The Future of HPC Systems: Towards Large-Scale Time-Slice Execution and Adaptive Accelerator Allocation

Eiji Yoshida

Computing Laboratory Fujitsu Ltd.



© 2024 Fujitsu Limited



• Better interactivity for new applications and users

- Instant execution is difficult on conventional HPC systems
- New HPC users are not familiar with batch queue systems
 - Users access systems through interactive terminals like Jupyter Notebook

• Maximize the <u>efficiency</u> of resource usage

- Accelerators are valuable resources
 - Supply shortage of GPUs
 - Increased cost of accelerator devices

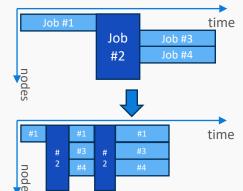
AI Computing Broker - Adaptive accelerator allocation

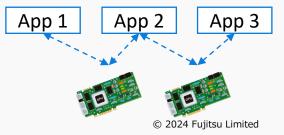
- Dynamic resource allocating mechanism for GPU applications
- → Application-aware efficient GPU usage within multiple jobs

Our Technologies for new Requirements

• Scalable Fine-grained gang scheduling

- Scalable time-sliced parallel computing job execution
- Providing better interactivity on HPC systems
- → Improving job execution efficiency





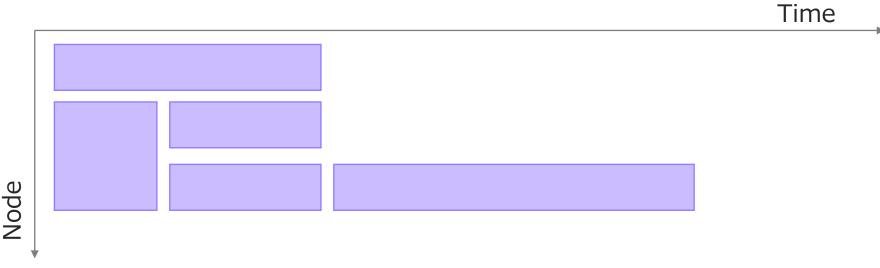


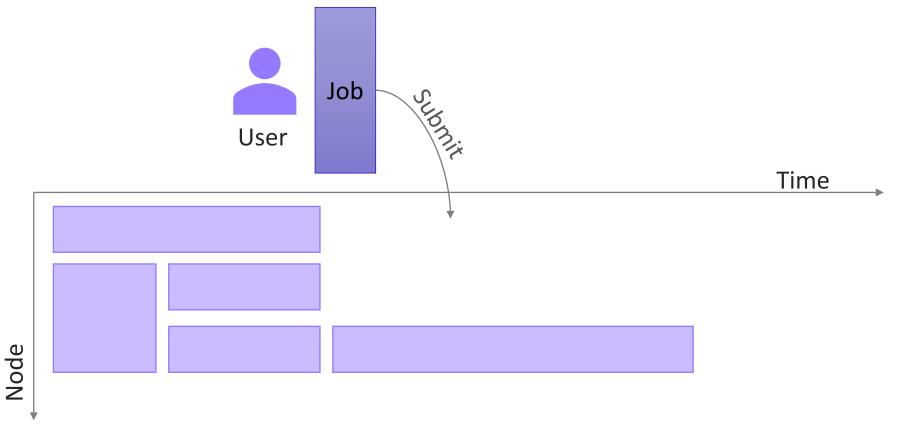


Scalable Fine-grained gang scheduling

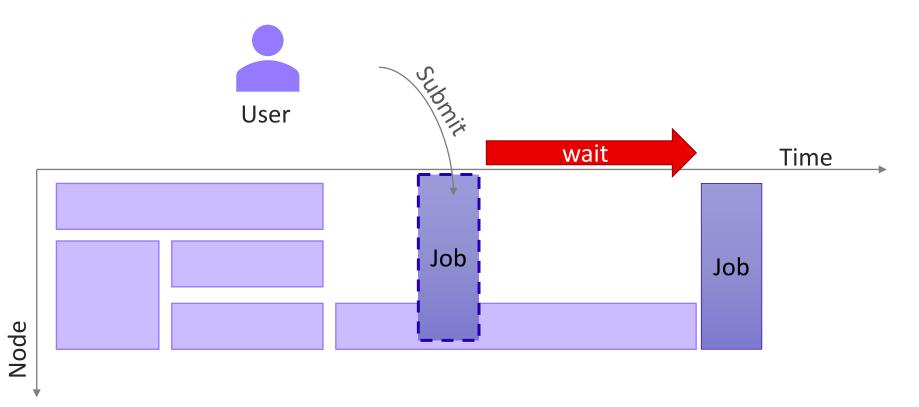
Long waiting time is required even for short jobs

• The limitation of batch queueing systems

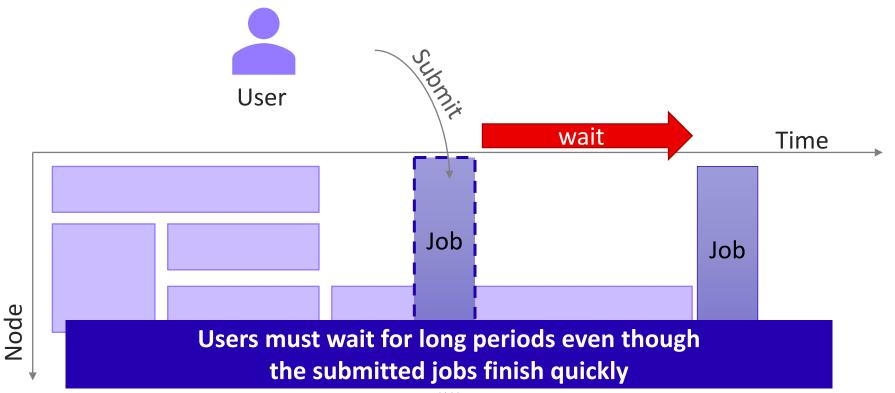




FUJITSU



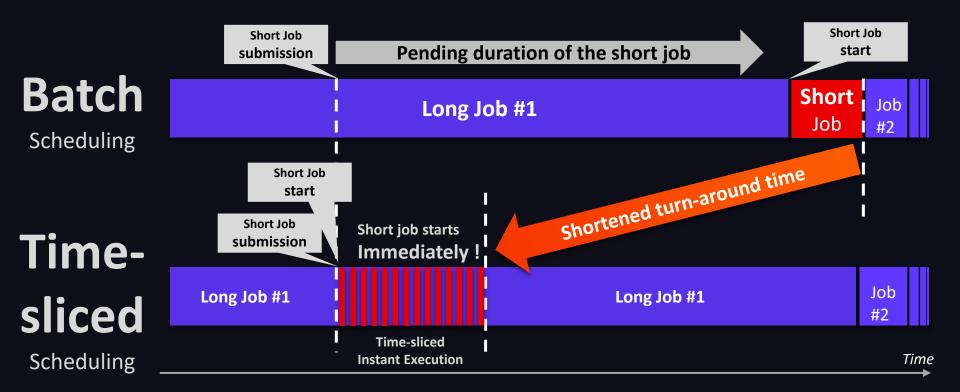
FUJITSU



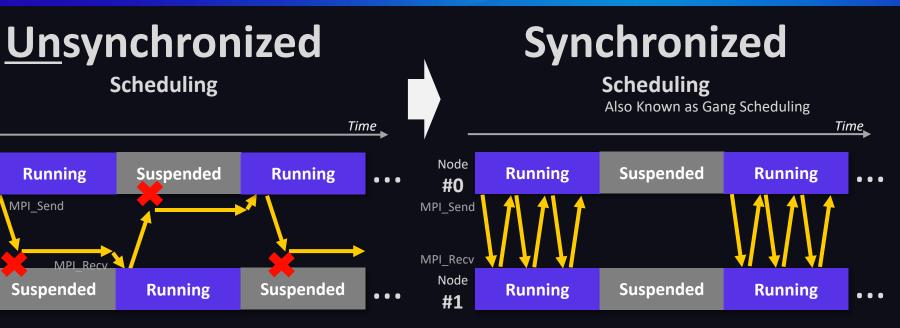
UTSU

Transition from Batch Scheduling





Requirements of Parallel Applications



Poor Performance

Node

#0

Node

#1

Running

Suspended

MPI Send

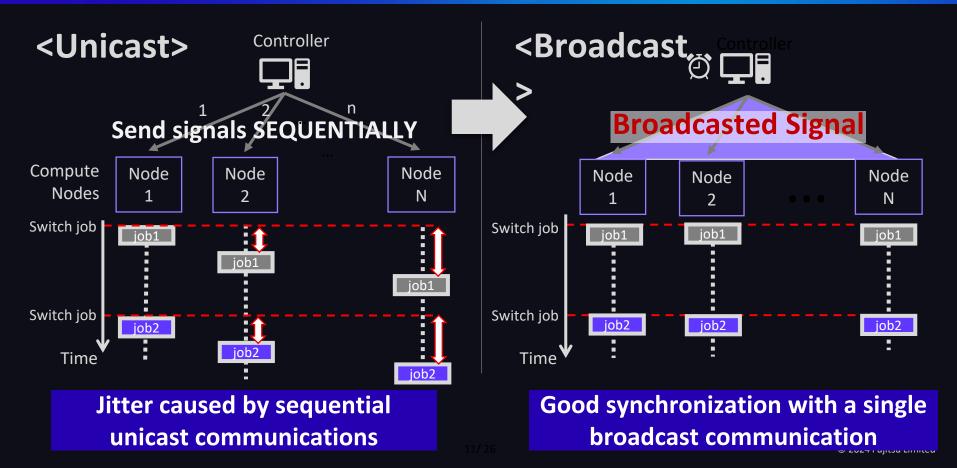
Only one message can be sent per time-slice (the worst case)

The Same Performance as usual

Multiple messages can be sent in a single time-slice

FUITSU

Scalable Synchronization with Broadcast Control Signals



Batch scheduling

 \leftarrow File 8

							Ъ										19						5	D
	× 🗵	Interactiv	re	×	+	~	-)	×	0	Home			×	a 1	nteracti	ve	>	<	+		\sim	-
① 127	7.0.0.1:88	88/noteb	ooks/Inte	eractive.i	pynb	6 1	*			:	\leftarrow	\rightarrow	C	i) 12	7.0.0. <mark>1</mark>	:8888	8/notek	books/I	nteract	ive.ip	ynb	Ê	☆	*
v Run	Kernel	Setting	6 Help						Trust	ted	File	Edit	View	Run	Kerr	nel	Setting	s Hel	р					
Ċ	•	C 🕨	Code	~	Jupyt	erLab 🖸	Pytho	n 3 <mark>(</mark> ipy	ykernel))		+ %	(Ĉ) I		44 5	Code		/	Jupy	terLab	3	Pyth
											I]:	I											
				(j)																				
T																							ාබ්	Ĵ
	Ų	ISES	a p	බ්රිඩ	1121	Clu	Iste	er e	is [its 1	ba 16	ck	en	් ල)[f@	CE	<u>2</u> 55	sing) s <u>j</u>	ys	îei	n.	©	2024
	① 127 v Run 1 ①	× ① 127.0.0.1:88 W Run Kernel ① ト ● This	× Interactive ↑ 127.0.0.1:8888/noteb ↑ Run Kernel Settings ↑ • • C • •	× Interactive ① 127.0.0.1:88888/notebooks/Inte W Run Kernel Settings Help ① ► ■ C ► Code This is a de	× Interactive × O 127.0.0.1:8888/notebooks/Interactive.in Run Kernel Settings Help C ► Code ~ Code ~ This is a demo	× Interactive × + O 127.0.0.1:8888/notebooks/Interactive.ipynb Run Kernel Settings Help C → Code → Jupyt C → Code → Jupyt	x Interactive x + 127.0.0.1:8888/notebooks/Interactive.ipynb P Run Kernel Settings Help C P Code v JupyterLab C	Interactive Iz7.0.0.1:8888/notebooks/Interactive.ipynb Run Kernel Settings Help Or >> Code JupyterLab Pythol This is a demonstration	x Interactive x + O 127.0.0.1:8888/notebooks/Interactive.ipynb Run Kernel Settings Help P = C >> Code JupyterLab C Python 3 (ip) P = C >> Code JupyterLab C Python 3 (ip) This is a demonstration of	Interactive x +	x Interactive x + · · · · × ① 127.0.1:8888/notebooks/Interactive.ipynb 2 * • • • • · · × W Run Kernel Settings Help Trusted I • • C • Code · JupyterLab [? Python 3 (ipykernel) • This is a demonstration of an in	Interactive x +	x Interactive 0 127.0.0.1:8888/notebooks/Interactive.ipynb x x Run Kernel Settings Home Trusted File Edit Image: Settings Home This is a demonstration of an interaction	x Interactive () 127.0.0.1:8888/notebooks/Interactive.ipynb () <t< td=""><td>× ▲ Interactive × + Image: Interactive ipynb P + + Image: Image</td><td>x Interactive x + Y - C X O 127.0.0.1:8888/notebooks/Interactive.ipynb</td><td>x Interactive x + - × + <td< td=""><td>X Interactive X + Y - C X O 127.0.0.1:8888/notebooks/Interactive.ipynb</td><td>× < Interactive</td> × + · · · × () 127.0.0.1:8888/notebooks/Interactive.jpynb () () 127.0.0.1:8888/notebooks/Interactive.jpynb () () () <</td<></td><td>× ▼ Interactive × + × - × I 127.0.0.1:8888/notebooks/Interactive.ipynb ☆ ★ □ ▲ : K Run Kernel Settings Help Trusted File Edit View Run Kernel Settings Help I + ☆ ○ ○ 127.0.0.1:8888/notebooks/Interactive File Edit View Run Kernel Settings Help I + ☆ ○ ○ + Code × Loode × I upyterLab □ Python 3 (ipykerne) ● I + ☆ ○ □ > □ C → Code × I is I This is a demonstration of an interactive Jupyter Note</td><td>x Interactive x + Y - Y X Home X Interactive X () 127.0.0.1:8888/notebooks/Interactive.jpynb (2) x * 1 : (2) C (0) 127.0.0.1:8888/notebooks/Interactive.jpy Run Kernel Settings Help Trusted File Edit View Run Kernel Settings Help () + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy File Edit View Run Kernel Settings Help () + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy File Edit View Run Kernel Settings Help () + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy File Edit View Run Kernel Settings Help () + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy () + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy File Edit View Run Kernel Settings Help () + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy () + K (0) + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy () + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy () + K (0) + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy () + K (0) + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy () + K (0) + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy () + K (0) + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy () + K (0) + K (0) + C (0) 127.0.0.1:8888/notebooks/Interactive.jpy () + K (0) + K (0</td><td>× Interactive × + · · × + <t< td=""><td>× Interactive × + · <t< td=""><td>x Interactive x + Y - □ X 0 127.0.0.1.8888/notebooks/Interactive.jpynb 12 x * □ • • C ⊙ 127.0.0.1.8888/notebooks/Interactive.jpynb 12 x v Run Kernel Settings Help Trusted File Edit View Run Kernel Settings Help ■ + X □ • • C → Code × JupyterLab □ f []: I This is a demonstration of an interactive Jupyter Notebook that uses a parallel cluster as its backend processing system.</td></t<></td></t<></td></t<>	× ▲ Interactive × + Image: Interactive ipynb P + + Image: Image	x Interactive x + Y - C X O 127.0.0.1:8888/notebooks/Interactive.ipynb	x Interactive x + - × + <td< td=""><td>X Interactive X + Y - C X O 127.0.0.1:8888/notebooks/Interactive.ipynb</td><td>× < Interactive</td> × + · · · × () 127.0.0.1:8888/notebooks/Interactive.jpynb () () 127.0.0.1:8888/notebooks/Interactive.jpynb () () () <</td<>	X Interactive X + Y - C X O 127.0.0.1:8888/notebooks/Interactive.ipynb	× < Interactive	× ▼ Interactive × + × - × I 127.0.0.1:8888/notebooks/Interactive.ipynb ☆ ★ □ ▲ : K Run Kernel Settings Help Trusted File Edit View Run Kernel Settings Help I + ☆ ○ ○ 127.0.0.1:8888/notebooks/Interactive File Edit View Run Kernel Settings Help I + ☆ ○ ○ + Code × Loode × I upyterLab □ Python 3 (ipykerne) ● I + ☆ ○ □ > □ C → Code × I is I This is a demonstration of an interactive Jupyter Note	x Interactive x + Y - Y X Home X Interactive X () 127.0.0.1:8888/notebooks/Interactive.jpynb (2) x * 1 : (2) C (0) 127.0.0.1:8888/notebooks/Interactive.jpy Run Kernel Settings Help Trusted File Edit View Run Kernel Settings Help () + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy File Edit View Run Kernel Settings Help () + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy File Edit View Run Kernel Settings Help () + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy File Edit View Run Kernel Settings Help () + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy () + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy File Edit View Run Kernel Settings Help () + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy () + K (0) + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy () + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy () + K (0) + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy () + K (0) + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy () + K (0) + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy () + K (0) + C > C (0) 127.0.0.1:8888/notebooks/Interactive.jpy () + K (0) + K (0) + C (0) 127.0.0.1:8888/notebooks/Interactive.jpy () + K (0) + K (0	× Interactive × + · · × + <t< td=""><td>× Interactive × + · <t< td=""><td>x Interactive x + Y - □ X 0 127.0.0.1.8888/notebooks/Interactive.jpynb 12 x * □ • • C ⊙ 127.0.0.1.8888/notebooks/Interactive.jpynb 12 x v Run Kernel Settings Help Trusted File Edit View Run Kernel Settings Help ■ + X □ • • C → Code × JupyterLab □ f []: I This is a demonstration of an interactive Jupyter Notebook that uses a parallel cluster as its backend processing system.</td></t<></td></t<>	× Interactive × + · <t< td=""><td>x Interactive x + Y - □ X 0 127.0.0.1.8888/notebooks/Interactive.jpynb 12 x * □ • • C ⊙ 127.0.0.1.8888/notebooks/Interactive.jpynb 12 x v Run Kernel Settings Help Trusted File Edit View Run Kernel Settings Help ■ + X □ • • C → Code × JupyterLab □ f []: I This is a demonstration of an interactive Jupyter Notebook that uses a parallel cluster as its backend processing system.</td></t<>	x Interactive x + Y - □ X 0 127.0.0.1.8888/notebooks/Interactive.jpynb 12 x * □ • • C ⊙ 127.0.0.1.8888/notebooks/Interactive.jpynb 12 x v Run Kernel Settings Help Trusted File Edit View Run Kernel Settings Help ■ + X □ • • C → Code × JupyterLab □ f []: I This is a demonstration of an interactive Jupyter Notebook that uses a parallel cluster as its backend processing system.

Gang scheduling

$\begin{array}{cccccccccccccccccccccccccccccccccccc$	C Home	× 🦲 Interactive	× +	\vee	-		×
B + ☆ □ □ ► ■ C ► Code ∨ JupyterLab □ Python 3 (ipykernel) ○ []:	\leftrightarrow \rightarrow C (i) 12	7.0.0.1:8888/notebooks/Int	eractive.ipynb		* 0		:
[]:	File Edit View Run	Kernel Settings Help				Trus	sted
	B + X 🗅 🗂	► Code	∽ Jupyt	erLab 🗋	Python 3 (ipykerne	el) (

Batch scheduling	Gang scheduling
C Home × K Interactive × + ✓ - □ ×	C Home × P Interactive × + · · · · ×
\leftrightarrow \rightarrow C (1) 127.0.0.1:8888/notebooks/Interactive.ipynb \bowtie \bigstar (1) \clubsuit (1) $𝔅$ (1	\leftrightarrow \rightarrow C (i) 127.0.0.1:8888/notebooks/Interactive.ipynb 🖄 🏠 🛱 🚨 😩 :
File Edit View Run Kernel Settings Help Trusted	File Edit View Run Kernel Settings Help Trusted
<pre> + % Python 3 (ipykernel) *</pre>	<pre></pre>

Preparing the parallel environment

Writing the paralles code to execute

Batch sch	nedul	ing
-----------	-------	-----

Dateri Scheduling	
C Home × K Interactive × + ✓ − □ ×	C Home
← → C 💿 127.0.0.1:8888/notebooks/Interactive.ipynb 🖻 🖈 🖬 😩 🗄	\leftrightarrow \rightarrow C (1) 12
File Edit View Run Kernel Settings Help Trusted	File Edit View Rur
🖻 + 🛠 🗇 🗳 🕨 🔳 C 🍽 Code 🗸 JupyterLab 🖓 Python 3 (ipykernel) 🌑	B + % D D
<pre>[*]: import ipyparallel as ipp rc = ipp.Client(profile='example-profile')</pre>	<pre>[4]: import ipyparallel rc = ipp.Client(pr</pre>
[*]:	[5]:
<pre>%%px from mpi4py import MPI import random import time comm = MPI.COMM_WORLD rank = comm.Get_rank() print(f"rank = {rank}")</pre>	<pre>%%px from mpi4py import import random import time comm = MPI.COMM_WW rank = comm.Get_ra print(f"rank = {ra</pre>
C 1:	%px: 100%
	<pre>[stdout:3] rank = [stdout:5] rank = [stdout:2] rank = [stdout:4] rank = [stdout:6] rank = [stdout:7] rank = [stdout:0] rank = [stdout:1] rank =</pre>
We can see the result immedia	te <mark>l</mark> y:on gan

Gang scheduling Interactive × × +

\leftarrow	\rightarrow G	① 12	7.0.0.1	1:8888/1	notebo	ooks/Inte	eractive.ip	bynb		*			:
File	e Edit V	iew Run	Ker	nel Se	ettings	Help						Trust	ed
٦	+ %	ē ů	► I	C	**	Code	~	Jupyte	erLab 🖸	Python	n 3 (ip	ykernel)	0
	4]: import ipy rc = ipp.(ple-p	rofile')						*
ſ	5]:												
	%%px from mpi4p import rar import tin comm = MPI rank = con print(f"ra	ndom ne .COMM_WO nm.Get_ra	DRLD ank()										0
%	рх: 100%							8/8 [00	:01<00:00	, 6.60tas	ks/s]		
	<pre>stdout:3] stdout:5] stdout:2] stdout:4] stdout:6] stdout:7]</pre>	rank = rank = rank = rank =	5 2 4 6										

ng scheduling system, but nothing appears on the batch scheduling system

X

File

im

			-0			0.0
Home ×	Interactive	× +	─	×	C Home	× 🥏 Interactiv
→ C ③ 127.0.0.1:8	8888/notebooks/Intera	active.ipynb	* * 🛛 😩) : <	- > C (0 ·	127.0.0.1:8888/notebo
Edit View Run Kerne	el Settings Help		Tr	rusted Fi	le Edit View Ru	un Kernel Settings
+ % 🗇 🖞 🕨 🔳	C ➡ Code	✓ JupyterLab	Python 3 (ipyker	nel) 🌑 🖬	+ % 🗅 🖞	▶ ■ C →
]:					[4]:	2
<pre>mport ipyparallel as ipp c = ipp.Client(profile='</pre>					<pre>import ipyparall rc = ipp.Client(</pre>	el as ipp profile='example-p
]:					[5]:	
<pre>%px rom mpi4py import MPI mport random mport time omm = MPI.COMM_WORLD ank = comm.Get_rank() rint(f"rank = {rank}")</pre>					<pre>%%px from mpi4py impo import random import time comm = MPI.COMM_ rank = comm.Get_ print(f"rank = {</pre>	WORLD rank()
]:				ç	%px: 100%	
•					<pre>[stdout:3] rank = [stdout:5] rank = [stdout:2] rank = [stdout:4] rank = [stdout:6] rank = [stdout:7] rank =</pre>	= 5 = 2 = 4 = 6 = 7

Gang scheduling

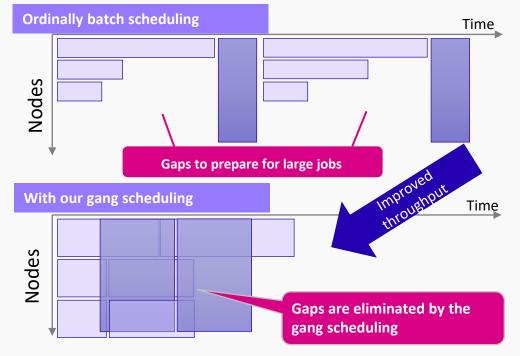
🔵 Home	× <i>e</i> Interactive	×	+	~		U	^
\leftrightarrow \rightarrow C (3) 127	7.0.0.1:8888/notebo	ooks/Interactive	e.ipynb ເ		* 0		:
File Edit View Run	Kernel Settings	Help				Tru	sted
a + % © ů	▶ ■ C >>	Code 🗸	Jupyter	Lab 🖸	Python 3	(ipykern	el) ()
<pre>[4]: import ipyparallel rc = ipp.Client(pro</pre>		rofile')					*
[5]:							
<pre>%%px from mpi4py import import random import time comm = MPI.COMM_WOF rank = comm.Get_ran print(f"rank = {ran</pre>	RLD nk()						c
%px:100% [stdout:3] rank = 3 [stdout:5] rank = 5 [stdout:2] rank = 2 [stdout:4] rank = 4 [stdout:6] rank = 6 [stdout:7] rank = 7			8/8 [00:01	1<00:00, (6.60tasks/	/s]	

Providing interactive parallel programming environment and running large-scale real-time applications immediately

Improved Interactivity and Throughput

• We have already deployed the mechanism on the actual system

- Successfully eliminated gaps caused by large jobs
- We confirmed
 - Shortened waiting time
 - Job throughput acceleration
 on our on-premise cluster





Al Computing Broker



AI Computing Broker



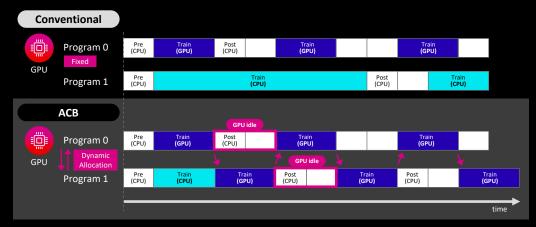
Realize mo	re efficient use of GPU in AI learning process
Challenges in using GPU	 Since the GPU is allocated on a per-program basis, even if the GPU idle time is large, other programs cannot use the GPU until the program ends. When virtualizing the GPU to share it among multiple programs, the GPU memory is divided for each program, reducing the capacity per program.

Features of AI Computing Broker

- GPU scheduling specialized for deep learning
- Control of GPU allocation at the processing unit level within the program
- Improves GPU utilization rate, reducing overall processing time
- More processes can be executed with the same GPU resources
- Unlike sharing of GPUs through virtualization technology, the program can fully utilize the GPUequipped memory

Technology of AI Computing Broker

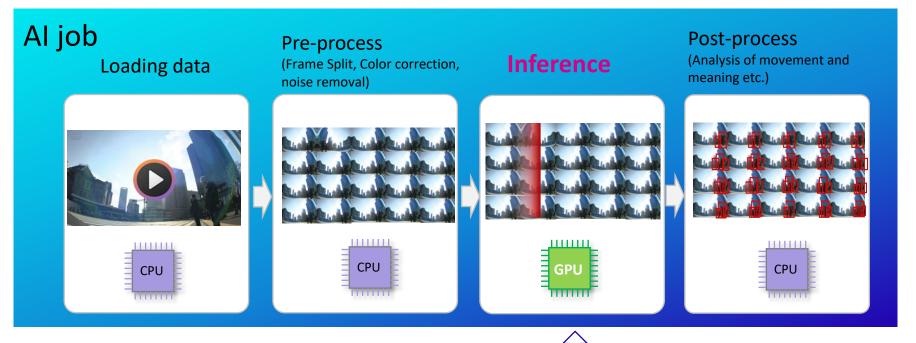
Analyze AI processing content, dynamically allocate GPU to processes that require GPU



One of the causes of low GPU utilization



AI jobs do not always require dedicated GPUs



ACB detects GPU utilization phases and dynamically allocates GPU(s)

2 GPU

1

GEFORCE RTX

\odot streamlit-m \cdot Streamlit \times	+		9	×
$\leftarrow \ \rightarrow \ \mathbf{G}$	0 D localhost:8502	☆	⊘ ± ≡	-
Conventional 2GPU	vhaFold2 Demo 🔞 Dashboards			
			Deploy	

2GPU

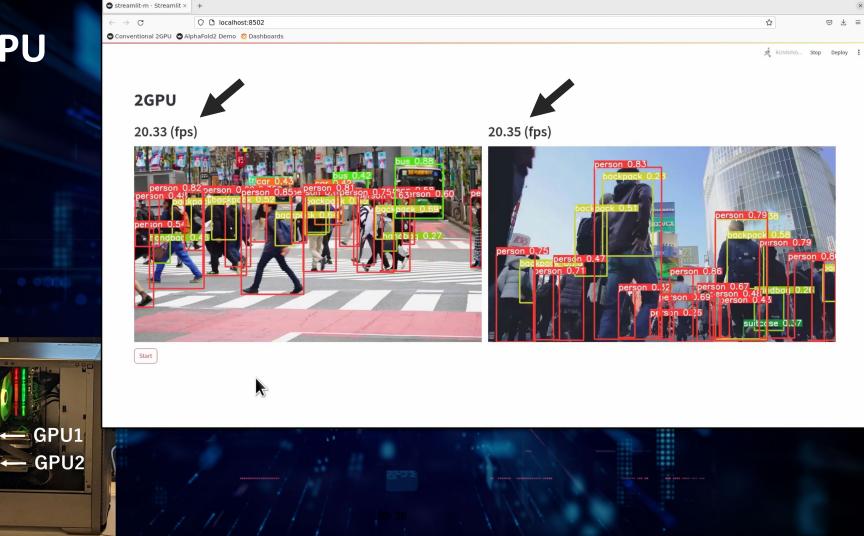




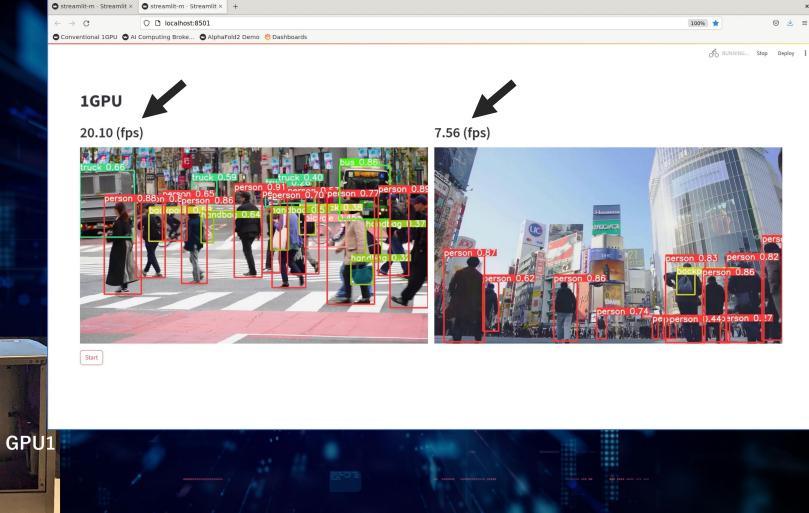


2 GPU

GEFORCE RTX



1 GPU (w/o ACB)



 Streamlit.m · Streamlit.x
 Streamlit.m · Streamlit.x
 +

 ← → C
 ○
 localhost:8502
 100%

 © Conventional IGPU
 AI computing Broke...
 O LaphaFold2 Demo © Dashboards

ACB

1 GPU





Start

GPU1



⊚ 坐 ≡

RUNNING... Stop Deploy

erson 0.90

😋 streamlit-m · Streamlit × 🛛 😋 streamlit-m · Streamlit × 🚽 ⊚ 👱 ≡ O D localhost:8502 100% **1** GPU Conventional 1GPU Computing Broke... Conventional 1GPU Constants RUNNING... Stop Deploy 1GPU + Alterptive GPU Allocator 20.17 (fps) 19.97 (fps) erson 0.90

ACB achieved 2-GPU performance using just 1 GPU



GPU1

Start

ACB

with ACB



Conclusion

Conclusion



 We developed two technologies to improve interactivity and efficiency on HPC systems

- Scalable Fine-grained gang scheduling
- Al Computing Broker Adaptive accelerator allocation

• We have already started deploying to real systems.

If you're interested, please reach out at Booth #H30.



Thank you

