of a beginning reduction of their respective transistor and storage density predictions due to physical limits of the current silicon and magnetic storage technologies.

2. Working group motivation
In order to prepare for a more systematic optimisation activity across different activities a working group has been formed across all individual services in the CERN IT department to organise the collection of metrics and their analysis with the goal to have a more complete quantitative picture of the resource utilisation across the computer centre and to optimise the science throughput per financial investment. The focus of this activity is explicitly on contract an end-to-end model across different component services taking into account also experiment workflow information such as well defined physics tasks to optimise the aggregate processing rate rather than individual benchmark jobs. Emphasis is placed on medium to long term
(weeks to years) analysis rather than the observation of short term (hours to days) trends that is already performed as part of the operational monitoring by many service and experiment groups. As direct consequence, this longer term analysis requires a larger active volume of combined and aggregated metrics from a larger variety of sources and data formats than most existing monitoring and accounting systems. For this reason the AWG was also involved in defining a more modern collection and aggregation system for infrastructure metrics described in more detail in[2]. From the analysis technique point of view, this combined longer term studies quickly go beyond simple time series observation and often require statistical aggregates and distributions, cross system joins and the development of statistical models. The different challenges of typical analysis phases are described in the following section.

3. Analysis phases
The data analysis of infrastructure metrics, quite similar to e.g. physics analysis, usually passes through a set of phases, which are characterised by increasing data quality and understanding of the metrics.

3.1. Metrics collection and statistical description
At the most basic level one need to check the completeness of input metrics and their extreme values to expose measurement or transport problems that have not been spotted on operational dashboards. This first steps also includes the data conversion from unstructured logs or semi-structured textual formats like XML or JSON to more efficient analysis formats. The result of this first step is a simple characterisation of the status quo of a single system with the basic tools of descriptive statistics: averages, variance and extreme values. The second phase is usually to exploit intrinsic data redundancy in order to check the predictive value of the data. Quite often measurements are collected with high numerical precision, but the actual measurement shows much lower resolution. This can often be spotted by looking at the width of metric distributions for measurements that are either frequently repeated or otherwise constrained. These methods
are well known in physics data analysis but not always fully exploited in infrastructure analysis studies. The main outcome of this analysis phase is the quantitative statement on the accuracy of the measured data and its related predictive value. Since many simpler monitoring applications only refer to relative metric changes (e.g. “worse” or “better”) this step will also expose some misunderstandings about metric units and formats (CPU utilisation in seconds, fractional minutes, concatenated “hh:mm:ss” strings). In addition most metrics are expected to obey semantic constraints like sum rules. For the total CPU seconds spent in a batch system one may expect that the sum of all jobs is close to the amount of scheduled CPU and also to the total amount of available core wall time on all configured hosts.

\[ \sum_{cpu_{job}} \approx \sum_{cpu_{sched}} \approx \sum_{cpu_{host}} \] (1)

For a disk system one may assume that the transferred bytes of all disk IOs will approximate the sum of all IOs in user jobs plus the sum of IO logged for internal storage system tasks like replication or re balancing.

\[ \sum IO_{disk} \approx \sum IO_{user} + \sum IO_{internal} \] (2)

Even if these constraints represent only approximate relationships — their confirmation or the observation of significant violations is required to establish a quantitative understanding of the system behaviour.

3.2. Combining data and abstraction

Even more value than from individual metrics can be extracted by connecting measurements from several subsystems — in particular by connecting end user metrics with infrastructure metrics. One example is the bottleneck analysis of a larger ensemble of user jobs that are processing data from a shared data service (e.g. remote files or DB access). In this case it is often difficult for a single user or an individual service responsible to evaluate the efficiency of the total system: For a single user it is easy to quantify the absolute throughput of their individual job, but this usually does not allow to judge, if the system as a whole is working effectively. For a service responsible of e.g. the batch service it is easy to judge the utilisation of the set of CPUs (e.g. by comparing CPU and wall seconds) but this utilisation alone does not allow to conclude on the reasons for incomplete core utilisation: are the currently running jobs expected to be CPU bound? Is there a bottleneck in their associated remote IO paths that limits their current CPU utilisation? Is the IO bottleneck related to data bandwidth or rather to IO latency? In order to address questions like this, one needs to combine metrics from different subsystems and to systematically conclude on the aggregate behaviour of the composite system of CPU, storage and user access pattern.

In order to facilitate this connection, the analysis typically will require information on the system topology (which job ran on which CPU server accessing which storage replica over which network path). In addition to this time dependent, but usually low volume topology information one also needs to include static quantitative performance ratings for the heterogeneous components, such as ratings for CPU, disk and network interface speed. Examples for results obtained by connecting the information of end user throughput and specific hardware type are shown in figure 2. The plot shows the achieved CPU utilisation for a number of very similar experiment jobs running in the heterogeneous CERN batch system. The coloured dots show the CPU and wall times for different hardware configurations. The left plot shows a clear correlation of the CPU utilisation with the hardware type. While some hardware types result in reliably efficient CPU utilisation (e.g. close to the ideal diagonal line), there are clearly other hardware setups, which typically feature low CPU utilisation.
To investigate the correlation with different network topologies figure 3 shows the same job population, but is this time coloured by the nature of the IO client to server connection. The blue entries have been by jobs using a WAN connection (client process and input file replica server have been located at different sites). The red entries have been using the local area network (LAN) within one site. From this plot we can not associate the reduced CPU utilisation with the type of network connection.

To further understand the inefficiencies we have compared the job distributions for each of the hardware types that this production has used. In the box plot shown in figure 4 we can see again large differences in not only in average CPU utilisation, but also very different width and extreme values for the different hardware types, which confirm a stronger correlation of the CPU utilisation with the hardware configuration.

In the limited space of this contribution we can only show a few examples of still a basic statistical analysis, but we hope this shows already the potential of applying simple methods to
combined system and user metrics to understand and optimise larger heterogeneous computing systems.

3.3. Model predictions and interpretation

The analysis phase following basic accounting focuses on constructing an model, with the aim to predict future system behaviour and to allow the evaluation of different system configuration options without the need realising all alternatives. Again, an approach conceptually close to the use of e.g. Monte Carlo detector models to study design and implementation alternatives. Such a system model is often constructed from an assumed or measured relationship between system parameters (e.g. system topology, CPU count, storage size etc), user workload and measured throughput.

A first order model for CPU bound processes would for example assume that the amount of processed server workload depends linearly on the CPU speed rating (e.g. measured as
Figure 6. Prediction accuracy: RMSE between predicted and measured job CPU time

HEPSPEC rating) of the processing core. For jobs with similar CPU and IO pattern this would allow to first obtain the measurement of the CPU rating from a set of training jobs and then to predict the throughput or alternatively the job duration for a known amount of processing work. This approach is conceptually very close to the process of calibrating the response of a particle detector with a set of well known events and then using this calibration constants to measure unknown event shapes. This approach has been applied to large set of ATLAS and CMS production jobs from the CERN batch system, using the experiment task identifier to define job groups that are assumed to behave similarly. The detailed data selection and processing, which involves fitting a large set of results for different job types on different hardware models will be described in[3].

Preliminary results show a predictive power similar or better than results obtained with traditional methods such as CPU benchmarks, which are far more resource consuming and less applicable in more volatile cloud scenarios. Since this method does allow to obtain a fairly precise CPU speed classification without the need to run specific benchmark programs it has been named “passive benchmark” approach. Figure 6 shows a comparison of the accuracy of this passive benchmark approach compared to HEPSPEC, for scenarios where the training and test data consists of only stable production jobs, only more varied user analysis jobs, or a mix of both. The precision of the approach can be seen in figure 7 and is well within the acceptable range for practical applications. The option to perform benchmarks at virtually no cost anytime even after the fact now allows for applications that would be impossible, or prohibitively expensive with normal active benchmarks. For example, in figure 8 we show snapshot estimates of the CPU factor for a given host, using passive benchmarking on a sliding window that considers only the last 4 jobs. The plot shows roughly 3 different levels of performance of the same host, which we were able to correlate with actual incidents: normal operations (CPU factor ∼ 2.5), one technical problem followed by a reboot (CPU factor ∼ 2) and one case where a sister VM on the same hypervisor was down, leaving all hardware resources for the remaining host (CPU factor ∼ 3.1). As opposed to typical raw metrics used for monitoring (e.g. CPU utilisation), such a performance plot allows to see how well a machine is working from the user perspective, i.e. how well it is processing the actual jobs, rather than how busy a given component is.
Figure 7. CPU prediction: error distribution

Figure 8. CPU prediction: evolution over time (represented here as job execution index)
4. Summary
In this short paper we have described first examples for applying statistical methods to combined computing infrastructure metrics from the analytics working group metrics repository at CERN. We have described the different phases of data cleaning and data quality checking, the combination of operational metrics from different subsystems and configuration databases. We further have presented first examples of new modelling techniques that will help to increase the quantitative understanding of the increasingly complex and volatile computing infrastructures required for high energy physics data processing.

Acknowledgements
Parts of this work were sponsored by the Wolfgang Gentner Programme of the German Federal Ministry of Education and Research.

References
[1] Analysis of CERN computing infrastructure and monitoring data, Nieke C, Lassnig M, Menichetti L, Motesnitsalis E and Duellmann D Proceedings of the 21st International Conference on Computing in High Energy Physics (Okinawa)
[2] Hadoop and friends — first experience at CERN with a new platform for high throughput analysis steps, Duellmann D, Surdy K, Menichetti L, Toebbieke R, These proceedings.
[3] Quantitative performance evaluation of large scale computing infrastructures using passive benchmarking, Nieke C, Doctoral Thesis at Technische Universitaet Braunschweig, In preparation