Extracting Performance Characteristics of Parallel I/O Using Machine Learning

Eugen Betke

University of Hamburg
Department Informatik
Scientific Computing

25.09.2015

Agenda

1. Introduction
2. SIOX-Plugin (Workflow)
3. Experiment
4. Analysis
5. Summary

Table Of Content

1. Introduction
2. SIOX-Plugin (Workflow)
3. Experiment
4. Analysis
5. Summary

SIOX - Scalable I/O for Extreme Performance

- Performance Analysis Framework
- Open-Source-Framework published under LGPL
- Supports MPI-, POSIX-, HDF5- and NETCDF4-Layers
- Modular design
- Online Analysis
  - Analyse activities during program execution
- Offline Analysis
  - Analyse activities after program termination

SIOX-Activity

POSIX-Operations

```c
size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream);
size_t fwrite(const void *ptr, size_t size, size_t nmemb, FILE *stream);
int fseek(FILE *stream, long offset, int whence);
```

Activity-Attributes

Type	Name	Description
`ActivityID`	`aid`	unique identifier
`UniqueComponentActivityID`	`ucaid`	type of the I/O operation
`Timestamp`	`time_start`	start time in nano seconds
`Timestamp`	`time_stop`	stop time in nano seconds
`vector<ActivityID>`	`parentArray`	first I/O operation(s)
`vector<RemoteCall>`	`remoteCallsArray`	-
`vector<Attribute>`	`attributeArray`	parameters, return value, ...
`RemoteCallIdentifier*`	`remoteInvoker`	-
`ActivityError`	`errorValue`	-

Activity-Sequence

$a_{open} a_{write} a_{write} a_{write} a_{open} a_{open} a_{read} a_{close} a_{read} \cdots$

Mapping: Activity → Feature Vector

- Machine learning requires a suitable representation

- A feature vector contains a set of features
- A feature describes a property of an object
- Success depends on the right choice
  - What are the right features?

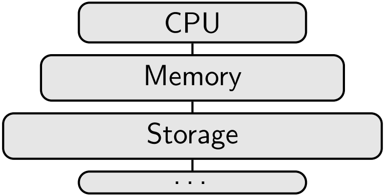
Position	Feature	Description
1	`delta time`	$start\_time_{prev\_act.} - start\_time_{curr\_act.}$
2	`operation type`	UniqueComponentActivityID
3	`file descriptor`	posix file identifier
4	`duration`	activity runtime
5	`size`	amount of data
6	`offset`	$end\_pos_{prev\_act.} - start\_pos_{curr\_act.}$

Goals

Feature Vector

Position	Feature
1	`delta time`
2	`operation type`
3	`file descriptor`
4	`duration`
5	`size`
6	`offset`

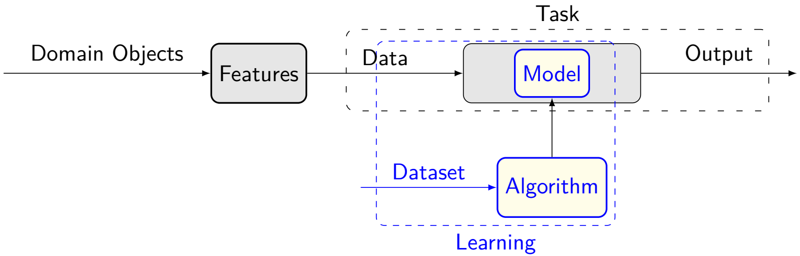
Simplified Cache Hierarchy



Goal: SIOX-Plugin

- Mapping from activity (to feature vector) to cache type
    - Create statistics
- Mapping from activity (to feature vector) to performance value (`duration` · `size`)
    - Create hints for developers
- Automatization of the process

Machine Learning

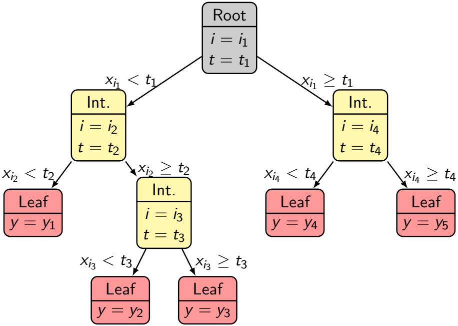


1. Learn from data
2. Predict labels

Binary Decision Trees

Feature vector:

$x := (x_1, x_2, \ldots, x_i, \ldots, x_d)$



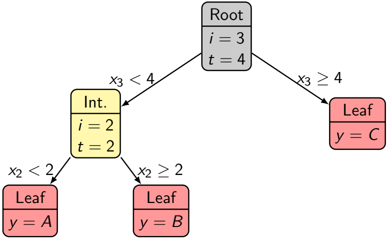
Advantages

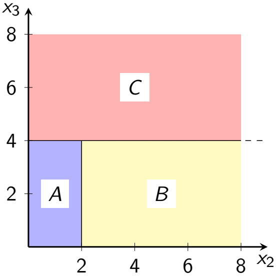
- Simplicity
- Convertible to rules
- Feature filtering

i	attribute index
t	threshold
y	output

Binary Decision Trees - Example

Example: Compute $M(x)$ using feature vector $x = (4, 5, 2)$.





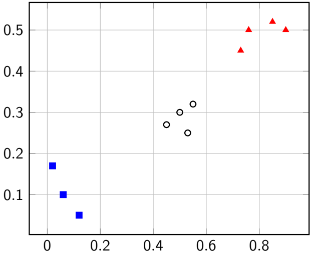
i	attribute index
t	threshold
y	output

$$x_3 < 4 \lor x_2 < 2 = A \quad (1)$$

$$x_3 < 4 \lor x_2 \geq 2 = B \quad (2)$$

$$x_3 \geq 4 = C \quad (3)$$

Clustering-Algorithm



Task

- Group similar vectors

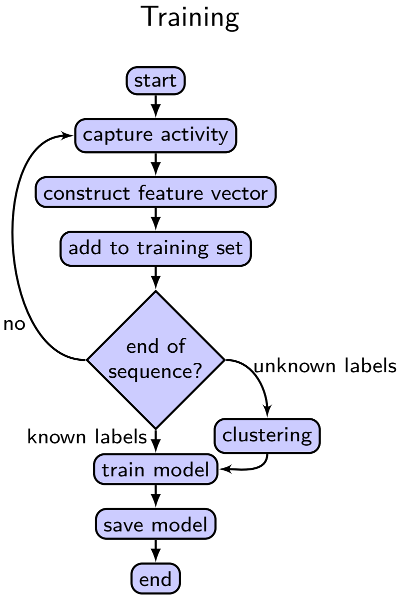
Purpose

- Label feature vectors
- Discover unexpected groups / anomalies

Table Of Content

1. Introduction
2. SIOX-Plugin (Workflow)
3. Experiment
4. Analysis
5. Summary

Workflow


Training
start
capture activity
construct feature vector
add to training set
end of sequence?
no
unknown labels
clustering
known labels
train model
save model
end

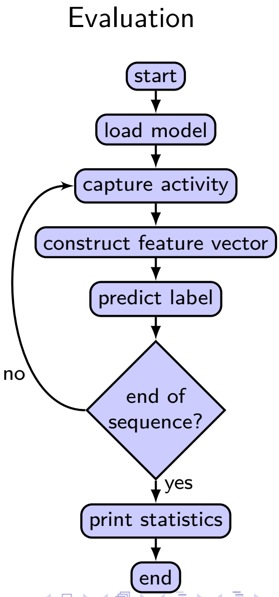

Evaluation
start
load model
capture activity
construct feature vector
predict label
end of sequence?
no
yes
print statistics
end

Table Of Content

1. Introduction
2. SIOX-Plugin (Workflow)
3. Experiment
4. Analysis
5. Summary

Dataset

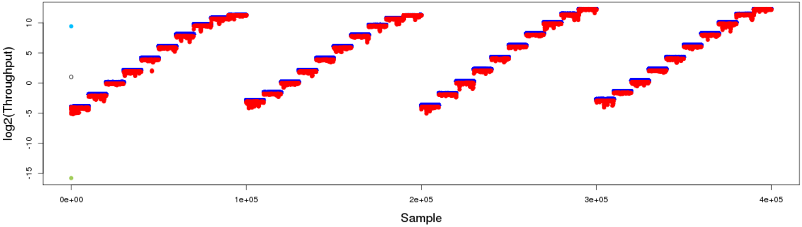
Hardware:

- 10 Compute-Nodes
- 10 I/O-Nodes
  - CPU: Intel Xeon E3-1278@3.4 GHz
  - RAM: 16 GByte
  - HDD: Seagate 7200.12 ($\approx$ 100 MiB/s)
- Nodes are interconnected with a Gigabit Ethernet
- Operation system: CentOS 6.5
- Filesystem: Lustre 2.5.

Experiment:

- Performance is measured for different data sizes
- (optimal) data sizes: 4, 16, 64, . . . , 262144 bytes
- (suboptimal) data sizes: 5, 17, 65, . . . , 262145 bytes

Correct and wrong clustering of the dataset



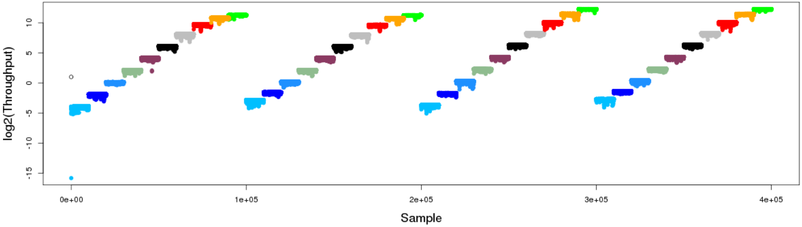
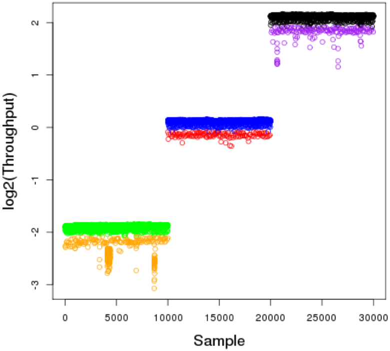


Table Of Content

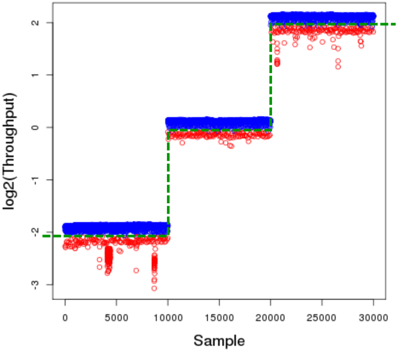
1. Introduction
2. SIOX-Plugin (Workflow)
3. Experiment
4. Analysis
5. Summary

Wrong clustering



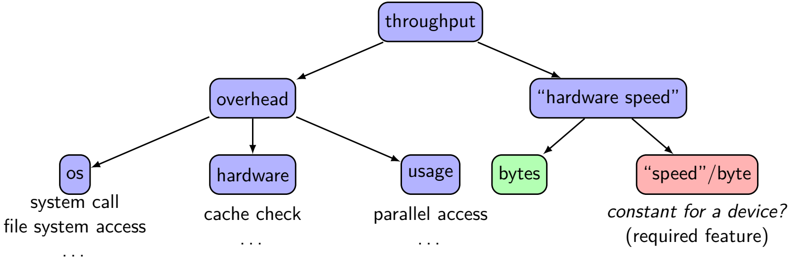
- Suppose cache type can be separated
- How to connect the clusters?

Complicated trees



- Separation through vertical and horizontal lines
- Too many separations

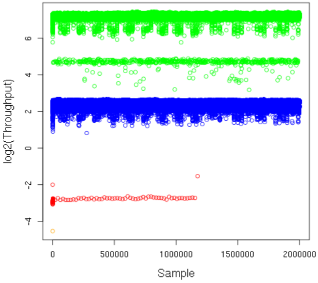
ToDo: Decomposition of I/O Path

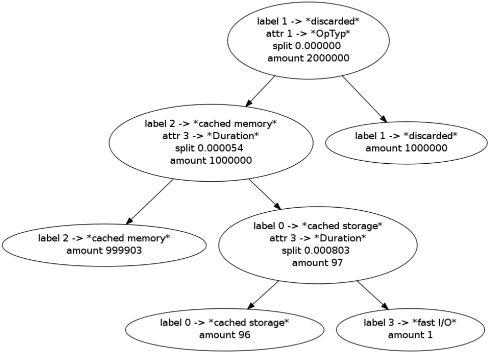


- Speed/byte can be directly mapped to cache component
- Challenging task
  - Throughput is sensitive
  - Hardware dependent
  - Operating system dependent
  - Usage dependent

Expected Result

```
siox-inst posix dd of=/dev/null if=testfile bs=100 count=100000
```





label 1 -> *discarded*
attr 1 -> *OpTyp*
split 0.000000
amount 2000000

label 2 -> *cached memory*
attr 3 -> *Duration*
split 0.000054
amount 1000000

label 1 -> *discarded*
amount 1000000

label 2 -> *cached memory*
amount 999903

label 0 -> *cached storage*
attr 3 -> *Duration*
split 0.000803
amount 97

label 0 -> *cached storage*
amount 96

label 3 -> *fast I/O*
amount 1

- 100*kb* read-operations
- write-operations to null device
- Accuracy ≈ 99.7%

Table Of Content

1. Introduction
2. SIOX-Plugin (Workflow)
3. Experiment
4. Analysis
5. Summary

Summary

- Introduction
  - SIOX, Activities, Activity-Traces
  - Binary Decision tress
  - Clustering algorithm
- Workflow
  - 2-Phase-Procedure
- Problems
  - Choosen algorithm not suitable for data
- Solutions
  - Split throughput
  - Cluster dataset and assign labels

End

Questions?