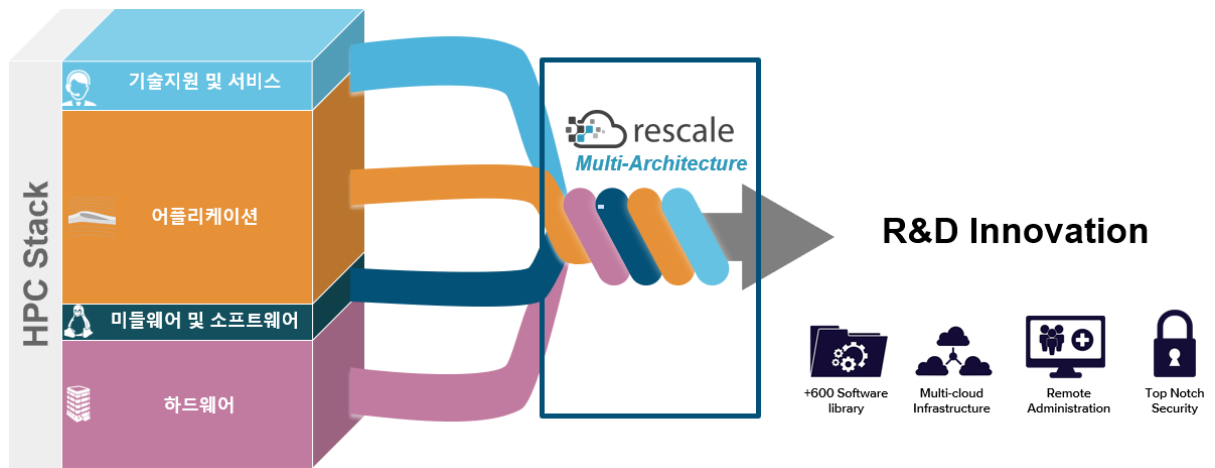


Intelligent Computing for Digital R&D

[Rescale 플랫폼 소개]

Rescale의 ScaleX 플랫폼은 Public 클라우드를 기반으로 CAE를 위하여 필요한 다양한 소프트웨어 및 하드웨어, 관리 시스템을 포함하는 플랫폼이다. 사용자들은 Rescale 플랫폼에서 AWS, Azure, GCP 등 다양한 클라우드 업체의 연산 자원들을 활용하여 각 워크로드별로 최적의 하드웨어 유형을 선택할 수 있어 시뮬레이션 소요 시간을 단축하고, 기존 On-premise에서 연산 자원의 한계로 수행하기 어려웠던 대규모 시뮬레이션을 진행할 수 있다.



아울러 Rescale ScaleX 플랫폼은 HPC 운용에 필요한 모든 항목들을 단일 플랫폼에서 제공하므로 이를 통하여 IT 자원 관리의 효율성을 향상할 수 있으며, HPC 클라우드 환경을 제공하는 것뿐만 아니라 On-premise와의 하이브리드 구성 등 기업이 보유하고 있는 기존 자원을 최대로 활용하면서 HPC 클라우드의 장점을 최대로 누릴 수 있도록 지원하고 있다.

Rescale 플랫폼의 특징을 요약하면 다음과 같다.

- 600개 이상의 어플리케이션(소프트웨어)

분류	주요 소프트웨어
Commercial	Fluent, CFX, STAR-CCM+, ABAQUS, LS-DYNA, HyperWorks, MATLAB, Nastran, HFSS, CST, PowerFlow, MoldFlow, etc.
Open-Source	OpenFOAM, SU2, GROMACS, LAMMPS, CalculiX, Code_Aster, etc.
Container	Singularity
Bring Your Own	In-House code using MPICH, Intel MPI, Open MPI, Platform MPI
Others	FireFox, VS Code, PyCharm, Anaconda, BeeGFS, Intel Parallel Studio

- 100개 이상의 코어타입(하드웨어 유형)

분류	설명 및 주요 용도
General Purpose	일반적인 사양의 유형으로 다양한 작업에 대응 가능
High Interconnect	노드간 데이터 전송속도가 빠른 유형(500코어 이상 필요한 작업)
High Memory	대용량 메모리로 구성된 유형(코어당 16GB, 노드당 256GB 이상)
High Clock-speed	CPU 주파수가 높은 유형(적은 수라도 빠른 CPU가 필요한 작업)
High Disk	대용량 스토리지로 구성된 유형(결과의 크기가 수 TB이상인 작업)
GPU	GPU로 구성된 유형(머신러닝/딥러닝, GPGPU 활용 작업)

- 관리자 포털

- 효율적인 플랫폼 사용을 위한 성능, 비용, 보안 대시보드 제공
- 팀, 프로젝트별 예산, 사용 가능 어플리케이션 및 코어타입 설정 등 개별적으로 플랫폼 최적화를 위한 설정 기능 제공

[Rescale 플랫폼에서의 시뮬레이션]

앞서 소개한 내용과 같이 Rescale 플랫폼은 HPC에서 필요한 모든 항목들이 단일 플랫폼에 구축되어 있으며 사용자의 업무 환경, 특성에 맞추어 최적화할 수 있도록 다양한 작업 유형 및 관련 기능들을 제공하며 이를 요약하면 다음 그림과 같다.

The screenshot shows the Rescale platform interface. At the top, there is a navigation bar with 'rescale' logo and menu items like 'New Job', 'Jobs', 'Desktops', 'Storage Devices', 'Files', 'Transfers', 'Help', and a user profile 'junghoon@rescale.com'. Below the navigation bar, there are buttons for 'Create a job' and 'Workspace'. The main content area features a 'Welcome to Rescale' message with a sub-message: 'Rescale makes running simulations fast and easy, so you can focus on discovery, not on the infrastructure. Click "Run Simulation" on any of the below to get started. Click "Read Tutorial" for a detailed walkthrough.' Below this message is a table summarizing the supported interfaces and modes.

Interface	Jobs mode	Interactive	Virtual desktop
Rescale WebUI	Basic	In-Browser SSH	GUI based processing
API	DOE	Livetail for logs	CPU / GPU
CLI	Optimization	API SSH	Windows
	End-to-End Desktop	End-to-End Desktop	Linux

사용자가 작업을 실행할 수 있는 방법은 총 3가지이며 각각의 특징은 다음과 같다.

- Rescale WebUI: 가장 일반적으로 사용하는 방법으로 웹 페이지에 접속하여 입력 파일을

업로드하고, 사용할 소프트웨어 및 하드웨어 설정을 완료한 후 작업 실행

- Rescale CLI: 작업 실행에 필요한 항목들을 Rescale에서 프로그램으로 제작한 것으로 사용하는 이를 활용하여 WebUI에 접속하지 않고 간단한 명령어를 통하여 작업 실행
- Rescale API: CLI에서 수행하기 어려운 복잡한 절차의 시뮬레이션의 경우 사용자가 Python 혹은 CURL을 활용하여 스크립트로 구성하여 WebUI에 접속하지 않고 작업 실행

Rescale 플랫폼에서 제공하는 작업 유형은 총 4가지이며 각각의 특징은 다음과 같다.

- Basic: Rescale 플랫폼에서 가장 많이 사용되는 유형으로 일반적으로 말하는 Batch 작업과 동일하게 하나의 작업을 생성해서 한 개의 시뮬레이션만 수행하거나, 순차적으로 여러 개의 시뮬레이션을 수행 가능
- End-To-End Desktop: 리눅스 기반의 GUI 환경을 제공하는 유형으로 시뮬레이션 진행 도중 수렴 데이터를 확인하며 필요시 진행 중인 작업을 중지하고 해석 파라미터를 변경하여 재시작하는 등 Interactive하게 시뮬레이션을 수행 가능
- Optimization: 파라미터 최적화 시 사용되는 유형으로 Isight, LS-OPT, 그리고 자체 개발한 Python 최적화 코드를 활용할 수 있으며, Basic 유형에서 사용 가능한 모든 시뮬레이션 소프트웨어를 Optimization 유형에서도 사용 가능
- DOE: 시뮬레이션을 활용한 실험계획법 수행 시 사용되는 유형으로 변수를 생성하는 방법과 그에 따른 변화를 반영하는 결과 값을 지정하고 각 케이스를 동시에 여러 개의 클러스터로 계산하여 각 인자의 영향도를 분석 가능

■ Optimization vs DOE

- 1) Optimization은 목적 함수를 만족할 때까지 지정한 파라미터를 조정하면서 반복적으로 하나의 클러스터를 활용하여 계산을 수행
- 2) DOE는 지정한 총 케이스들을 계산을 완료할 때까지 각 변수의 조합들을 여러 개의 클러스터를 활용하여 동시에 계산을 수행
- 3) 예를 들어, Optimization에서 Emerald 코어 타입을 3 노드로 지정하여 클러스터를 생성하면 1개의 시뮬레이션 케이스가 108개의 코어로 계산되며, DOE에서 Emerald 코어 타입을 3 슬롯, 1 노드로 지정하여 클러스터를 생성하면 동시에 3개의 시뮬레이션 케이스가 각각 36코어로 계산됨

Rescale 플랫폼에서는 계산을 위한 작업 유형 외에도 시뮬레이션 모델의 전처리 및 후처리를 수행할 수 있는 Virtual Desktop 또한 제공하며 그 특징은 다음과 같다.

- OS 유형은 윈도우 및 리눅스 모두를 지원하며, GPU 및 대용량 메모리로 구성된 코어 타입들을 기반으로 활용 가능

- 기존에 완료된 시뮬레이션 결과를 가져오거나, 가상 데스크탑 내에서 작업한 내용을 이후 계산 작업에서 사용할 수 있도록 내보내기 가능
- 특히, 연구소 내 인터넷 회선의 속도가 느리거나 계산된 시뮬레이션 결과 파일의 크기가 매우 클 경우(1TB 이상) Virtual Desktop 활용을 추천

■ Virtual Desktop vs End-To-End Desktop

- 1) Virtual Desktop의 경우 시뮬레이션 데이터의 전처리 및 후처리가 주요 목적이므로 정해진 설정 값 외에 코어 수를 변경하거나 여러 개의 노드를 사용하는 것은 불가능
- 2) End-To-End Desktop의 경우 계산이 주요 목적이며 필요 시 사용자가 interactive 하게 작업을 할 수 있도록 GUI를 추가로 제공해주는 것이므로 사용자가 자유롭게 코어 수 혹은 노드 수를 조정하는 것이 가능
- 3) 다만 시뮬레이션 모델의 검증 및 계산 부하가 적은 시뮬레이션의 경우 Virtual Desktop에서 모델 구성 후 이어서 시뮬레이션까지 진행하는 것이 효율적임

[Rescale 플랫폼을 활용 Tip (Basic 작업 유형)]

- 기본적으로 사용 가능한 어플리케이션 실행 명령어 외에도 필요한 명령어를 추가하거나, 시뮬레이션에 사용하는 코어 수를 직접 지정하는 것이 가능

```
OMP_NUM_THREADS=$((RESCALE_CORES_PER_SLOT/RESCALE_SOCKETS_PER_SLOT));
cd $HOME/work/shared
unzip Tank_Sloshing.zip
cd scene
mpirun -machinefile $MACHINEFILE -np 1 app.solver.double -p . -k simd -n 6
mpirun -machinefile $MACHINEFILE -np 1 app.solver.double -p . -k simd -n 12
mpirun -machinefile $MACHINEFILE -np 1 app.solver.double -p . -k simd -n 24
mpirun -machinefile $MACHINEFILE -np 1 app.solver.double -p . -k cuda -n 6 -g 0
mpirun -machinefile $MACHINEFILE -np 1 app.solver.double -p . -k cuda -n 12 -g 0 -g 1
mpirun -machinefile $MACHINEFILE -np $RESCALE_SOCKETS_PER_SLOT app.solver.double -p . -k cuda -n 12 -g 0 -g 1
mpirun -machinefile $MACHINEFILE -np $RESCALE_SOCKETS_PER_SLOT app.solver.double -p . -k cuda -n 24 -g 0 -g 1 -g 2 -g 3
mpirun -machinefile $MACHINEFILE -np $RESCALE_SOCKETS_PER_SLOT app.solver.double -p . -k cuda -n $OMP_NUM_THREADS -g 0 -g 1 -g 2 -g 3
```

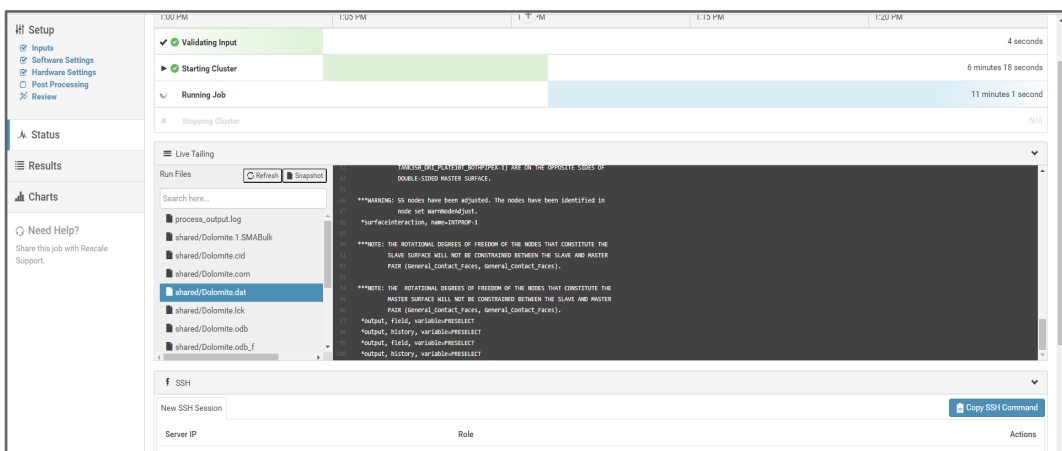
- 자동으로 생성되는 환경 변수로 계산에 사용할 코어 수를 지정하거나, 해당 파라미터에 직접 숫자를 입력 가능(-np 이후 항목 참조)
- 어플리케이션 실행 명령어를 여러 줄로 입력하여 하나의 입력 파일에 대하여 코어 수에 따른 성능 평가를 수행하거나, 여러 개의 입력 파일을 업로드한 후 순차적으로 시뮬레이션 수행 가능
- ANSYS HPC Pack과 같이 코어 수에 따라서 Pack 사용량이 달라지는 경우 라이선스의 효율적인 사용을 위하여 생성한 클러스터의 모든 코어를 활용하는 것이 아닌 라이선스의 제약에 맞추도록 설정하는 것이 가능

예를 들어, HPC Pack을 4개 가지고 있는 경우 Emerald 3 노드(144 코어)로 클러스터를 생성하여 시뮬레이션을 실행할 경우 12코어를 더 쓰기 위해서 Pack 1개가 추가로 사용되나,

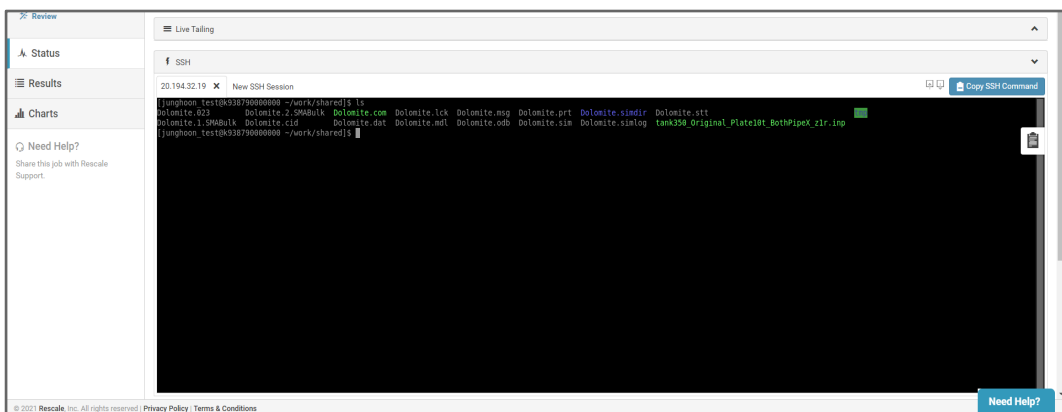
Pack이 1개 증가 시 활용 가능한 코어 수가 4배가 되는 것을 고려하면 불필요한 낭비가 되므로 하드웨어 자원을 일부 활용하지 않더라도 코어 수를 작게 지정하는 것이 필요

```
fluent 3ddp -gu -ssh -cnf=$FLUENT_HOSTS -t$RESCALE_CORES_PER_SLOT -i <journal filename> -driver x11 >> using 144 Cores
fluent 3ddp -gu -ssh -cnf=$FLUENT_HOSTS -t132 -i <journal filename> -driver x11 >> using 132 Cores
```

- \$RESCALE_CORES_PER_SLOT을 사용하면 자동으로 활용 가능한 모든 코어 수를 시스템 변수에서 확인 후 시뮬레이션 시 사용
 - 숫자를 지정하여 입력하면 해당 코어 수만큼만 시뮬레이션 시 사용
- Live Tailing은 시뮬레이션 진행 상황을 실시간으로 확인할 수 있는 탭으로 다음과 같은 기능이 있음



- 시뮬레이션 결과로 생성되는 텍스트 파일들(log, message, out 등)의 실시간 모니터링
 - 시뮬레이션 결과로 생성되는 그림 파일들(jpg, png 등)의 확인
 - 시뮬레이션 진행 도중 좌측의 Snapshot 버튼을 활용하여 현재까지 생성된 결과를 압축하여 저장한 후 Files 페이지에서 해당 파일을 다운로드 가능
- In-Browser terminal은 시뮬레이션이 시작되면 Live Tailing과 함께 자동으로 활성화되는 기능으로 계산이 진행 중인 노드에 원격으로 접속하여 특정 파일의 모니터링, 중간 결과의 압축 등의 작업을 수행 가능

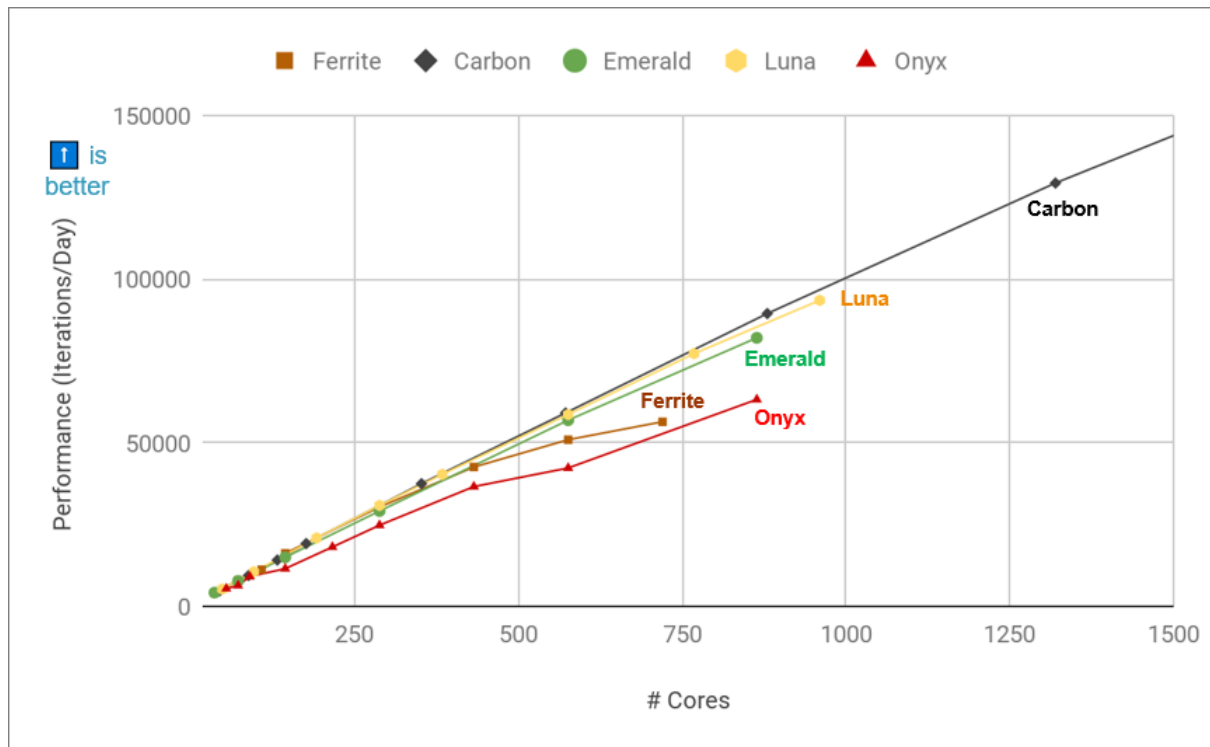


- 2개 이상의 소프트웨어를 활용하여 Coupling 시뮬레이션 수행이 필요할 경우 소프트웨어 선택 창에서 필요한 소프트웨어들을 모두 선택하면 클러스터 생성 시 모두 로드되어 동시에 사용 가능

[Rescale 플랫폼 활용 Tip (코어타입 설정)]

앞서 플랫폼 소개의 내용과 같이 Rescale 플랫폼에서는 워크로드 특성에 맞추어 필요한 연산 성능을 제공할 수 있도록 다양한 코어타입들이 있으므로 사용자는 어플리케이션 및 시뮬레이션 모델의 크기에 따라 적절한 코어타입을 선택하는 것이 필요하다.

STAR-CCM+의 공식 Benchmark 모델인 LeMans 104M Cells 모델로 테스트한 결과를 예시로 코어 타입 설정에 대하여 설명하면 다음과 같다.



일반적으로 모델의 크기가 커질수록 계산 시 더 많은 코어 수의 활용이 불가피한데, 코어 수(노드 수)가 증가함에 따라 계산 노드간 데이터 전송이 많아져 전체 시뮬레이션 성능이 저하될 수 있어 이에 따른 영향도를 최소화하기 위하여 노드간 네트워크 속도가 빠른(100 Gbps 이상) 코어타입의 선택이 필요하다.

위 그림에서의 결과와 같이 Interconnect가 10 Gbps인 Ferrite와 Onyx같은 경우 약 400 코어 부근부터 코어 수 증가 대비 시뮬레이션 속도 향상의 폭이 크게 저하되는 것에 비하여 상대적으로 Interconnect가 빠른 Emerald, Luna(25 Gbps) 그리고 Carbon(100 Gbps)의 경우 더 많은 코어 수에서도 코어 수가 증가함에 따라 시뮬레이션 속도 또한 거의 일정한 비율로 증가하게 된다. Ferrite

와 Carbon의 CPU는 Intel Xeon Platinum 8268(Skylake)@2.7GHz CPU로 동일하나 Interconnect의 차이로 전체적인 계산 성능에서 차이가 나타난다.

위의 경우와 같이 Rescale의 코어타입들은 같은 CPU를 사용하는 경우에도 가상머신의 구성에 따라 노드당 코어 수, 메모리 용량, 저장장치 용량, 노드간 네트워크 속도 등이 다르므로 실제 시뮬레이션 업무에서 사용하는 모델을 활용하여 코어타입별로 성능 평가가 또한 필요한데 소프트웨어, 시뮬레이션 워크로드마다 병렬 처리 효율이 달라지기 때문이다.

Jasper		AMD EPYC 7742(Rome) @ 2.4 GHz / 60 Cores per Node 480 GB Memory per Node / 480 GB Storage(SSD) per Node 200 Gbps Interconnect								
NumberOfWorkers	InitialElapsedTimeSeconds	AverageElapsedTimeSeconds	MinimumElapsedTimeSeconds	MaximumElapsedTimeSeconds	SpeedUp	VirtualMemoryKBytes	ResidentMemoryKBytes	Parallel Efficiency	Time to solutionseconds	
60	14.248	3.062	2.981005556	3.194590586	60	155875680	62765900	100.00%	79.699	
120	15.407	1.891	1.729859538	2.2213883	97.141	294745028	86872416	80.95%	57.772	
240	17.792	1.135	0.946535879	1.353168294	161.886	657356252	142963052	67.45%	44.875	
480	11.303	0.922	0.674876316	1.315353053	199.281	1569826556	186780184	41.52%	36.353	

Peridot		AMD EPYC 7742(Rome) @ 2.4 GHz / 44 Cores per Node 480 GB Memory per Node / 480 GB Storage(SSD) per Node 200 Gbps Interconnect								
NumberOfWorkers	InitialElapsedTimeSeconds	AverageElapsedTimeSeconds	MinimumElapsedTimeSeconds	MaximumElapsedTimeSeconds	SpeedUp	VirtualMemoryKBytes	ResidentMemoryKBytes	Parallel Efficiency	Time to solutionseconds	
44	16.821	3.969	3.829304881	4.265621348	44	116862364	53357736	100.00%	101.395	
88	14.207	2.101	2.02748073	2.2513328	83.108	211919416	75989876	94.44%	58.846	
176	16.28	1.415	1.296083605	1.6376461	123.439	420725988	111853288	70.14%	49.094	
352	9.671	0.838	0.702720992	1.033877107	208.491	907416712	145125804	59.23%	30.631	
484	11.319	0.814	0.621294479	1.227857804	214.656	1374532404	186544736	44.35%	32.742	

추가적으로 시뮬레이션에 사용할 코어 수를 설정할 때 병렬 처리 효율과 클라우드의 과금 체계를 동시에 고려하는 것이 필요한데 가상의 시나리오를 예시로 설명하면 다음과 같다.

- 코어 수의 증가에 따른 시뮬레이션 계산 속도 향상의 폭이 1:1에 가까울 경우 2배의 코어 수로 계산하면 시뮬레이션은 2배 빠르게 완료되나 비용을 차이가 없음
- 실제로는 Interconnect 속도가 200 Gbps 정도로 높더라도 코어 수(노드 수) 증가 시 1:1로 계산 속도가 향상되지 않으므로 비용 대비 성능이 가장 잘 나오는 값을 찾는 것이 필요

[맺음말]

Rescale에서는 사용자들이 On-premise 환경에서 HPC 클라우드 환경으로 변화 시 Soft landing을 위하여 성능 평가 결과에 기반한 코어타입 추천, 시뮬레이션 워크플로우 효율성 향상을 위한 API 자동화, 기존 On-premise와의 하이브리드 구축 등 다양한 방법에 대한 가이드를 드리고 있으므로 도움이 필요하시면 info.korea@rescale.com으로 문의주시기 바랍니다.