Report of ASC Student Supercomputer Challenge 2017 Preliminary Contest

Zhibin Xie, Zhen'ning Wang, Jiafang Xue, Guohua Tian
School of Information Engineering,
NanChang University

Xianling Dong
School of Science,
NanChang University

1. Super Computing At Nanchang University

Nanchang University (NCU), recognized as one of the key university in Project 211 in the mid-western region of China, has shifted the focus on the education of cutting-edge technology and frontier research. In college of information science and technology, we provide undergraduate programs that allow student to know, touch, and learn about high performance computing and Big Data analysis, with the focus on the computation ablity. We have recently investigate multiple local and nation-wide sponsored projects on supporting advancing super computing in both research and education. In research, our university builds small scale computing clusters (i.e., over 20 nodes) to simulate near-space simulation and emulation in both 2D and 3D layout. The scale of the computing platform has been scaled up last year after our college of information and technology further investigate over 3million Yuan for cluster upgrade using the Inspur scientific computing solutions. To further exploit the capacity of super computing, we also offload large amounts of computation jobs onto the TianHe II platform. In this way, we build a distributed scientific computing hierarchy as local algorithm test/data verification, and remote hardcore high performance computing.

In the aspect of education, we also provide free charge platform to support students who are interested in super computing and data engineering. We have investigated 1.7 million to build a micro datacenter with 30 computation nodes, 8 storage nodes with network-attached storage (NAS), and rest manage nodes. The purposes of building such platform are three folds: (1) Provide a sandbox of raw Linux system/HPC hardware/machine learning frameworks for student to play with; (2) Build a syllabus around high performance computing in order to train engineers, and possibly researchers in this field; (3) Support education-related research on high performance computing. In three months, we have over 50 undergraduate and 15 graduate students working on this platform to study, manage, do research, and work on their thesis. This is a relatively new trend on the revolutionize the out-of-date courses, and now combined into the most recent technique and system designs.

1.1. On-going Grant-support Projects

The super computing platform in NCU is currently supporting multiple on-going research projects. Research sponsored by National Science Foundation China (NSFC) and Jiangxi Provincial department of science and technology can be separated into three major categories. The first category is the scientific computing research on space science and technology, with a focus on near-space simulation, and Terahertz radiation application. The second group of on-going research are image processing and computer vision projects on Big data, such as medical images and records. Our professors use deep learning methods to explore and discover symptoms from CT/MRI images to help symptom diagnose and remote surgery. The last group focuses on the computation capacity. The research is mainly focus on how to build an efficient yet effective computing architecture to support the aforementioned two groups of research by draining every possible drop of computation frequency and I/O bandwidth, such that the latency is minimized while the correctness and stability of the system are preserved.

2. Introduction of NSBSS

We are a group of youths loving computer science who have a huge dream, and working hard for reaching higher. We gathered together due to strong interest in supercomputer.
It is very happy to us that we have get together to take challenge of ASC17, and it is a good chance to get skills and knowledge about supercomputer.

3. HPL Test

3.0.1. System environment.

Name	Configuration
Inspur NF6248	CPUIntel Xeon Phi 7210, 1.3Ghz, 64 cores Memory: 16G x4, DDR4, 2133Mhz Hard disk: 120G SSD x 1

Classification	Description	Version
OS	Linux	CentOS7
Compiler	Intel Composer	2017.1.043
MKL	Intel MKL	2017.1.043
MPI	Intel MPI	2017 UP1
PBS	Torque	

3.0.2. HPL install. In this test and optimize, we chose Intel Optimized LINPACK.

3.0.3. Test Step.

3.0.4. PxQ. Because Intel Optimized LINPACK use all CPU core in multi-thread by default, so we set P x Q to 1 x 1.

3.0.5. BLAS. There are many blas, such as openblas, mkl, gogoblas and atlas. Because we are using intel xeon phi, MKL will be the best choice

3.0.6. N. Experience tells us, N can be calculated by

$$N^2 * 8 = MEM * 80\%$$

so N should have been 82900, but in our test, when N is set to
35000, GFlops reach to 1.7GFlops, which is much higher when N is 829000

N	33000	34000	35000	36000	37000
GFlops	1702	1701.48	1714.1	1708.11	1501.21
Time (S)	14.08	15.4	16.68	18.21	22.5

3.0.7. NB.

NB	232	256	296
GFlops	1709.86	1714.1	1702.39

3.1. Best Score

Finally, we reached at 1.7Gflops.

```
--------------------------------------------------------------------------------

- The matrix A is randomly generated for each test.
- The following scaled residual check will be computed:
      ||Ax-b||_oo / ( eps * ( || x ||_oo * || A ||_oo + || b ||_oo ) * N )
- The relative machine precision (eps) is taken to be            1.110223e-16
- Computational tests pass if scaled residuals are less than                16.0

================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WC23C2C4       35000   256     1     1             16.66           1.71587e+03
HPL_pdgesv() start time Mon Mar  6 14:01:38 2017

HPL_pdgesv() end time   Mon Mar  6 14:01:54 2017

--------------------------------------------------------------------------------
```

3.2. Problems

Problem one, when N is set to theoretical size, the GFlops goes down straight.

When we used multi-node, the performance become lower than we run the test on single-node, its confusing to us very much.

4. MASNUM_WAVE

4.1. Ask question

Question one,we use the queue of our team running exp1 and exp2. Then we get two files and verify them with standard results, but the makefile of verifica can't run in the folder of masnum_wave. we compare pac_ncep_wav_20090228.nc to pac_ncep_wav_20090228_stardard.nc and compare global_ncep_wav_20090630.nc to globalncep_wav_20090630_stardard.nc. Them compare success. But when we start to compare them, we begin to make the makefile, then run executable file compare_exp1 getting a result "-bash: ./compare_exp1: connot execute binary file". Question two, we use qsub to submit job and need to find the best flags for it. Question three, when we use mpi + OpenACC* to optimize source code, and we test source code finding that the function of mpi_reduce() and mpi_bcast() which locate 251 line and 252 line in the file of source/ympi_mod/ympi_mod.f90 use the most time, mainly to input data to cache. However, when we use OpenAcc* to optimize the /source/ympi_mod/wammpi_mod.f90, we get errors. You can see following pictures in appendix A.

4.2. Analyze problems

Firstly, Ocean wave model, we need to solve the optimization problem is how to ensure the accuracy of the case, the forecast to spend the shortest time. The main problems are code optimization, mathematical optimization, compiler selection, and the use of qsub to submit tasks, the need for parameter selection and other issues, code optimization is to Fortran language loop optimization, function call optimization, mathematical optimization is mainly marine wave prediction Formula optimization. The choice of the compiler is mainly to optimize the parallelization.

4.3. Solve problems

quetion one, We find the reason , because the compiler make us get fault. After we use gfortran compiler to make makefile and run success. And question two, we are step by step to test the best flags. Our qsub submit command which is "bsub -n 64 -np 4 -p -q q_sw_asc_2 -share_size 7500 -host_stack 1024 -b -m 1 -o out.qrunout -cgsp 64 ./masnum.wam.mpi", We use Mpi + OpenACC* to optimize our source code. You can see following pictures in appendix A.

4.4. Get conclusion

Now we optimize exp1 get result which is five hours, about five minutes to forecast a day, finishing job spend five hours. But we can't slove error of using OpenACC* optimization. can see last one picture following in appendix A.

5. Deep learning contest

In-depth learning is a learning algorithm that simulates the cognitive pattern of the human brain, and the current Baidu Research Institute uses the strategic direction of data mining and artificial intelligence as an important development strategy of the 21st century.

PaddlePaddle is originated in Baidu's open source depth learning platform.And it's easy to use.

5.1. Analysis of the problem

Data are sampled from mobile phone APP.Using the dataset which are aggregated about 50 days from 00:00 a.m. on March 1st to 08:00 on April 20th for training. Your task is to predict each link's speed for every 5 minutes from 08:00 to 10:00 on April 20th.

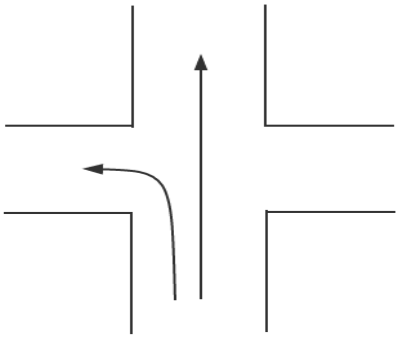
The model outputs are evaluated by the Root Mean Squared Error (RMSE) of predictions.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} \left({X_{actual,i}}^{1} - {X_{model,i}}^{2} \right)^2}{n}}$$

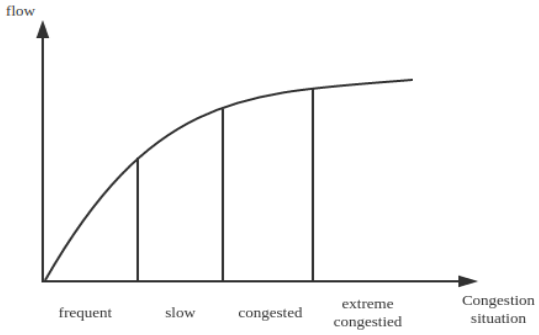
[1] $X_{actual,i}$ is actual value of moment i,

[2] $X_{model,i}$,i is your prediction moment of i

By simply observing the node graphs, we map the traffic nodes, for example, simple crossroads



Corresponding to the four traffic states.



The result of clustering is the four eigenvectors (1, 2, 3, 4) corresponding to the state defined by the topic. The four eigenvectors represent frequently, slow, congested and extreme congested.

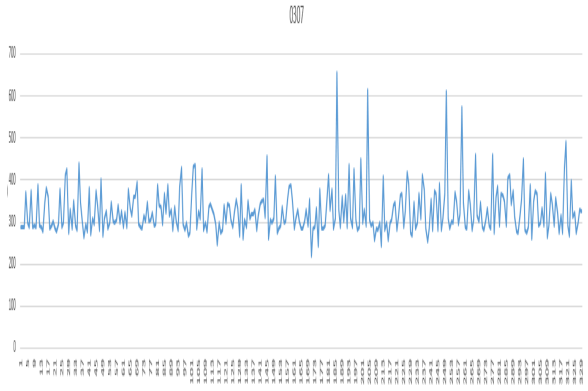
5.2. Questions

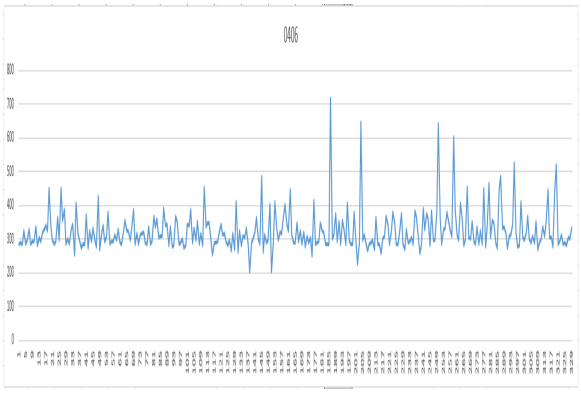
After we downloaded the data, we found that the amount of data is large, some of the data there is a large interference, so the data processing is very important.We also have little understanding of the depth of learning. so it is difficult to find theoretical support. And the results of the prediction, and the calculation is unclear.

5.3. Preliminary solution

5.3.1. Data processing. Through the analysis of the title and data, we selected the daily time from 00:00-02:00, 04:00-06:00, 08:00-10:00, 12:00-14:00, 16:00-18:00 and 20:00-22:00 six time periods as the representative.In order to reduce the error, we take the average of the time to process the data.

For examples, March 7th and April 6th. These images have similar places





5.3.2. Algorithm selection. Algorithm we used RMSProp, RMSProp is a very efficient algorithm, RMSProp slightly improved AdaGrad, making the algorithm no longer as radical as AdaGrad it is moving average of squared gradients.

The equations of this method as follows:

$$\nu\left(w,t\right)=\rho\nu\left(w,t-1\right)+\left(1-\rho\right)\left(\nabla Q_i(w)\right)$$

$$w=w-\frac{\eta}{\sqrt{\nu(w,t)+\varepsilon}}\left(\nabla Q_i(w)\right)$$

5.3.3. Result analysis. We perform the error analysis by plotting the original data, and comparing the resulting data

Relative error:

$$rerr=\frac{X_{actual,i}-X_{model,i}}{X_{actual,i}}$$

Absolute relative error:

$$mrerr=\frac{1}{N}\sum_{1}^{N}\left(\frac{X_{actual,i}-X_{model,i}}{X_{actual,i}}\right)$$

5.4. Algorithm improvement

Adam is a recently proposed algorithm that is similar to the RMSprop comparison with the momentum. The procedure is similar to:

$$m(w,t)=\beta_1 m(w,t-1)+(1-\beta_1)\nabla Q_i(w)$$
$$v(w,t)=\beta_2 v(w,t-1)+(1-\beta_2)(\nabla Q_i(w))^2$$
$$w=w-\frac{\eta}{\sqrt{\nu(w,t)+\varepsilon}}$$

The only difference between this method and RMSProp is the ”smooth” process, which uses m to do the smooth operation instead of using the original gradient vector dx.And we set eps=1e-6, bata1=0.9, beta2=0.999.

5.5. Parameter configuration

Choose a suitable learning rate is very difficult, in the beginning, we set the way through the schedule learning learning method, that is pre-designed a certain series of iterations to increase or increase learning rate.As the learning rate increases, we find that the relative error begins to increase.Finally we set the learning rate to 1e-6.

5.6. Cluster training

As little understanding of the GPU, we only refer to the PaddlePaddle document cluster training related to the introduction and methods.Due to the lack of knowledge at the hardware level, we encountered many problems when we built the cluster.

5.7. Result

TABLE 1. Ł

	$learngin\_rate$	$rerr_i$	$mrerr_i$
1	1e-1		
2	1e-2		
3	1e-3		
4	1e-4		
5	1e-5		
6	1e-6		
7	1e-7		
8	1e-8		
	Total Sqerr		

5.8. Summary

Due to the lack of relevant knowledge, our theoretical support is insufficient, the accuracy of the results can not be estimated, the actual accuracy is not very accurate.

6. Conclusion

No conclusion yet...

7. Appendix

7.1. A.

```
[ascusr191@ascusr191 exp1]$ ./compare_exp1
compare HS between ../../exp/exp1/pac_ncep_wav_20090228.nc and ./pac_ncep_wav_20090228_stardard.nc
Step 1: Open file ../../exp/exp1/pac_ncep_wav_20090228.nc
Step 1 Success
Step 2: Open file ./pac_ncep_wav_20090228_stardard.nc
Step 2 Success
Step 3
Step 3.1 Compare missing value
Step 3.1 Success
Step 3.2 Compare scale_factor
Step 3.2 Success
Step 3.3 Compare HS
```

```
#F77 = mpicc
F77 = swafort
LF77OPTS =
FFLAGS   = -O3

EXE = masnum.wam.mpi
```

```
[ascusr191@ascusr191 bin]$ make
swafort -I/usr/sw-mpp/apps/lib/netcdf/include -priv:ignore_call -ldmAnalyse -Minfo -O3 -c ../wave_cor/netcdf_mod.f90
swafort -I/usr/sw-mpp/apps/lib/netcdf/include -priv:ignore_call -ldmAnalyse -Minfo -O3 -c ../wave_cor/time_mod.f90
swafort -I/usr/sw-mpp/apps/lib/netcdf/include -priv:ignore_call -ldmAnalyse -Minfo -O3 -c ../wave_cor/wamvar_mod.f90
"wamvar_mod_host.f90", line 263: Warning: Multiple DATA initialization of storage for .data_init.in.wamvar_mod_. Some
swafort -I/usr/sw-mpp/apps/lib/netcdf/include -priv:ignore_call -ldmAnalyse -Minfo -O3 -c ../wave_cor/wamcpl_mod.f90
swafort -I/usr/sw-mpp/apps/lib/netcdf/include -priv:ignore_call -ldmAnalyse -Minfo -O3 -c ../wave_cor/wamfio_mod.f90
swafort -I/usr/sw-mpp/apps/lib/netcdf/include -priv:ignore_call -ldmAnalyse -Minfo -O3 -c ../wave_cor/wamcor_mod.f90
swafort -I/usr/sw-mpp/apps/lib/netcdf/include -priv:ignore_call -ldmAnalyse -Minfo -O3 -c ../vmpi_mod/vmpi_mod.f90
swafort -I/usr/sw-mpp/apps/lib/netcdf/include -priv:ignore_call -ldmAnalyse -Minfo -O3 -c ../ympi_mod/wammpi_mod.f90
Got AstNode: SgConcatenationOp, do not handled yet. Finished the analysis.
warnning: Can't deal with complex access, aynalysis is terminated[1408], code location:</home/export/online1/ascusr19
Got AstNode: SgConcatenationOp, do not handled yet. Finished the analysis.
Warning: Got one unknown function call of datevec, please check the result manually!
Warning: Got one unknown function call of datestr, please check the result manually!
Warning: Got one unknown function call of get_wind, please check the result manually!
Warning: Got one unknown function call of setspec, please check the result manually!
Warning: Got one unknown function call of set_uv, please check the result manually!
Warning: Got one unknown function call of set_ice, please check the result manually!
Warning: Got one unknown function call of mpi_get_timestep, please check the result manually!
Warning: Got one unknown function call of propagat, please check the result manually!
Warning: Got one unknown function call of implsch, please check the result manually!
Warning: Got one unknown function call of updatev, please check the result manually!
Warning: Got one unknown function call of mean1, please check the result manually!
Got AstNode: SgConcatenationOp, do not handled yet. Finished the analysis.
Warning: Can't deal with complex access, aynalysis is terminated[1400], code location:</home/export/online1/ascusr19
Warning: Got one unknown function call of outrst, please check the result manually!
Got AstNode: SgConcatenationOp, do not handled yet. Finished the analysis.
Warning: Got one unknown function call of output, please check the result manually!
wammpi_mod.f90:72: Warning: copyout in 'parallel' will be ignored.
```

```
key = 0
dtime = dtime0
!$ACC PARALLEL LOOP COPYIN(dtime,key,myid,halosize,ixs,ixl,ix2,iys,iyl,kl,jl,
e,ee);dimension(e(kl,jl,ix1,iyl),ee(kl,jl,ix1,iyl))) local(ctime,it,is,ic,j,k,
do it = 0, itend

-------------------------------------------------------------------------------

--- Set time & key for cool start.

  dtime = dtime + key * delttm / 1440.
  if(it >= number)key = 1
   dtime = dtime0 + key * (it - number) * delttm / 1440.
  itime = datevec(dtime)
  ctime = datestr(itime)
```

```
subroutine get_mpiminr(x)
  real(4), intent(inout) :: x
  real(4) :: y
  integer :: ierr
  y = x
  write(6, *) 'test getmpiminr()'
  call mpi_reduce(y, x, 1, mpi_real, mpi_min, 0, mpi_comm_ympi, ierr)
  write(6, *) 'mpi_reduce2 use time is '
  call mpi_bcast(x, 1, mpi_real, 0, mpi_comm_ympi, ierr)
  !write(6, *) 'mpi_bcast2 use time is '
end subroutine get_mpiminr

!-------------------------------------------------------------------------------
```

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

[2] XunqiangYin. A highly effective global surface wave numerical simulation with ultra-high resolution

[3] Yongzeng Yang and Xunqiang Yin. User Guide for MASNUM-WAM 2.2(2010-07-10)

[4] J. Kraus. NVIDIA, Multi GPU Programming with MPI and OpenACC(2016-10-26)

[5] National Supercomputing Center in Wuxi. Sunway TaihuLight Quick Start Guide

[6] National Supercomputing Center in Wuxi. Sunway TaihuLight OpenACC* Quick User Handset

[7] National Supercomputing Center in Wuxi. Sunway TaihuLight Compiler System User Handset