Applications Software Example

How to run an application on Cluster?

Rooh Khurram

Supercomputing Laboratory
King Abdullah University of Science and Technology (KAUST), Saudi Arabia

Cluster Training: Applications
17 October 2017, Thuwal, Saudi Arabia

Outline

- Fluent example
- Jobscripts
- Performance
  - What to expect on cluster?
  - Sharing early experience
- Material will be shared with you
- Large scale Fluent users should transition to Shaheen
- Shaheen beginners can find examples here:
  - Shaheen users can find all material here: /scratch/tmp/apps/
- Announcement about the 2nd KAUST-ANSYS Workshop

Test Case

This is a medium scale test case (4 million cells). This test case uses around 10GB of memory on one node. This is a numerical simulation of external flow over a passenger sedan car.

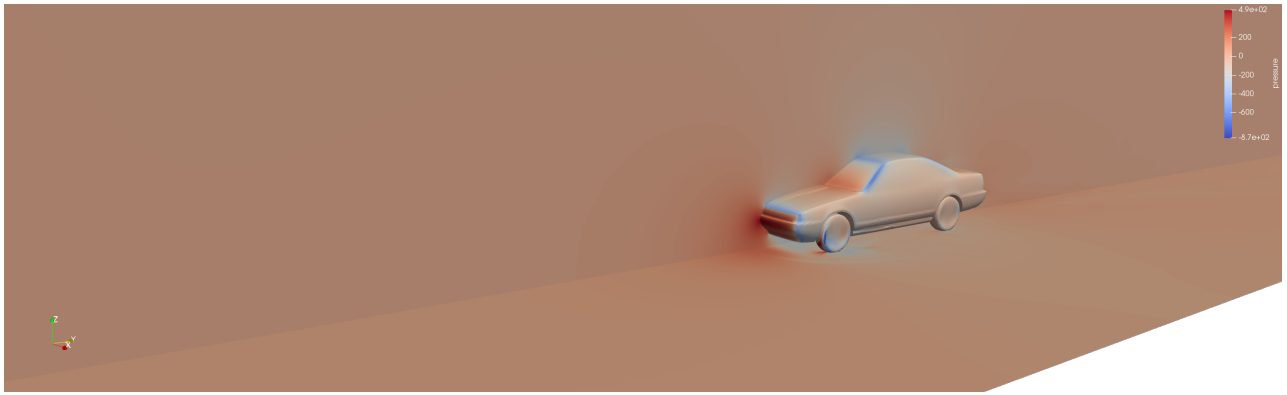


Figure: Pressure contours on the surface of a sedan car

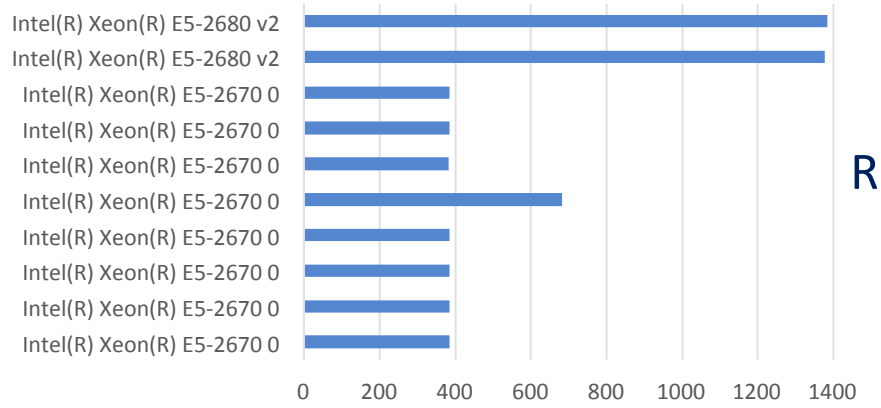
Code: Fluent v17
Size: 4M cells
Cell Type: Mixed
Solver: Pressure based coupled solver, Green-Gauss cell based, steady state
Models: Standard K-e Turbulence

Jobscript

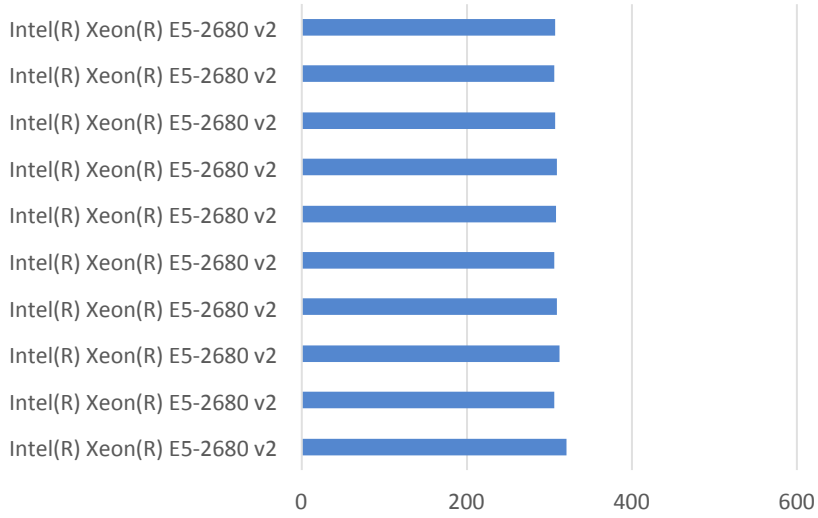
```
#!/bin/bash
#SBATCH -t 01:00:00
#SBATCH -J fluent
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=16
#SBATCH --constraint=intel
#SBATCH --mem=65536
# Print hostname job executed on
echo "My NODELIST is: $SLURM_NODELIST"
module load legacy
module load ansys
# activating slurm support
export FLUENT_ENABLE_SLURM_SUPPORT=1
# launching fluent on 1 node(s) x 16 cores = 16
time fluent -t16 3ddp -g -nmon -i sedan_4m.in <<EOF
exit
OK
EOF
```

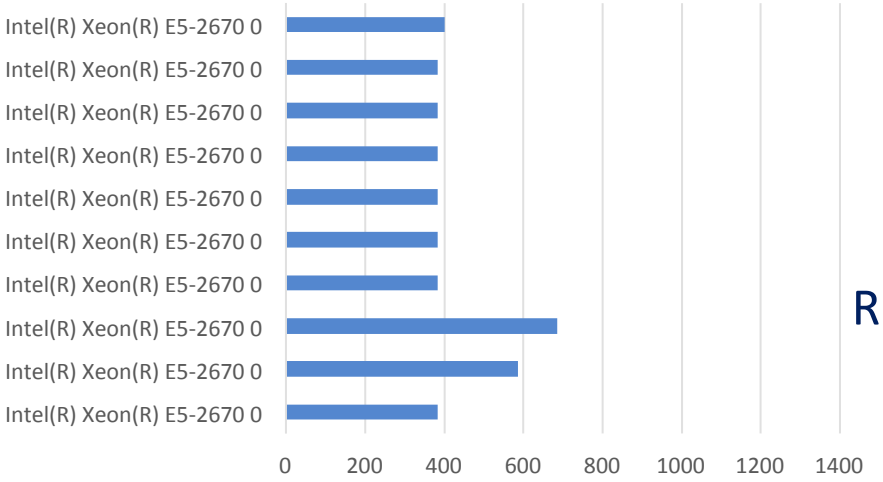
type	cpu_model	cores	nodes
Intel	Haswell	28	16
		32	1
	Ivy Bridge	20	177
	Sandy Bridge	16	100
	Westmere	12	1
		64	1
		80	1

Repeatability Test

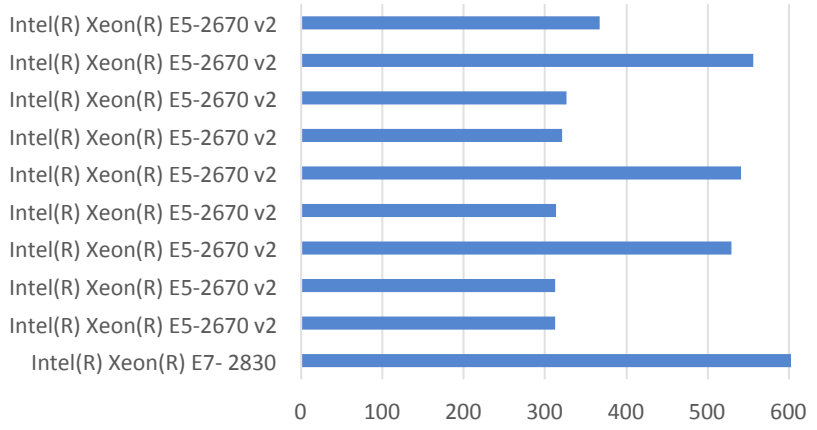


Run # 1





Run # 2



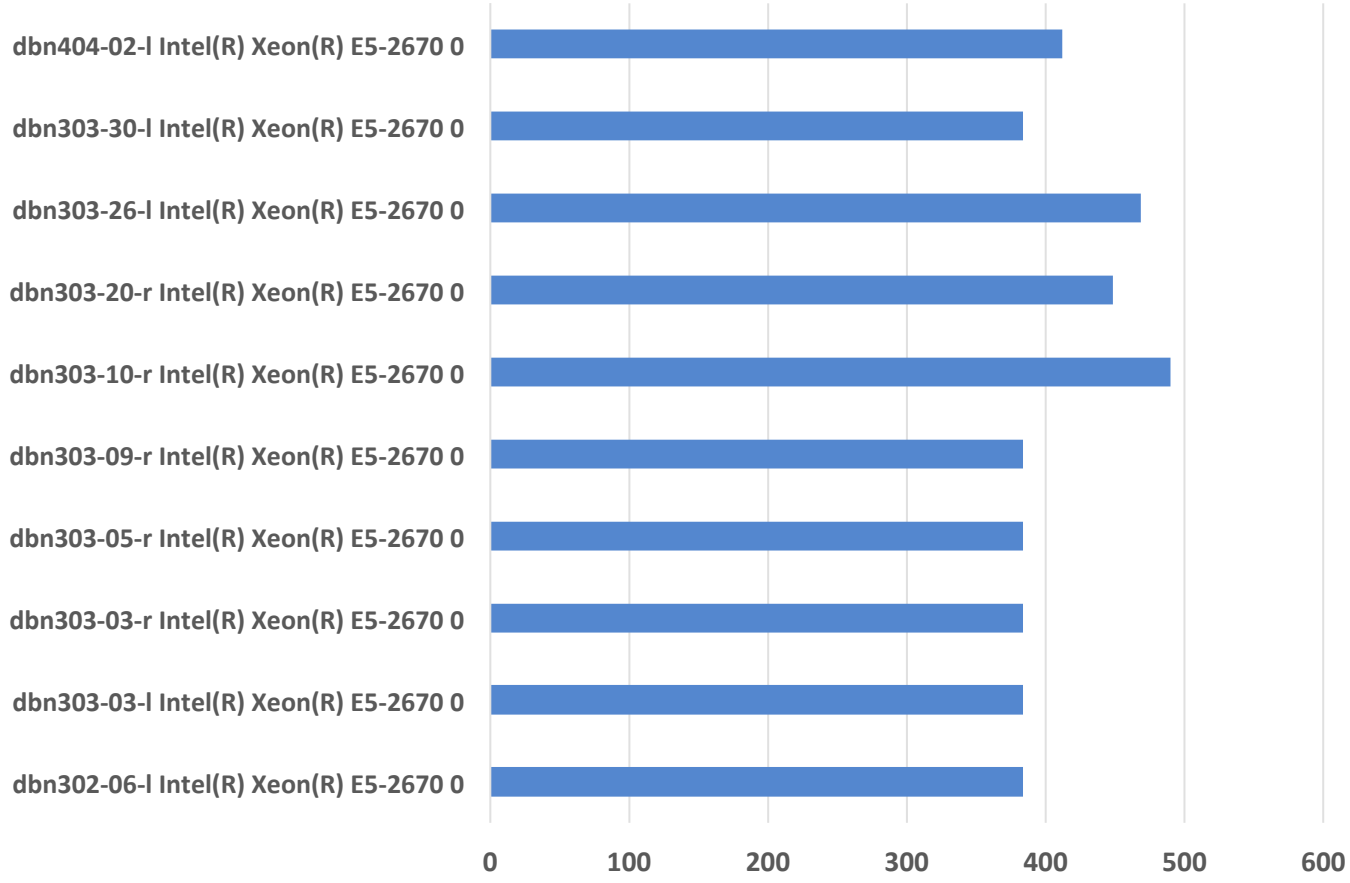
16 cores

20 cores

Running Jobs on Exclusive Nodes

```
#!/bin/bash
#SBATCH -t 01:00:00
#SBATCH -J fluent
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=16
#SBATCH --constraint=intel
#SBATCH --mem=65536
#SBATCH --exclusive
# Print hostname job executed on
echo "My NODELIST is: $SLURM_NODELIST"
module load legacy
module load ansys
# activating slurm support
export FLUENT_ENABLE_SLURM_SUPPORT=1
# launching fluent on 1 node(s) x 16 cores = 16
time fluent -t16 3ddp -g -nmon -i sedan_4m.in <<EOF
exit
OK
EOF
```

Repeatability Test

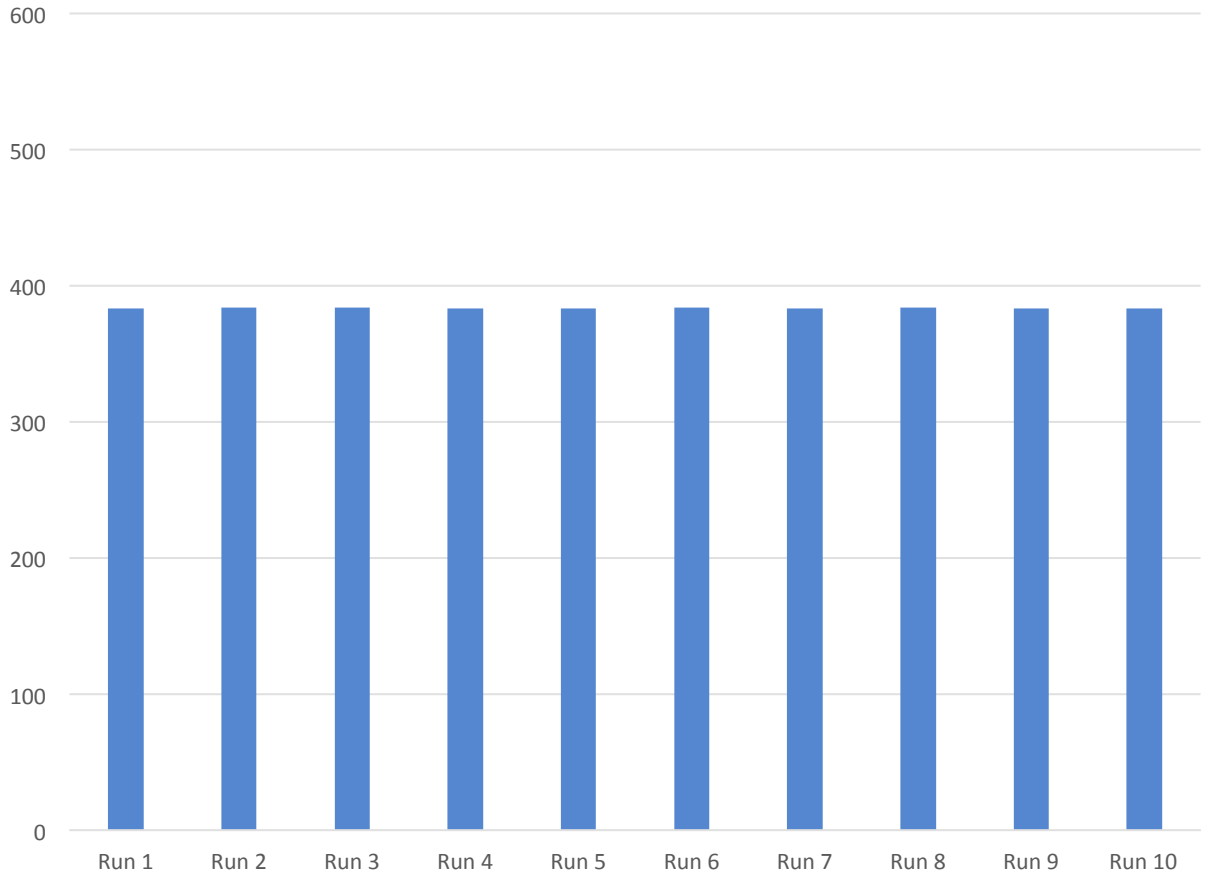


16 cores

Running Jobs on Exclusive Nodes after Reboot

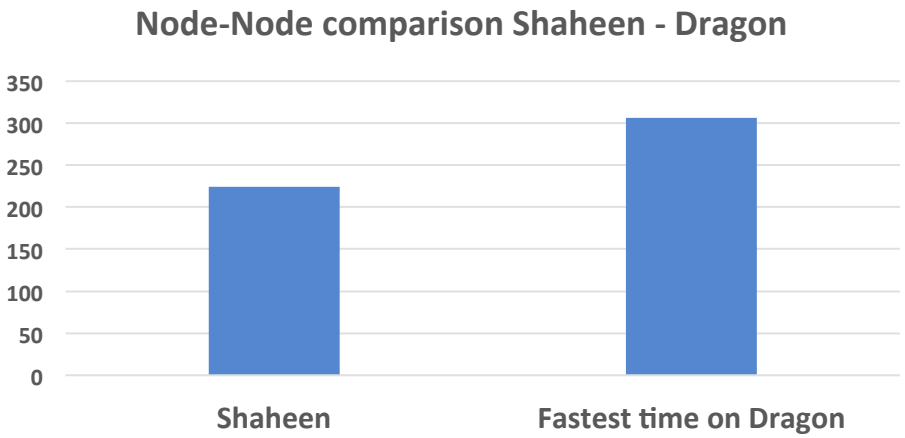
```
#!/bin/bash
#SBATCH -t 01:00:00
#SBATCH -J fluent
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=16
#SBATCH --constraint=intel
#SBATCH --mem=65536
#SBATCH –exclusive
#SBATCH -w dbn303-10-r
# Print hostname job executed on
echo "My NODELIST is: $SLURM_NODELIST"
module load legacy
module load ansys
# activating slurm support
export FLUENT_ENABLE_SLURM_SUPPORT=1
# launching fluent on 1 node(s) x 16 cores = 16
time fluent -t16 3ddp -g -nmon -i sedan_4m.in <<EOF
exit
OK
EOF
```

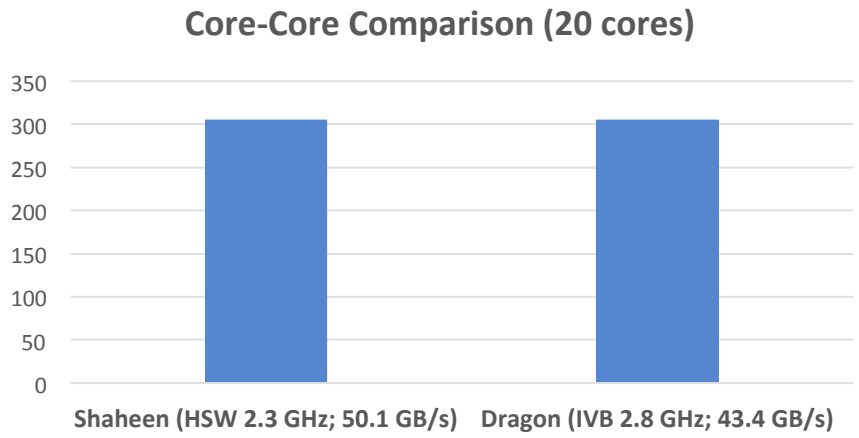
Repeatability Test



To test same “slow” node repeatedly (dbn303-10-r Intel(R) Xeon(R) E5-2670 0)

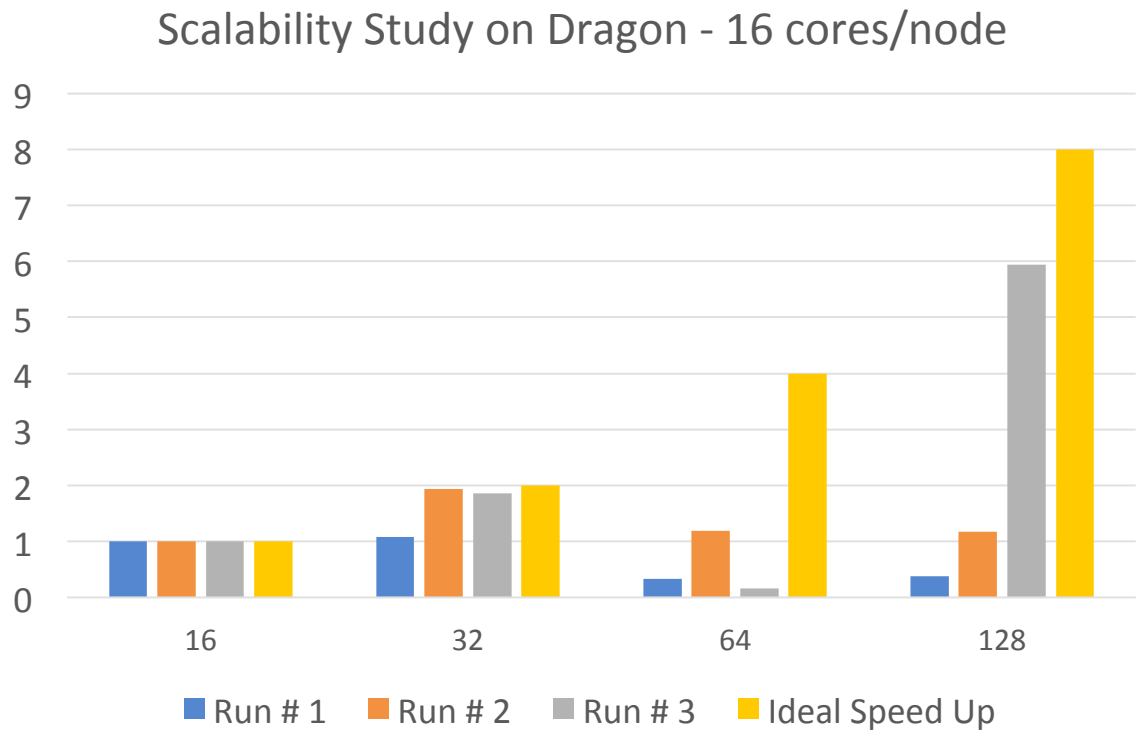
Rough Estimate of Baseline Performance on Cluster





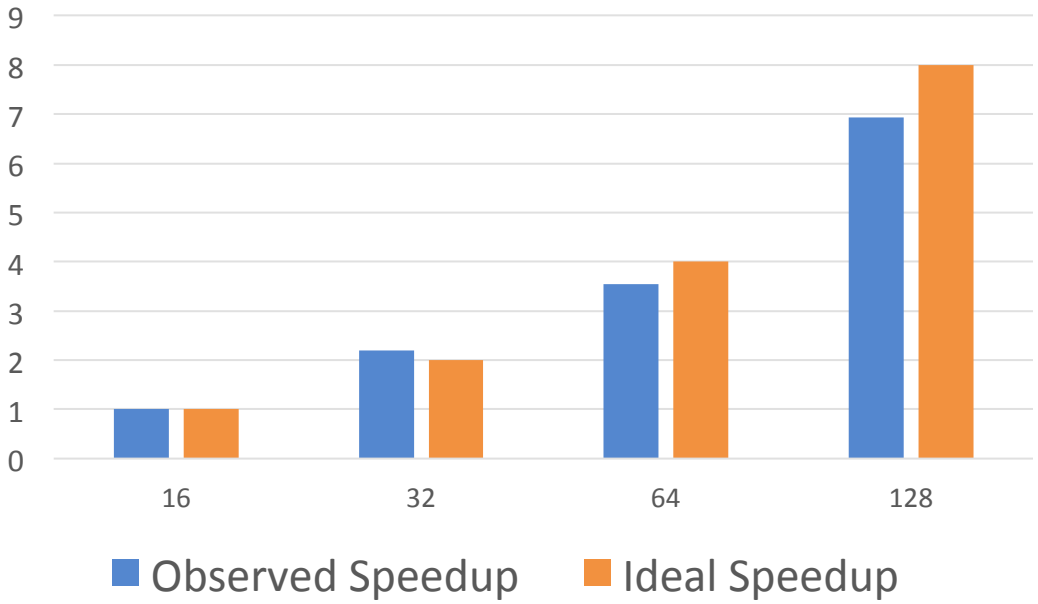
1-Node baseline comparison of the fastest run on Dragon and reference Shaheen run

Scalability Test



fluent –t128 3ddp -g -slurm -nmon -i sedan_4m.in

Scalability Test

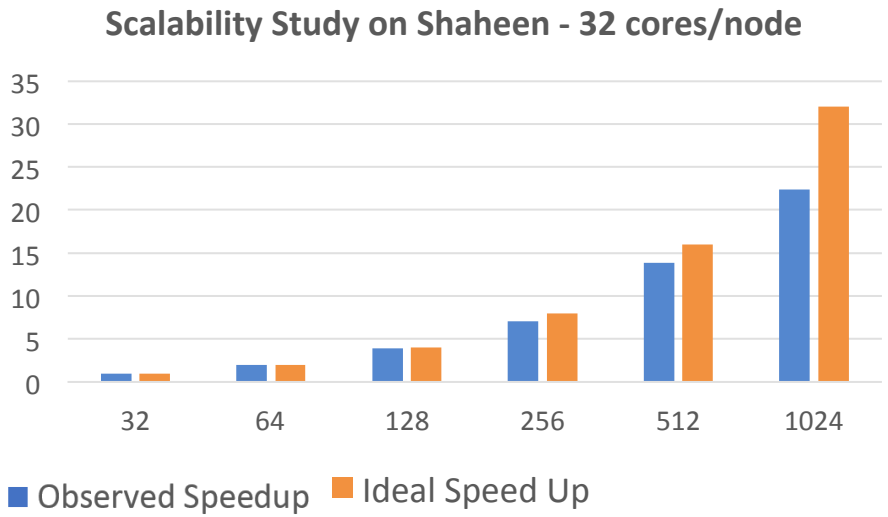


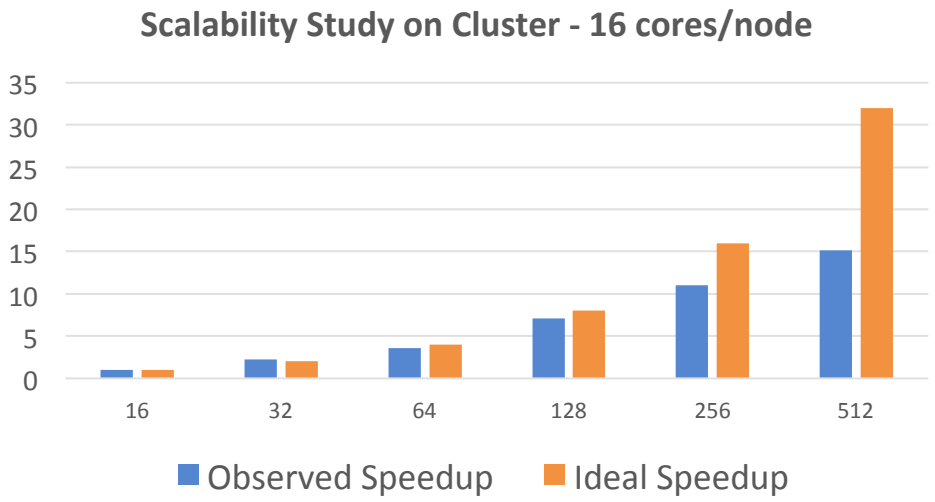

```
 host | 0    1    2    3    4    5    6    7
======|================================
    0 : SHM  IBV  IBV  IBV  IBV  IBV  IBV  IBV
    1 : IBV  SHM  IBV  IBV  IBV  IBV  IBV  IBV
    2 : IBV  IBV  SHM  IBV  IBV  IBV  IBV  IBV
    3 : IBV  IBV  IBV  SHM  IBV  IBV  IBV  IBV
    4 : IBV  IBV  IBV  IBV  SHM  IBV  IBV  IBV
    5 : IBV  IBV  IBV  IBV  IBV  SHM  IBV  IBV
    6 : IBV  IBV  IBV  IBV  IBV  IBV  SHM  IBV
    7 : IBV  IBV  IBV  IBV  IBV  IBV  IBV  SHM

Prot -  All Intra-node communication is: SHM
Prot -  All Inter-node communication is: IBV
```

fluent –t128 –pinfiniband 3ddp -g -slurm -mmon -i sedan_4m.in

Comparison and Remarks





- Performance comparison of Infiniband (Fat-tree topology) with Aires (Dragonfly topology)
- On 32-node run, Dragon's speedup is 33% less than Shaheen. But, the important thing is that it kept scaling till 32 nodes (512 cores)
- At scale, the users should transition to Shaheen for better performance

KAUST-ANSYS Workshop

Objective:

To engage academic & industrial partners, to educate engineering students, and to prepare KSL for providing computational science and engineering services at KAUST

- Last Conference held on April 16, 2017
- Tentative date: Spring 2018
- ANSYS/Fluid Codes experts will be at KAUST
- Full day workshop
- Hands on sessions
- HPC focus
- KSL will be sending the invites soon