Predicting I/O-performance in HPC using Artificial Neural Networks

Jan Fabian Schmid and Julian Kunkel

Universität Hamburg

Goals

- Gain knowledge about the storage system of a super computer for the development of computational models and tools in the future.
- Find a good model that is reliable in predicting performance of HPC-IO with sufficient quality.
- Extract information about the I/O-paths the storage system used for measured file accesses.

Introduction

Tools are demanded that help users of HPC-facilities to implement efficient input/output (I/O) in their programs. It is difficult to find the best access parameters and patterns due to complex parallel storage systems. To develop tools which support the implementation of efficient I/O a computational model of the storage system is key.

For single hard disk systems such a model can be derived analytically [1]; however, for the complex storage system of a super computer these models become too difficult to configure [2]. Therefore we searched for good predictors of I/O-performance using a machine learning approach with artificial neural networks (ANNs). A hypothesis was then proposed: **The I/O-path significantly influences the time needed to access a file.**

In our analysis we used ANNs with different input information for the prediction of access times. To use I/O-paths as input for the ANNs, we developed a method, which approximates the different I/O-paths the storage system used during a benchmark-test. This method utilizes error classes.

Artificial Neural Networks

ANNs are bio-inspired function approximators. An ANN is able to approximate a function that maps given input vectors onto their associated output vectors. This is done by adapting the connection weights in the network, using gradient decent during a supervised learning process (backpropagation). With appropriate topology **ANNs can approximate any continuous functions** on compact subsets of $\mathbb{R}^n$ [3].

Model of the I/O-path

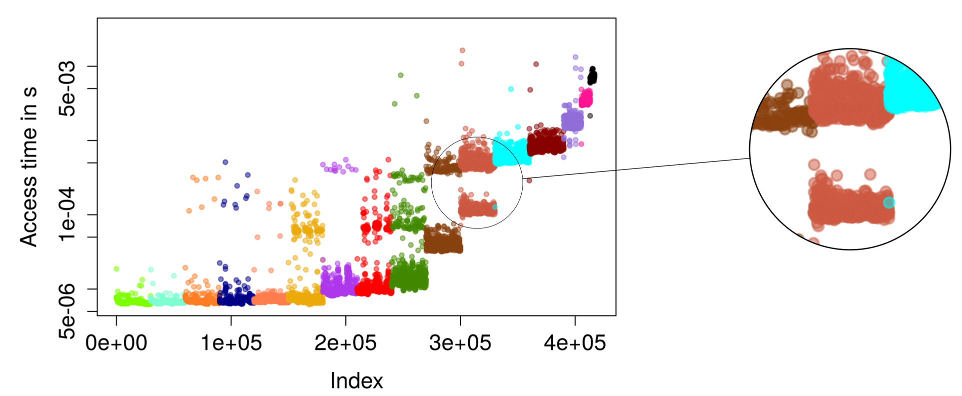
The processing of a file access in the storage system can be viewed using the I/O-path which is **the path from the invoking processor to the storage medium that contains the data**. The resulting access time depends on the depth of this I/O-path because storage media further along the path are increasingly slower. While the first storage levels (Caches) are very quick to respond, the main memory is already magnitudes slower; the same applies for the step into the parallel storage system, that is connected via network to the computer nodes.

Benchmark-Tests

The measurements were done on the super computer Mistral of the DKRZ (Deutsches Klimarechenzentrum). Mistral operates with the parallel distributed file system Lustre and consists of over 1500 computing nodes and 30 petabyte storage. **Two different use cases were tested: Sequential and random file access.** Access sizes varied from 1 B to 16 MiB.

Our measurements have shown, as can be seen in figure 1, that the access times increase with access sizes. However, a phenomenon that can be explained with the I/O-path model can be seen as well: **Access times of measurements with equal parameter values are split into several groups.** The file accesses can be distinguished by a step in the magnitude of access time that is caused by the different I/O-paths the system used.

(a) Sequential access pattern



(b) Random access pattern

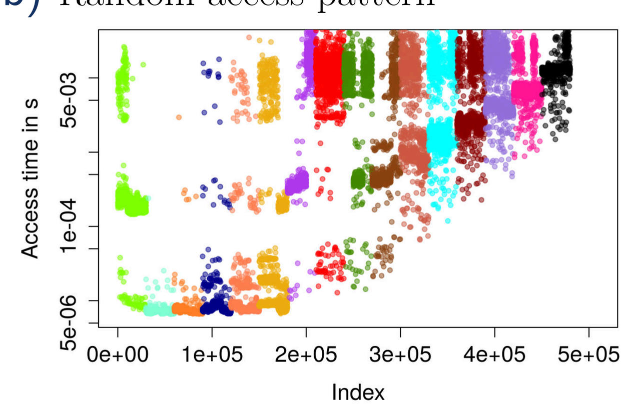
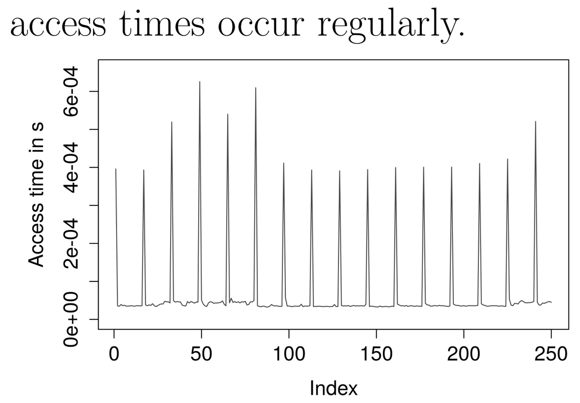


Figure 1: Access times of measured file accesses. The access sizes increase from left to right. All measurements with equal parameter values have the same color. The highlighted area shows a split of measurements with equal parameter values into groups of access times.

Models

Different models were used to predict access times of file accesses. Linear regression was used as a baseline model with a simple mapping of access size to access time. Additionally, three **models with different input information utilizing ANNs were used**. Every ANN-model received information about access sizes, file offsets and access types as input. One ANN also received information about the past data throughputs of the system, which can be used to exploit time dependencies of the I/O-performance (see figure 2). The last ANN received error classes as additional input.

(a) In this sequence, peaks in the access times occur regularly.



(b) Exponential moving average in red to predict peaks.

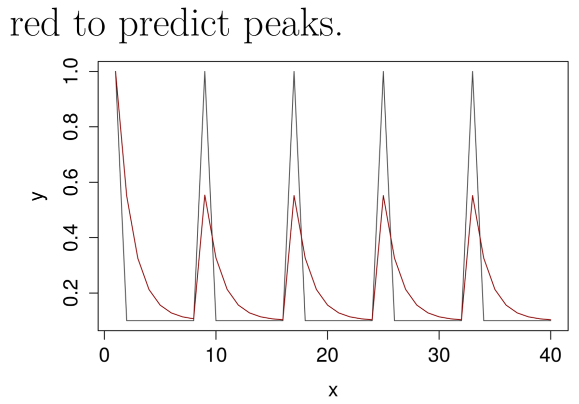
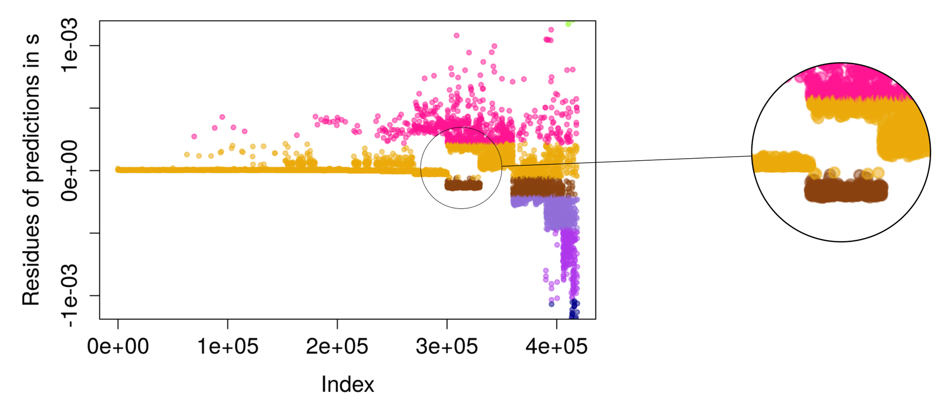


Figure 2: Time dependencies can be utilized for better access time predictions.

Error Classes

The idea of error classes is based on the fact that **the residues of a simple model are characteristic for the groups of measurements with different typical access times** we observed in figure 1. The residues therefore represent the I/O-paths used by the storage system for the corresponding file accesses. Error classes are obtained by clustering the residues with a k-Means algorithm.

(a) Sequential access pattern



(b) Random access pattern

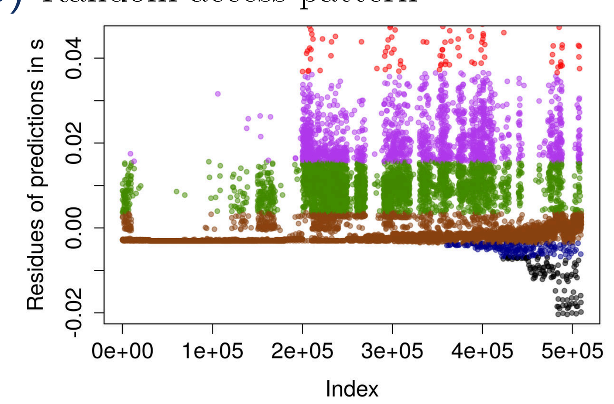


Figure 3: Residues from the linear regression model clustered into error classes. Measurements with equal parameter values can be differentiated through error classes.

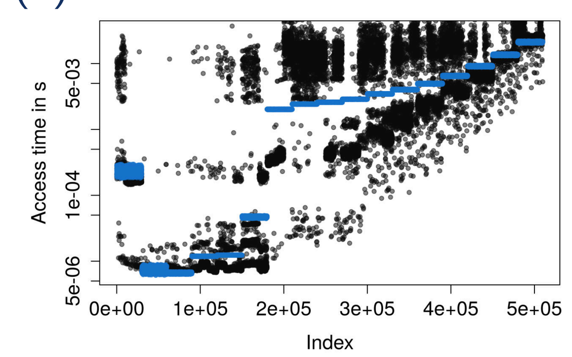
Results

From the residues of our models (see table 1) it became clear that access times shouldn't be modeled linearly. Furthermore, it has been found to be difficult to exploit time dependencies of the storage system. As our hypothesis predicted, **it is essential to use knowledge about I/O-paths for precise predictions of access times.** The error classes allow the ANN to decrease the error significantly. In figure 4 the benefit of error classes as input can be seen: The second ANN is often able to classify file accesses correctly in respect to their I/O-path.

	MAE (s)		MSPE (%)	
Model	seq. access	rnd. access	seq. access	rnd. access
Linear regression	$7.6 \cdot 10^{-5}$	$4.76 \cdot 10^{-3}$	59	14185
ANN	$6.0 \cdot 10^{-5}$	$3.13 \cdot 10^{-3}$	22	530
ANN + time dependencies	$5.7 \cdot 10^{-5}$	$3.05 \cdot 10^{-3}$	22	619
ANN + error classes	$2.0 \cdot 10^{-5}$	$1.03 \cdot 10^{-3}$	14	119

Table 1: Mean absolute errors and mean square percentage errors of the models for the two different file access patterns.

(a) ANN without error classes



(b) ANN with error classes

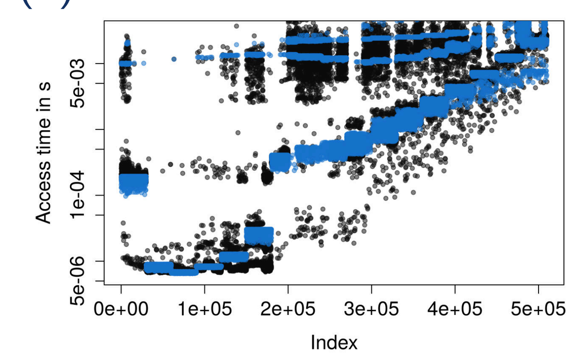


Figure 4: Measurements with random access pattern. Measured access times in black; predictions in blue.

Conclusion

The hypothesis is supported by our data. It is therefore necessary to deduce information about I/O-paths to build a good model of HPC storage systems. If measured access times are available, this can be done with the introduced method.

The method could be a starting point to develop a tool that provides information about I/O-paths used during execution of a program.

References

[1] Chris Ruemmler and John Wilkes.
An introduction to disk drive modeling.
IEEE Computer, 27:17–28, 1994.

[2] Lei Zhang, Guiquan Liu, Xuechen Zhang, Song Jiang, and Enhong Chen.
Storage Device Performance Prediction with Selective Bagging Classification and Regression Tree.
In *Network and Parallel Computing, IFIP International Conference, NPC 2010, Zhengzhou, China, September 13-15, 2010. Proceedings*, pages 121–133, 2010.

[3] George Cybenko.
Approximation by Superpositions of a Sigmoidal Function.
Mathematics of Control, Signals, and Systems, 2:303–314, 1989.

Contact Information:

- 2schmid@informatik.uni-hamburg.de