

# 2019 APAC HPC-AI Competition

## Team BowhiVolar

Chun-chieh, Chuang  
Hsien-ting, Hsieh  
Chun-wei, Yang

Cheng-Yu, Huang  
Zhong-wei, Hsu  
Yen-fu, Liu

Coaches:

Prof. Chi-Chuan, Hwang

Chao-Chin, Li

Part I –  
AI task  
Tensorflow

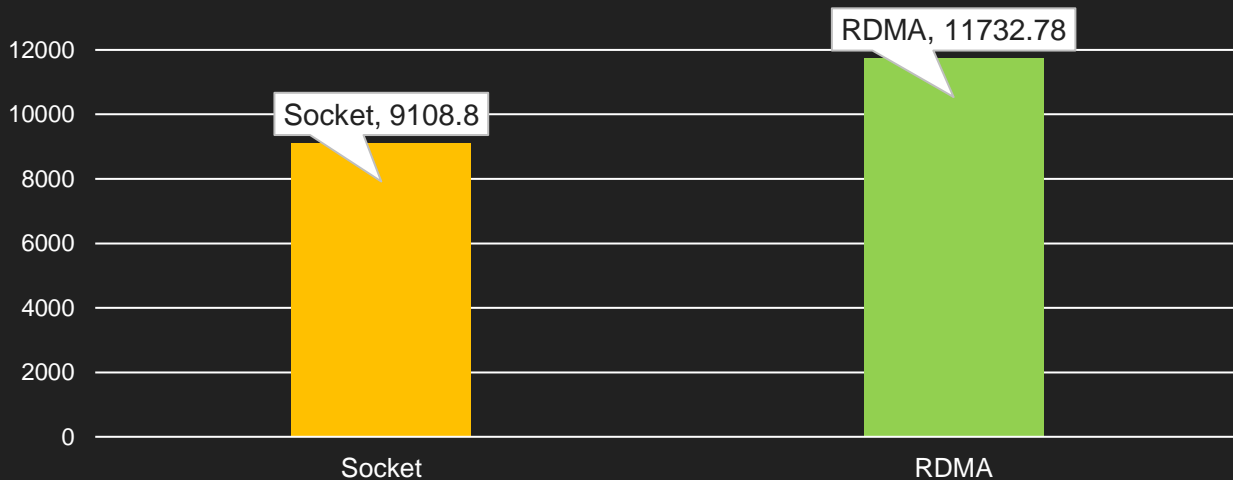


# Experiment – Socket v.s. RDMA

## NSCC

Environment	CPU	2 x Xeon 2698 V4
	GPU	2 x DGX1 (16xV100)
	OS	Ubuntu 16.04 LTS
	CUDA / driver	9.0.176 / 384.81
	cuDNN	7.6
	NCCL	2.4.7
	Infiniband	EDR
	Tensorflow	1.13
	Benchmark github hash	cnn_tf_v1.13
	Horovod	0.16.2
	openMPI	3.1.3
	sync async	Sync
	Local Parameter Device	CPU
Hyperparameter	Dataset	ILRSV2012
	Model	Resnet_v1.5
	Optimizer	Momentum
	precision	fp16

	Socket	RDMA
Batch size	256	240
image / s	9108.8	11732.78





# Optimizer

## Our cluster

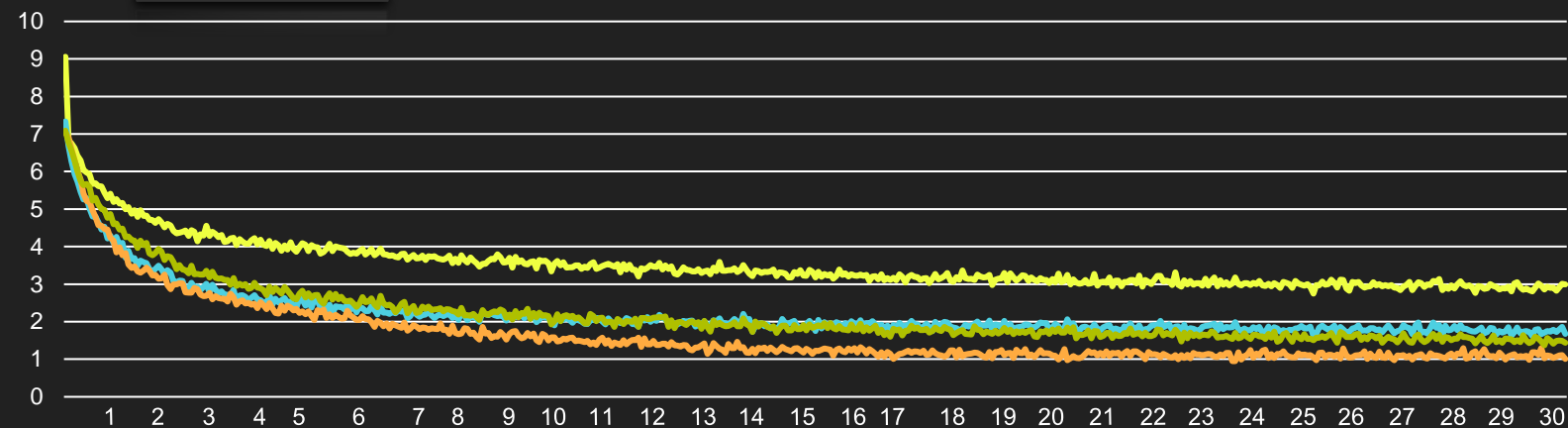
Environment	CPU	Core i9-9900X
	GPU	Nvidia 2080Ti * 2
	OS	Ubuntu 16.04 LTS
	CUDA / driver	10.0.130 / 430.14
	cuDNN	7.6.0.64-1
	NCCL	2.4.7-1
	Infiniband	EDR
	Tensorflow	1.13
	Benchmark github hash	cnn_tf_v1.13
	Horovod	0.16.2
	openMPI	4.0.0
	Local Parameter Device	CPU
Hyperparameter	Dataset	ILRSV2012
	Model	Resnet_v1.5
	Batch Size	192
	epoch	30
	Variable update	horovod
	precision	fp16

## run in two node

Optimizer/ parameters in use	initial learning rate	warm up	learning rate decay factor	num epoch per decay
SGD	0.01	5 batches	0.9	1
Momentum	0.8	1 epoch	0.99	0.03
RMSprop	0.01	5 batches	None	None
Adam	0.0001	5 batches	None	None

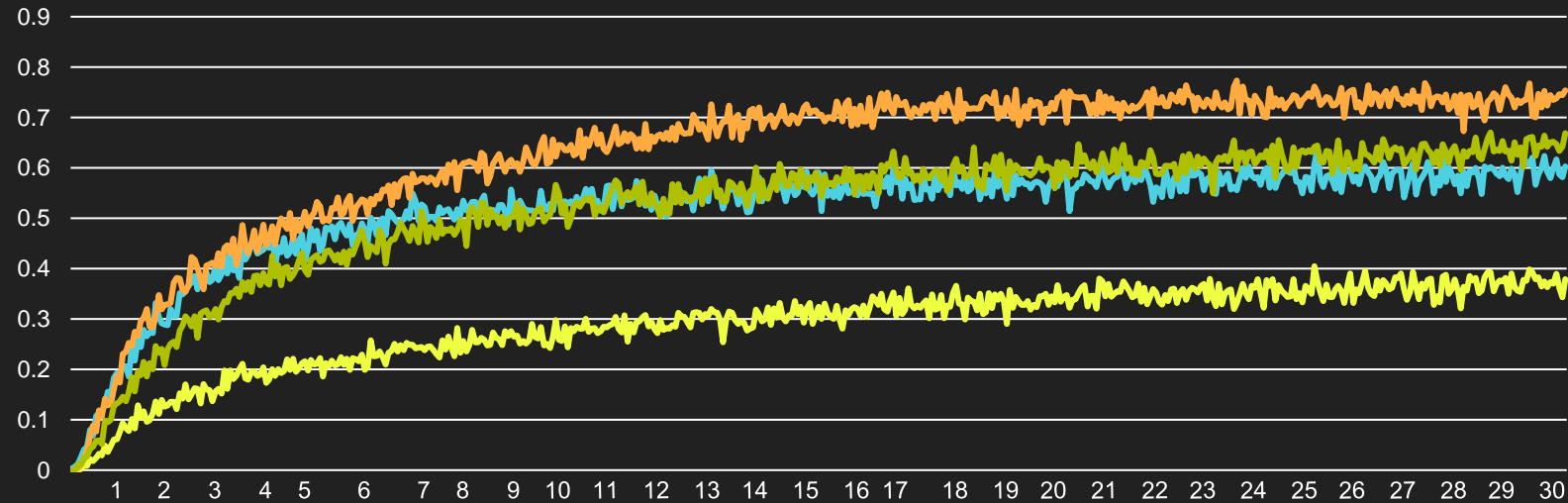


# Result



## Loss

- SGD
- adam
- momentum
- RMSprop



## Top 1

- SGD
- adam
- momentum
- RMSprop



# Hyperparameter choice - Strategies about learning rate

Compare two different strategies:

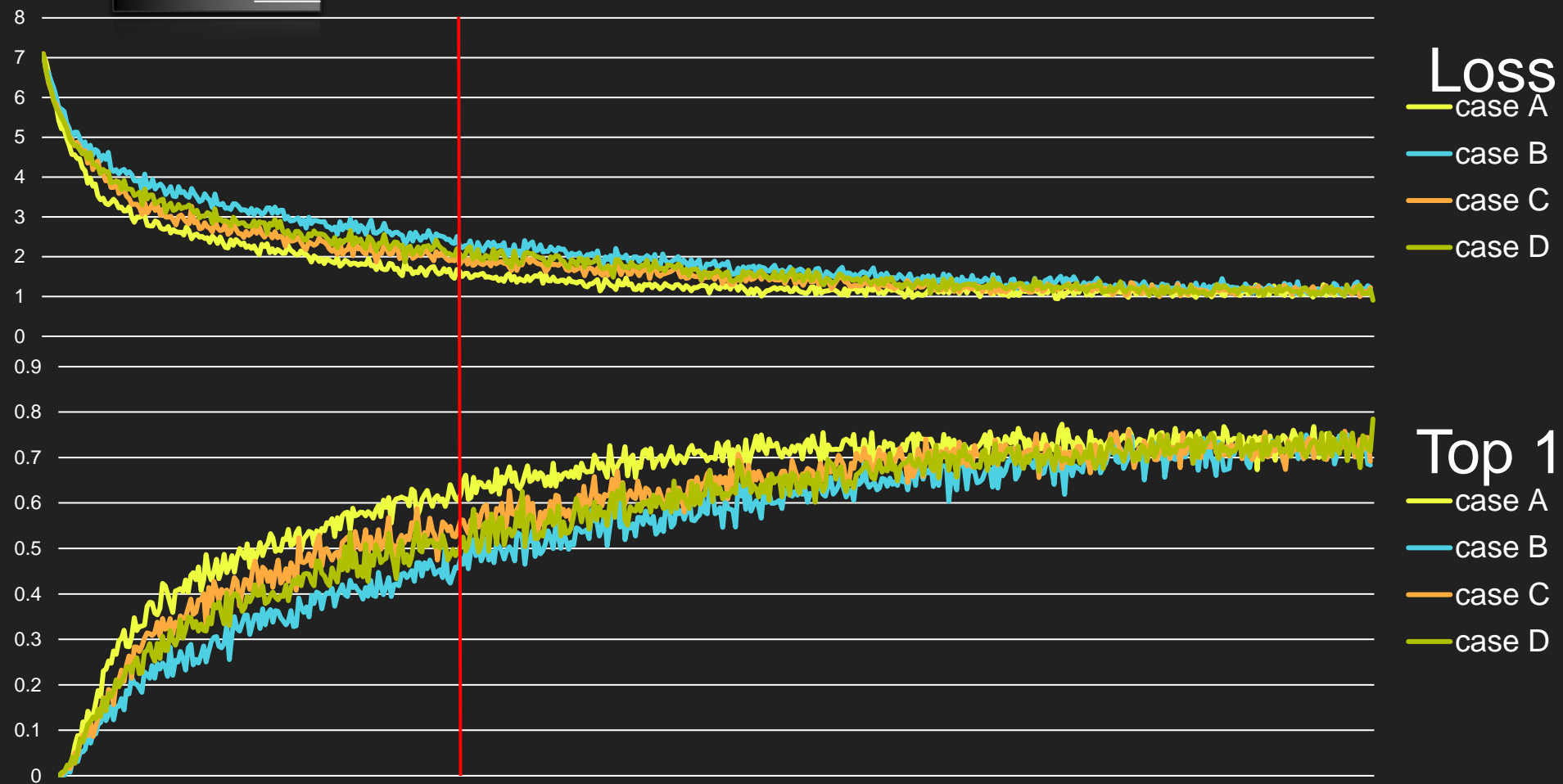
1. Large batch size to read dataset as many times as possible in limit times
2. Small batch size to get more time to update parameters

	case A	case B	case C	case D
Optimizer	Momentum			
batch size	<b>192</b>	<b>64</b> $\sqrt{K}$	64	64
initial learning rate	0.8	0.8	<b>0.27</b>	<b>0.46</b>
warm up epoch	1	1	1	1
learning decay factor	0.99	0.99	0.99	0.99
num epoch per decay	0.03	0.03	0.03	0.03

Diagram illustrating hyperparameter choices for Momentum optimizer across four cases (A, B, C, D). The table shows batch size, initial learning rate, warm up epoch, learning decay factor, and num epoch per decay. Case A uses a large batch size (192) and a high initial learning rate (0.8). Case B uses a smaller batch size (64) and a high initial learning rate (0.8). Case C uses a smaller batch size (64) and a lower initial learning rate (0.27). Case D uses a smaller batch size (64) and a higher initial learning rate (0.46). A blue arrow indicates the relationship between batch size and initial learning rate for cases A, B, and C, showing a decrease in learning rate as batch size decreases. A red arrow indicates the relationship between batch size and initial learning rate for cases B and C, showing an increase in learning rate as batch size decreases.



# Result





# The best result on our cluster

Our cluster		
Environment	CPU	Core i9-9900X
	GPU	Nvidia 2080Ti * 2
	OS	Ubuntu 16.04 LTS
	CUDA / driver	10.0.130 / 430.14
	cuDNN	7.6.0.64-1
	NCCL	2.4.7-1
	Infiniband	EDR
	Tensorflow	1.13
	Benchmark github hash	cnn_tf_v1.13
	Horovod	0.16.2
	openMPI	4.0.0
	Local Parameter Device	CPU
	Dataset	ILRSV2012
Hyperparameter	Model	Resnet_v1.5
	Variable update	horovod
	precision	fp16
	distortions	no

Optimizer	Momentum
Initial learning rate	0.46
Batch size	64
Epochs	20
Learning rate decay factor	0.99
Num epoch per decay	0.03
Warm up epoch	1

## Evaluate result

Top 1 accuracy	Top 5 accuracy
73.22%	91.25%



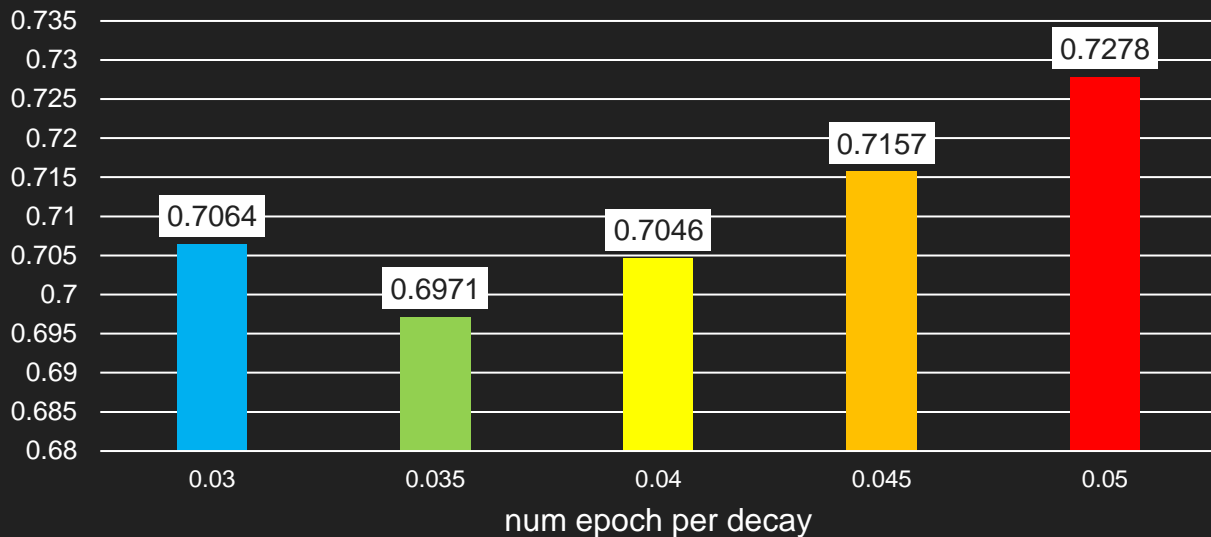


# Results on 2 DGX node in 90 minutes

Top 1

NSCC

Environment	CPU	2 x Xeon 2698 V4
	GPU	2 x DGX1 (16xV100)
	OS	Ubuntu 16.04 LTS
	CUDA / driver	9.0.176 / 384.81
	cuDNN	7.6
	NCCL	2.4.7
	Infiniband	EDR
	Tensorflow	1.13
	Benchmark github branch	cnn_tf_v1.13
	Horovod	0.16.2
	openMPI	3.1.3
	Local Parameter Device	CPU
	Dataset	ILRSV2012
Hyperparameter	Model	Resnet_v1.5
	Variable update	horovod
	precision	fp16
	distortions	no



Optimizer	Initial learning rate	Batch size	Epochs	Learning rate decay factor	Warm up epoch
momentum	0.8	240	40	0.99	3



# The best result on 2 DGX nodes in 90 minutes

NSCC		
Environment	CPU	2 x Xeon 2698 V4
	GPU	2 x DGX1 (16xV100)
	OS	Ubuntu 16.04 LTS
	CUDA / driver	9.0.176 / 384.81
	cuDNN	7.6
	NCCL	2.4.7
	Infiniband	EDR
	Tensorflow	1.13
	Benchmark github branch	cnn_tf_v1.13
	Horovod	0.16.2
	openMPI	3.1.3
	Local Parameter Device	CPU
	Dataset	ILRSV2012
Hyperparameter	Model	Resnet_v1.5
	Variable update	horovod
	precision	fp16
	distortions	no

Optimizer	Momentum
Initial learning rate	0.4
Batch size	240
Epochs	45
Learning rate decay factor	0.99
Num epoch per decay	0.05
Warm up epoch	5

## Evaluate result

Loss	Top 1	Top 5	images/sec
0.891	73.84%	91.63%	11720.22

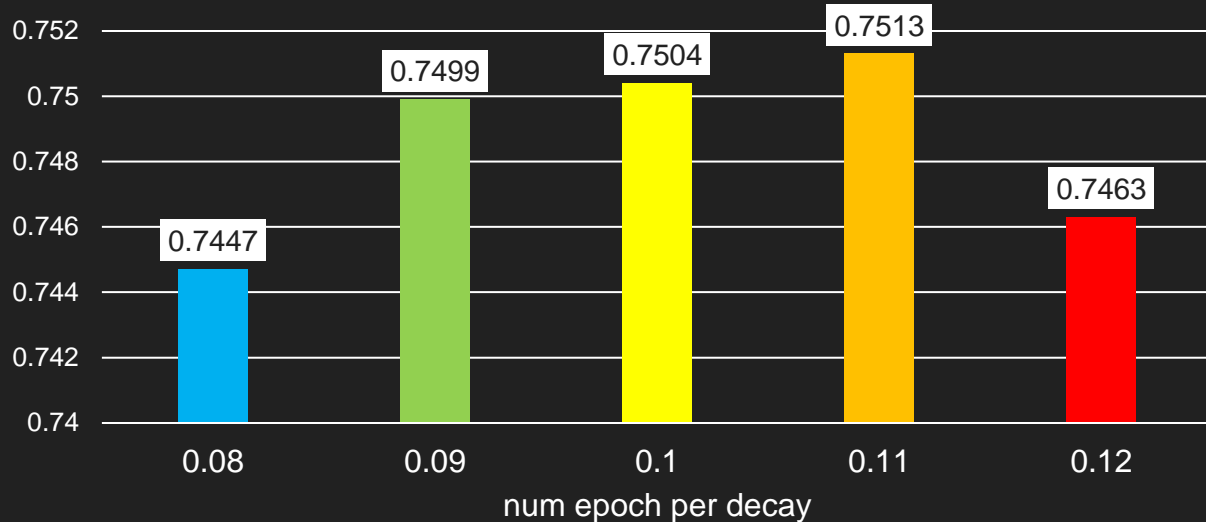


# Results on 2 DGX node in 180 minutes

NSCC

Environment	CPU	2 x Xeon 2698 V4
	GPU	2 x DGX1 (16xV100)
	OS	Ubuntu 16.04 LTS
	CUDA / driver	9.0.176 / 384.81
	cuDNN	7.6
	NCCL	2.4.7
	Infiniband	EDR
	Tensorflow	1.13
	Benchmark github branch	cnn_tf_v1.13
	Horovod	0.16.2
	openMPI	3.1.3
	Local Parameter Device	CPU
	Dataset	ILRSV2012
	Model	Resnet_v1.5
Hyperparameter	Variable update	horovod
	precision	fp16
	distortions	no

Top 1



Optimizer	Initial learning rate	Batch size	Epochs	Learning rate decay factor	Warm up epoch
momentum	0.8	240	90	0.99	10



# The best result on 2 DGX nodes in 180 minutes

NSCC		
Environment	CPU	2 x Xeon 2698 V4
	GPU	2 x DGX1 (16xV100)
	OS	Ubuntu 16.04 LTS
	CUDA / driver	9.0.176 / 384.81
	cuDNN	7.6
	NCCL	2.4.7
	Infiniband	EDR
	Tensorflow	1.13
	Benchmark github branch	cnn_tf_v1.13
	Horovod	0.16.2
	openMPI	3.1.3
	Local Parameter Device	CPU
	Dataset	ILRSV2012
Hyperparameter	Model	Resnet_v1.5
	Variable update	horovod
	precision	fp16
	distortions	no

Optimizer	Momentum
Initial learning rate	0.4
Batch size	240
Epochs	90
Learning rate decay factor	0.99
Num epoch per decay	0.1
Warm up epoch	10

## Evaluate result

Loss	Top 1	Top 5	images/sec
0.691	75.35%	92.48%	11697.49

Part II –  
High Performance Computing  
Swiftsim

# Analyze yaml

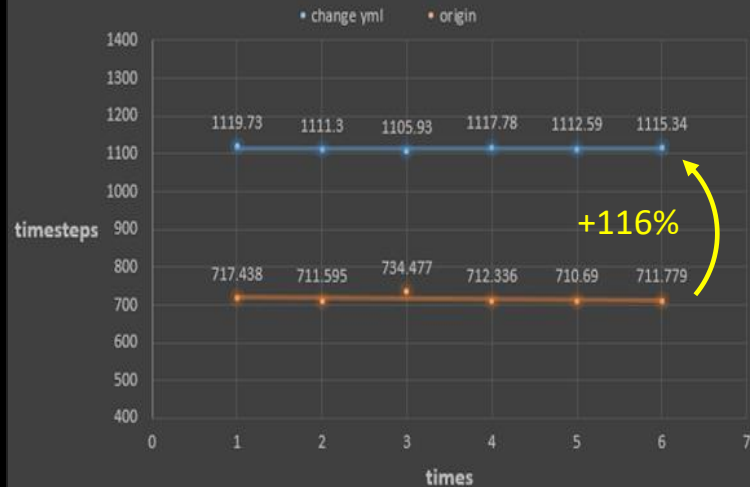
```
Scheduler:
  max_top_level_cells: 16
  cell_max_size: 8000000
  cell_sub_size_pair_hydro: 256000000
  cell_sub_size_self_hydro: 32000
  cell_sub_size_pair_grav: 256000000
  cell_sub_size_self_grav: 32000
  cell_split_size: 400
  cell_subdepth_diff_grav: 4
  cell_extra_parts: 0
  cell_extra_parts: 0
  cell_extra_parts: 0
  cell_extra_parts: 0
  nr_queues: 8
  tasks_per_cell: 0
  links_per_tasks: 10
  mpi_message_limit: 4
```

```
DomainDecomposition:
  initial_type: memory
  repartition_type: fullcosts
  trigger: 0.05
  minfrac: 0.95
  usemetis: 0
  adaptive: 1
  itr: 100
  use_fixed_costs: 0
```

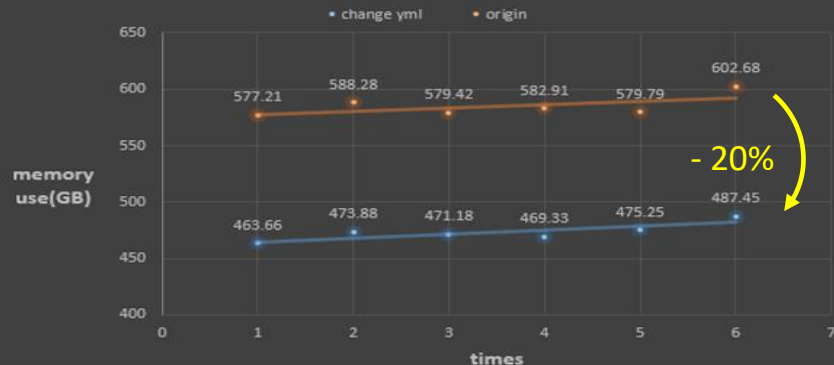
```
Scheduler:
  max_top_level_cells: 37
  cell_max_size: 8000000
  cell_sub_size_pair_hydro: 256000000
  cell_sub_size_self_hydro: 12800
  cell_sub_size_pair_grav: 256000000
  cell_sub_size_self_grav: 12800
  cell_split_size: 200
  cell_subdepth_diff_grav: 4
  cell_extra_parts: 0
  cell_extra_parts: 0
  cell_extra_parts: 0
  cell_extra_parts: 0
  nr_queues: 96
  tasks_per_cell: 0
  links_per_tasks: 256
  mpi_message_limit: 4096
```

```
DomainDecomposition:
  initial_type: memory
  repartition_type: none
  trigger: 0.05
  minfrac: 0.95
  usemetis: 0
  adaptive: 1
  itr: 100
  use_fixed_costs: 0
```

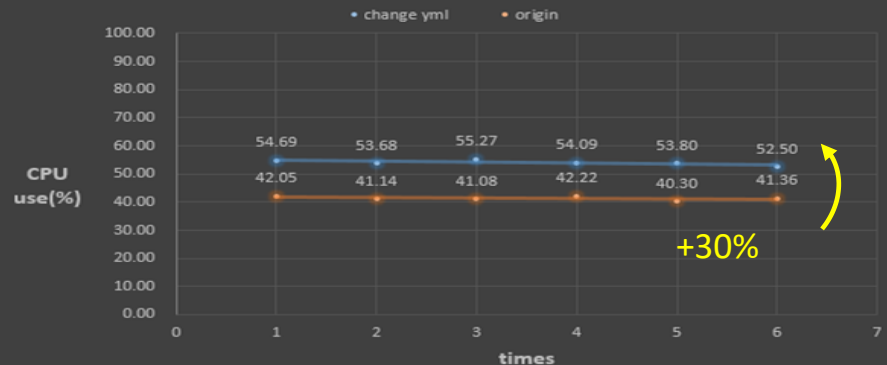
## Compare to timesteps



## Compare to memory use

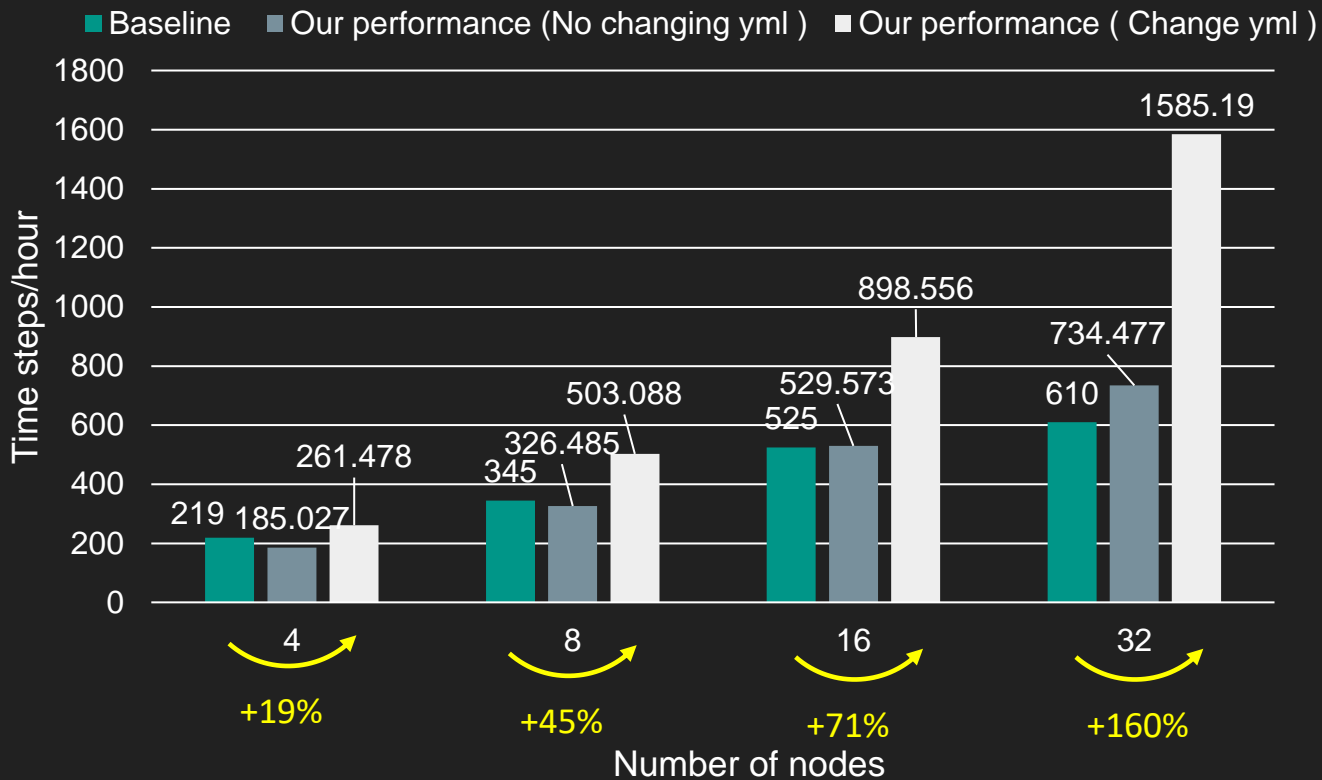


## Compare to percentage of CPU use



# Our Best Result

## SWIFT EAGLE\_50 performance



NSCC		
Environment	CPU	E5-2690v3
	RAM	96G DDR4
	OS	CentOS release 6.10 LTS
	IntelMPI	2018.0.128
	Infiniband	EDR
Library	HDF5	1.10.4
	FFTW	3.3.8
	GSL	2.5
	metis	5.1.0
	parmetis	4.0.3
	swiftsim	3d44fb65ea39b9f7a2a99525f15c4cd464045c38