OAID / **CaffeOnACL**

Branch: **master** ▾ **CaffeOnACL** / acl_openailab / **Userguide.md** Find file Copy path

xhbdahai update perfomermance summary pic 8de24c5 on 11 Jul

1 contributor

```
131 lines (95 sloc)   7.09 KB
```

1. Purpose

This document will help user utilize the code of CaffeOnACL(Caffe+ACL) to improve the performance of their applications based on the Caffe framework.

2. Setup applicaiton developmnet evironment

2.1 ACL and Caffe Libraris build

Please refer to build caffeOnACL.

2.2 ACL and Caffe Libraris installation

There are two ways to use the libraris after build them.

Install libraries to standard direcotry '/usr/local/lib'

```
sudo cp ~/oaid/ComputeLibrary/build/arm_compute/libarm_compute.so /usr/local/lib/.
sudo cp ~/oaid/caffeOnACL/distribute/lib/libcaffe.so /usr/local/lib/.
```

Or, install libraries to a temporarily directory. then set the environment variable LD_LIBRARY_PATH is this direcotry. For example, install libraries to `~oaid`

```
mkdir ~/oaid/lib
cp ~/oaid/ComputeLibrary/build/libarm_compute.so ~oaid/lib/.
cp ~/oaid//caffeOnACL/distribute/lib/libcaffe.so ~/oaid/lib/.
export LD_LIBRARY_PATH=~/oaid/lib
```

2.3 How to write Makefile for applications

First, please make sure that environmnet variable "ACL_ROOT" and "CAFFE_ROOT" is set properly (them are set during the process of ACL and Caffe libraries buiding). Double check them by command just like "echo $VAR".
In the Makefile, it needs include the following lines :

```
include $(CAFFE_ROOT)/Makefile.config
CAFFE_INCS = -I$(CAFFE_ROOT)/include -I$(CAFFE_ROOT)/distribute/include/
CAFFE_LIBS = -L$(CAFFE_ROOT)/distribute/lib -lcaffe  -lglog -lgflags -lprotobuf -lboost_system
-lboost_filesystem
```

3. Application configuration guide

3.1 Configuration options on compiling time

Modify the vaule of Make Variables in `$(CAFFE_ROOT)/Makefile.config` to USE or *NOT* USE some function by Caffe.

- "USE_ACL := 1" (Enable ACL support on ARM Platform), "USE_ACL :=0" (Disable ACL support on ARM Platform)
- "USE_PROFILING := 1" (Enable profiling), "USE_PROFILING := 0" (Disable profiling)
- Experimental functions:

  When USE_PROFILING is true, enable "Layer's performance statistic" which controlled by Marco "LAYER_PERF_STAT", is defined by "-DLAYER_PERF_STAT" in "$(CAFFE_ROOT)/Makefile", can remove it to disable the feature. Add "-DUSE_CONV_CACHE" to "COMMON_FLAGS" into "$(CAFFE_ROOT)/Makefile" to enable the cache of convolution layer

3.2 Configure the bypass of ACL Layer

Can set environment "BYPASSACL" to bypass ACL layers, the control bit definitions are listed in the table below:

```
BYPASS_ACL_ABSVAL       0x00000001
BYPASS_ACL_BNLL         0x00000002
BYPASS_ACL_CONV         0x00000004
BYPASS_ACL_FC           0x00000008
BYPASS_ACL_LRN          0x00000010
BYPASS_ACL_POOLING      0x00000020
BYPASS_ACL_RELU         0x00000040
BYPASS_ACL_SIGMOID      0x00000080
BYPASS_ACL_SOFTMAX      0x00000100
BYPASS_ACL_TANH         0x00000200
```

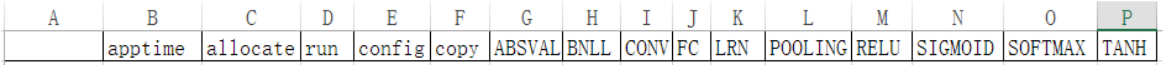
For instance, type "export BYPASSACL=0x100" to bypass ACL Softmax layer; and "export BYPASSACL=0x124" to bypass ACL Softmax, Pooling and Convolution layers.

3.3 Configure the log information

can set "LOGACL" to log the performance information of ACL and related caffe layers, the control bit definitions are listed in the table below:

```
ENABLE_LOG_APP_TIME     0x00000001
ENABLE_LOG_ALLOCATE     0x00000002
ENABLE_LOG_RUN          0x00000004
ENABLE_LOG_CONFIG       0x00000008
ENABLE_LOG_COPY         0x00000010
ENABLE_LOG_ABSVAL       0x00000020
ENABLE_LOG_BNLL         0x00000040
ENABLE_LOG_CONV         0x00000080
ENABLE_LOG_FC           0x00000100
ENABLE_LOG_LRN          0x00000200
ENABLE_LOG_POOLING      0x00000400
ENABLE_LOG_RELU         0x00000800
ENABLE_LOG_SIGMOID      0x00001000
ENABLE_LOG_SOFTMAX      0x00002000
ENABLE_LOG_TANH         0x00004000
```

For instance, type "export LOGACL=0x100" to output the performance information of FC layer; "export BYPASSACL=0x380" to output the performance information of LRN, FC and Convolution layers. You can copy the logs into Microsoft excel, the sum the time information with separated terms, the column of excel sheet like this :



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
	apptime	allocate	run	config	copy	ABSVAL	BNLL	CONV	FC	LRN	POOLING	RELU	SIGMOID	SOFTMAX	TANH

4. Test and Performance Tuning Guide

4.1 To run the application with ACL and log performance information

Assume your working directory is : ~\test

- Use all ACL layers by set BYPASSACL to 0

  export BYPASSACL=0

- If compile the caffeOnACL with "USE_PROFILING := 1", to decide which information is logged into file by setting LOGACL. For instance, we log all layers' information by setting LOGACL to 0x7fe1.

  export LOGACL=0x7fe1

- To check if "configure" take lots of time, can set LOGACL to 0x08.

  export LOGACL=0x08

- To check if "memory copy" take lots of time, we can set LOGACL to 0x10.

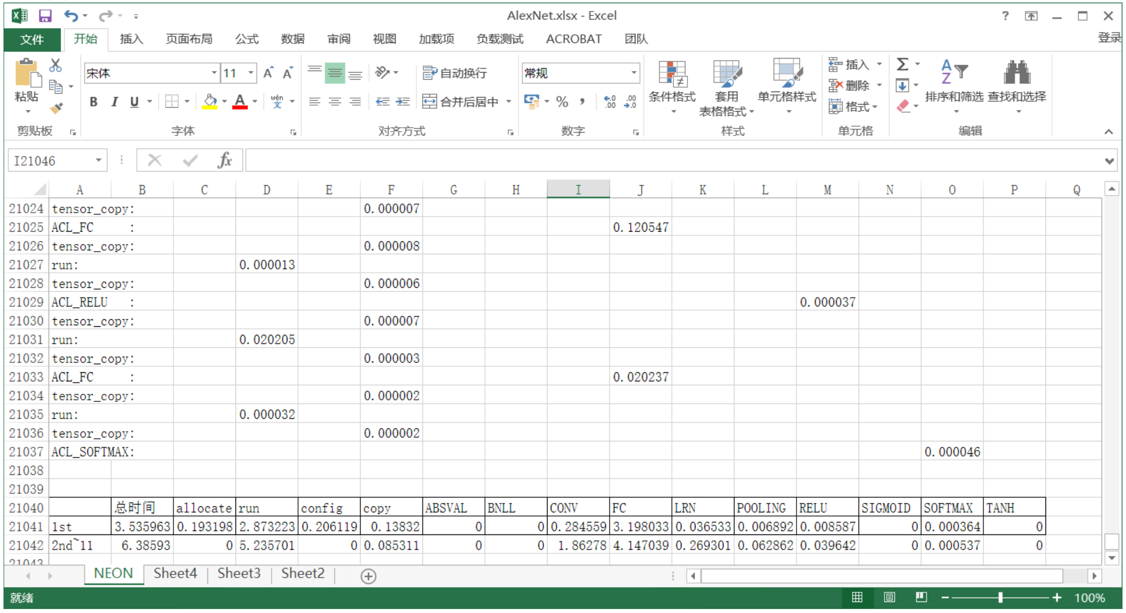
  export LOGACL=0x10

- Run your application and get the information of performance

  ./your_application parameters…

- When got the log, copy it into Microsoft excel, and sum the columns. For examle, run the AlexNet as the example – command line is :

taskset -a 10 ./distribute/bin/classification.bin ./models/bvlc_alexnet/deploy.prototxt ./models/bvlc_alexnet /bvlc_alexnet.caffemodel data/ilsvrc12/imagenet_mean.binaryproto data/ilsvrc12/synset_words.txt examples/images /cat.jpg



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
21024	tensor_copy:					0.000007											
21025	ACL_FC :									0.120547							
21026	tensor_copy:					0.000008											
21027	run:			0.000013													
21028	tensor_copy:					0.000006											
21029	ACL_RELU :												0.000037				
21030	tensor_copy:					0.000007											
21031	run:			0.020205													
21032	tensor_copy:					0.000003											
21033	ACL_FC :									0.020237							
21034	tensor_copy:					0.000002											
21035	run:			0.000032													
21036	tensor_copy:					0.000002											
21037	ACL_SOFTMAX:														0.000046		
21038																	
21039																	
21040		总时间	allocate	run	config	copy	ABSVAL	BNLL	CONV	FC	LRN	POOLING	RELU	SIGMOID	SOFTMAX	TANH	
21041	1st	3.535963	0.193198	2.873223	0.206119	0.13832	0	0	0.284559	3.198033	0.036533	0.006892	0.008587	0	0.000364	0	
21042	2nd~11	6.38593	0	5.235701	0	0.085311	0	0	1.86278	4.147039	0.269301	0.062862	0.039642	0	0.000537	0	

4.2 To run the application with original Caffe's layers and log performance information

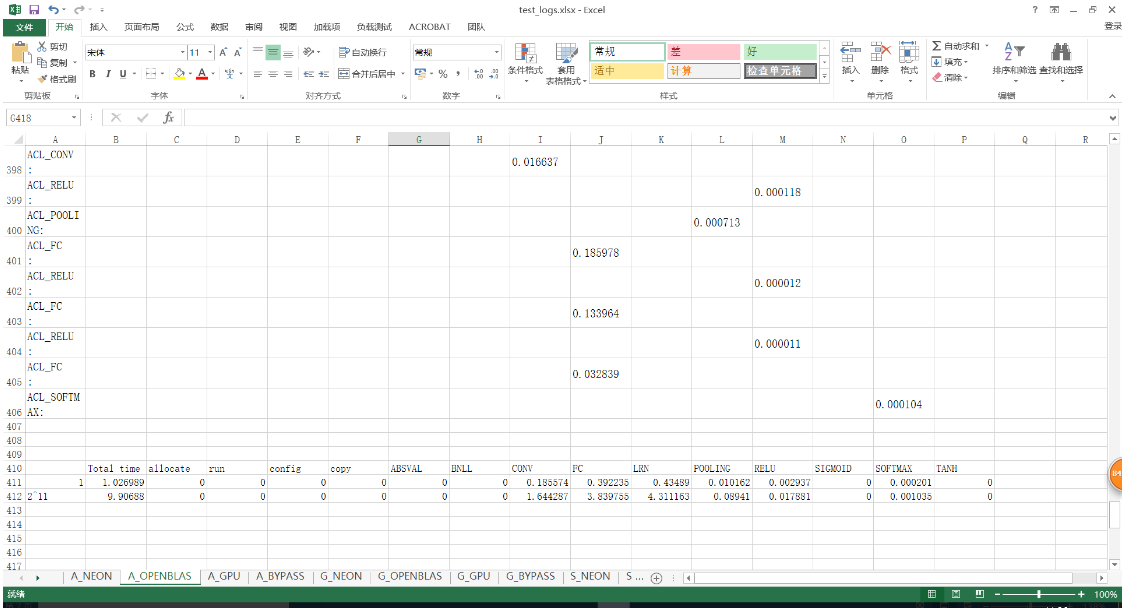
Assume your work directory is ~\test.

- to use all ACL layers by set BYPASSACL to 0xffffffff export BYPASSACL=0xffffffff
- If compile the caffeOnACL with "USE_PROFILING := 1", to decide which information is logged into file by setting LOGACL. For instance, we log all layers' information by setting LOGACL to 0x7fe1. (In this case, ENABLE_LOG_ALLOCATE、ENABLE_LOG_RUN、ENABLE_LOG_CONFIG and ENABLE_LOG_COPY are invalidate, these flags are all for ACL layers)

  export LOGACL=0x7fe1
- Run your application and get the information of performance

  ./your_application parameters…
-

When got the log, we can copy it into Microsoft excel, and sum the columns



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
398	ACL_CONV :								0.016637									
399	ACL_RELU :												0.000118					
400	ACL_POOLING:											0.000713						
401	ACL_FC :									0.185978								
402	ACL_RELU :												0.000012					
403	ACL_FC :									0.133964								
404	ACL_RELU :												0.000011					
405	ACL_FC :									0.032839								
406	ACL_SOFTMAX:														0.000104			
407																		
408																		
409																		
410		Total time	allocate	run	config	copy	ABSVAL	BNLL	CONV	FC	LRN	POOLING	RELU	SIGMOID	SOFTMAX	TANH		
411	1	1.026989	0	0	0	0	0	0	0.185574	0.392235	0.43489	0.010162	0.002937	0	0.000201	0		
412	2^11	9.90688	0	0	0	0	0	0	1.644287	3.839755	4.311163	0.08941	0.017881	0	0.001035	0		
413																		
414																		
415																		
416																		
417																		

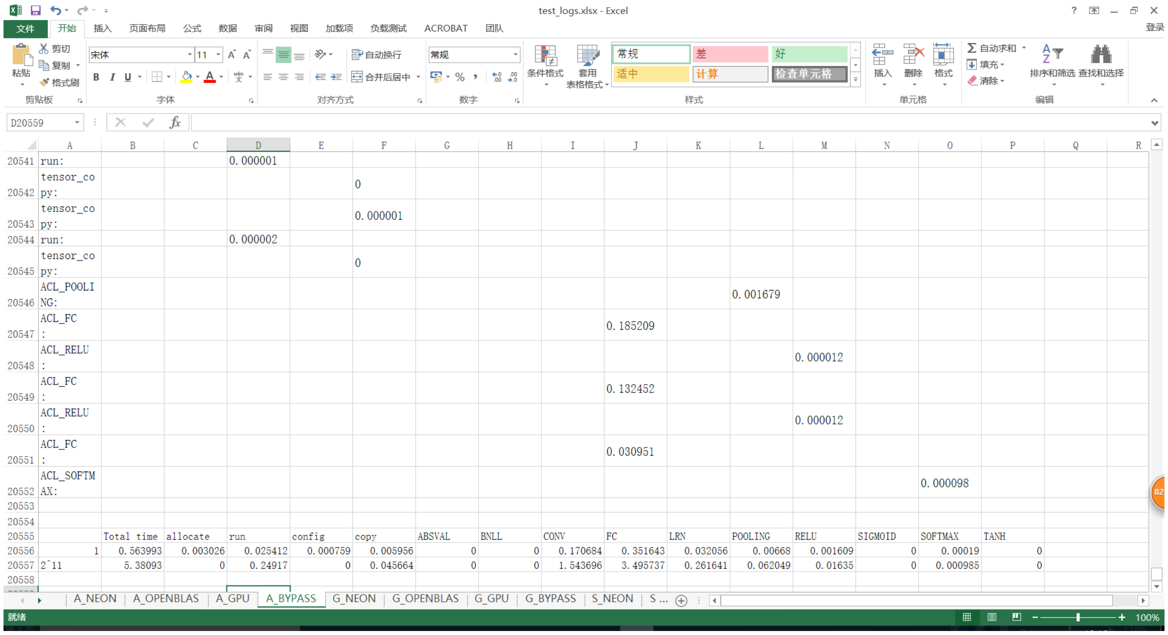
4.3 Improve the performance by mixing ACL Layers and Caffe’s original Layers

After retrieving the performance statistic data of Caffe's layers and ACL's layers in your application, we can compare their respective performances:

	Total time	CONV	FC	LRN	Pooling	RELU	SOFTMAX
ACL_NEON	3.535963	0.284559	3.198033	0.036533	0.006892	0.008587	0.000364
Caffe_Org (OpenBLAS)	1.026989	0.185574	0.392235	0.43489	0.010162	0.002937	0.000201

From the table above, we can observe that in the original caffe’s layer, CONV、FC、RELU and Softmax have faster running times than ACL’s layers. Therefore, we can set BYPASSACL to 0x14c to BYPASS the 4 ACL layers, and utilize the original caffe’s layers in the application. By choosing the layerset with the faster running time for each layer, we can optimize the total running time for this application

The performance data is :



D20559

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
20541	run:			0.000001														
20542	tensor_copy:					0												
20543	tensor_copy:					0.000001												
20544	run:			0.000002														
20545	tensor_copy:					0												
20546	ACL_POOLING:											0.001679						
20547	ACL_FC :									0.185209								
20548	ACL_RELU :												0.000012					
20549	ACL_FC :									0.132452								
20550	ACL_RELU :												0.000012					
20551	ACL_FC :									0.030951								
20552	ACL_SOFTMAX:														0.000098			
20553																		
20554																		
20555		Total time	allocate	run	config	copy	ABSVAL	BNLL	CONV	FC	LRN	POOLING	RELU	SIGMOID	SOFTMAX	TANH		
20556	1	0.563993	0.003026	0.025412	0.000759	0.005956	0	0	0.170684	0.351643	0.032056	0.00668	0.001609	0	0.00019	0		
20557	2^11	5.38093	0	0.24917	0	0.045664	0	0	1.543696	3.495737	0.261641	0.062049	0.01635	0	0.000985	0		
20558																		

As you can see, we obtain optimal performance in combined mode (ACL: LRN、Pooling, Caffe’s original Layers：Conv、FC、RELU、Softmax) as in the table below:

	Total time	CONV	FC	LRN	Pooling	RELU	SOFTMAX
BYPASS (CONV、FC、RELUand Softmax)	0.563993	0.170684	0.351643	0.032056	0.00668	0.001609	0.00019