# 論文の内容の要旨

論文題目　Research on efficient resource utilization in data intensive distributed systems
（データインテンシブ分散システムにおける高効率計算機資源利用に関する研究）

氏　名　グロート　スヴェン

In recent years there has been an unprecedented growth in the amount of data being gathered worldwide. Data may include business data, sensor data, web data, log data, and social data. Data volumes of terabytes are common, petabytes are not unusual anymore, and it is almost certainly not long before exabyte scale data becomes the norm.

Big data presents an opportunity for in-depth analysis. Trends and patterns can be derived from the data, enabling a wide variety of business and research opportunities. As a result, the ability to process large amounts of data—often too large to be stored and processed by traditional relational database systems—has become very important. Processing big data requires very large scale resources such as large clusters of commodity machines that allow the data to be processed in a distributed fashion. Google's MapReduce framework and its open source implementation Hadoop have emerged as the de-facto standard data processing platform for such environments.

At the same time cloud computing has emerged, enabling users to dynamically provision computing resources and pay only for usage on a pay-as-you-go basis. Cloud computing enables anyone to gain access to large scale computing resources without the need to create their own infrastructure, drastically driving down the cost and making it possible for even individuals or small organizations to afford the resources necessary for processing big data. Cloud computing adoption has been driven by advances in virtualization and the wide-spread availability of high-bandwidth network connectivity.

When dealing with resources at this level of scale it becomes increasingly difficult to efficiently utilize all resources. With the billing model used by most cloud providers, provisioning a single node for ten hours costs the same as provisioning ten nodes for one hour, so it would be highly desirable if distributed applications could achieve a linear speed-up when more nodes or nodes with higher hardware specifications are provisioned. However, a number of factors limit the scalability of these applications so they cannot fully use the provisioned resources. In the cloud, inefficient resource utilization directly leads to higher costs for no additional benefit.

In this thesis, I investigate two major causes that contribute to the inability of data intensive distribute applications to efficiently utilize resources in cloud environments or other large scale distributed systems, and propose methods to mitigate their effects.

The first major cause of inefficient resource utilization is workload imbalance. This occurs when certain workers have a longer execution time than others, leading to stragglers: a handful of tasks that hold up an entire application. When a workload has stragglers, some computing resources provisioned for the workload may be idle, or partially idle, while waiting for the stragglers to finish. Because these resources are not being utilized they are unable to assist in speeding up the processing of the workload.

Workload imbalance and stragglers can have several causes such as data skew, processing skew and performance heterogeneity. Data skew occurs when the data to be processed is not divided evenly among the workers. Processing skew occurs when certain records in the data---even if they are not bigger than others---take more processing time. Performance heterogeneity can be caused by differences in the hardware between nodes, but also by environmental factors such as background processes active on certain machines, or interference between virtual machines when they are sharing the same physical host.

It is possible to mitigate data skew—and to a lesser extent, processing skew—by sampling the data beforehand, but these approaches cannot deal with performance heterogeneity. Performance heterogeneity is more complicated to account for beforehand, because precise knowledge about the hardware configuration is usually not available in the cloud and even identical instances can have very different performance. Although measurements can be used to establish the performance of each node, this can still not account for environmental factors that may change during the execution of the workload.

I propose a method called Dynamic Partition Assignment, which is able to dynamically adjust data distribution among workers whenever imbalance is detected. The workload is divided into many more partitions which are dynamically reassigned to workers that have already finished. Dynamic Partition Assignment avoids the overhead of doing many small transfers by assigning partitions in groups that can be transferred in one operation whenever possible. Because imbalance is lazily detected by monitoring the completion times of workers Dynamic Partition Assignment is able to handle data skew, processing skew and performance heterogeneity without any prior knowledge about the data or hardware environment.

Dynamic Partition Assignment was implemented in Jumbo, a MapReduce-style experimental data intensive distributed processing platform created for the purpose of experimenting with workload imbalance, and evaluated both for data skew and by running workloads on a heterogeneous environment. Dynamic Partition Assignment was able to successfully reduce the effects of stragglers, in some cases improving processing times by 30 to 50%, and bringing the processing time to within 10% of the optimally balanced processing time.

The second major cause of inefficient resource utilization is I/O interference. When multiple applications are accessing the same resource simultaneously, they can interfere

with each other causing performance degradation. This is particularly a problem for I/O resources such as disk storage, which are often shared between processes and can have a dramatic reduction in performance when under contention. For example, when running parallel processes to exploit the many CPU cores of modern server nodes these processes contend for the same, more limited, I/O resources. The interference caused by this leads to sub-optimal utilization of both the node's CPU and disk resources.

The nature of I/O interference means that it is very difficult to mitigate after it is detected to occur. Reassigning work often requires additional I/O to move data to new locations, exacerbating the problem. Therefore, it is desirable to be able to predict the effects of I/O interference before it actually occurs so that scheduling and resource provisioning can be adjusted accordingly.

I propose a cost model that is able to predict the effects of I/O interference when multiple MapReduce tasks running on the same node contend for that node's I/O resources. The model uses several workload parameters measured directly from running a subset of the workload, and a number of hardware parameters derived from micro-benchmarks, including hardware specific interference functions that describe how the storage devices behave under contention. These parameters and functions are used by an analytical model of MapReduce, which uses knowledge of MapReduce's processing flow and I/O patterns to predict the performance of the workload when using specified numbers of parallel processes.

The I/O interference model was evaluated against several representative MapReduce workloads and was able to predict their performance to within 5 to 10% even for highly I/O intensive workloads or workloads that use a combination of I/O and CPU intensive processing. The information provided by this model can be utilized, for example, to determine how many nodes to provision with how many CPUs, and how many tasks to run simultaneously on any given node. Additionally, this model can be used by a scheduler to decide how to place tasks in the cluster based on their expected effect on I/O performance.

Improving resource utilization in the cloud helps to reduce application execution time, and reduces costs for both cloud providers and users. In this thesis, I address two aspects of this problem: I propose Dynamic Partition Assignment to mitigate the effect of stragglers and improve workload balancing, and I propose a cost model that addresses the problem of I/O interference. While both of these are targeted at MapReduce, the methods used in this thesis are not specific to MapReduce and can be applied to other data intensive applications in the cloud.