



vLLM Hardware Plugin And Ascend Best Practice

Xiyuan Wang

Xiyuan Wang (@wangxiyuan)

- **vllm-project/vllm-ascend maintainer**
- **Senior Software Engineer at Huawei**
- **10 years of experience in open source software**

Agenda

- vLLM Plugin System
- vLLM Ascend Integration
- Future plan

vLLM Plugin System

Diversity in vLLM

- 100+ supported models
- 23+ supported quantization methods
- 8+ supported hardware
- 100+ supported custom ops
- 12+ supported attention backends

Diversity in vLLM

POSITIVES



- Vibrant community
- Newest technology implementation
- Satisfy all kind of user requirements

NEGATIVES



- Complex compatibility
- Difficult maintainability
- Countless issues and questions

vLLM plugin system



https://docs.vllm.ai/en/latest/design/plugin_system.html

vLLM generic plugin

```
# inside `setup.py` file
from setuptools import setup
```

```
setup(name='vllm_add_dummy_model',
      version='0.1',
      packages=['vllm_add_dummy_model'],
      entry_points={
          'vllm.general_plugins':
              ["register_dummy_model = vllm_add_dummy_model:register"]
      })
```

• Setup python entry_points in out of tree project

```
# inside `vllm_add_dummy_model.py` file
```

```
def register():
    from vllm import ModelRegistry

    if "MyLlava" not in ModelRegistry.get_supported_archs():
        ModelRegistry.register_model("MyLlava",
                                     "vllm_add_dummy_model.my_llava:MyLlava")
```

• Register a new model from plugin into vLLM

```
# inside `vllm_add_dummy_quantization.py` file
```

```
def register():
    from vllm.model_executor.layers.quantization import register_quantization_config
    from vllm.model_executor.layers.quantization.base_config import QuantizationConfig

    ...
    @register_quantization_config("new_quant_method")
    class AscendQuantConfig(QuantizationConfig):
        ...
```

• Or register a new kind of quantization into vLLM

Next Step?

Generic plugin ✓

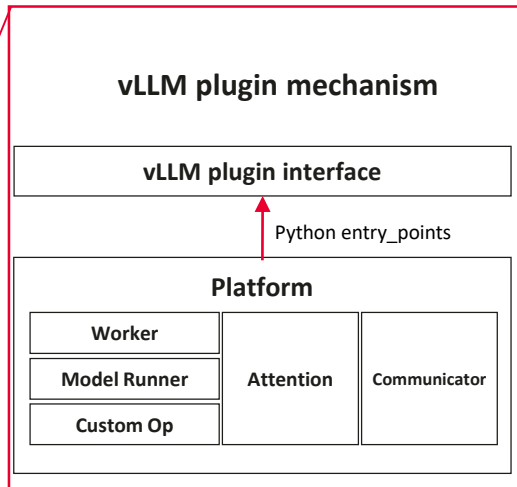
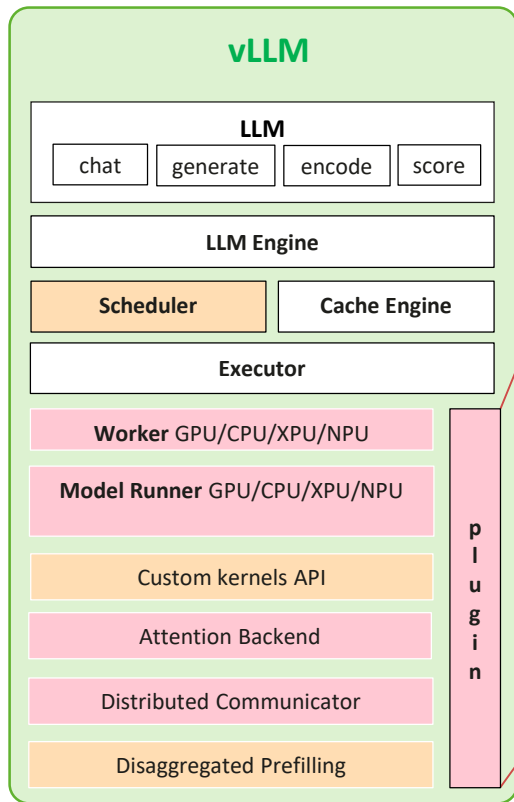
?



- 100+ supported models
- 23+ supported quantization methods
- 8+ supported hardware
- 100+ supported custom ops
- 12+ supported attention backends

vLLM platform plugin

Base on generic plugin, vLLM also supports platform plugin via Python entry_points from 0.7.1



```
# inside `setup.py` file
from setuptools import setup

setup(name='vllm_add_dummy_platform',
      version='0.1',
      packages=['vllm_add_dummy_platform'],
      entry_points={
          'vllm.platform_plugins': [
              "dummy_platform_plugin = vllm_add_dummy_platform:register"
          ]
      })

# inside `vllm_add_dummy_platform.py` file
def register():
    return "vllm_ascend.platform.NPUPlatform"
```

- The entry_points key is "vllm.platform_plugins" instead of "vllm.generic"
- The value is a function that return the new platform object.

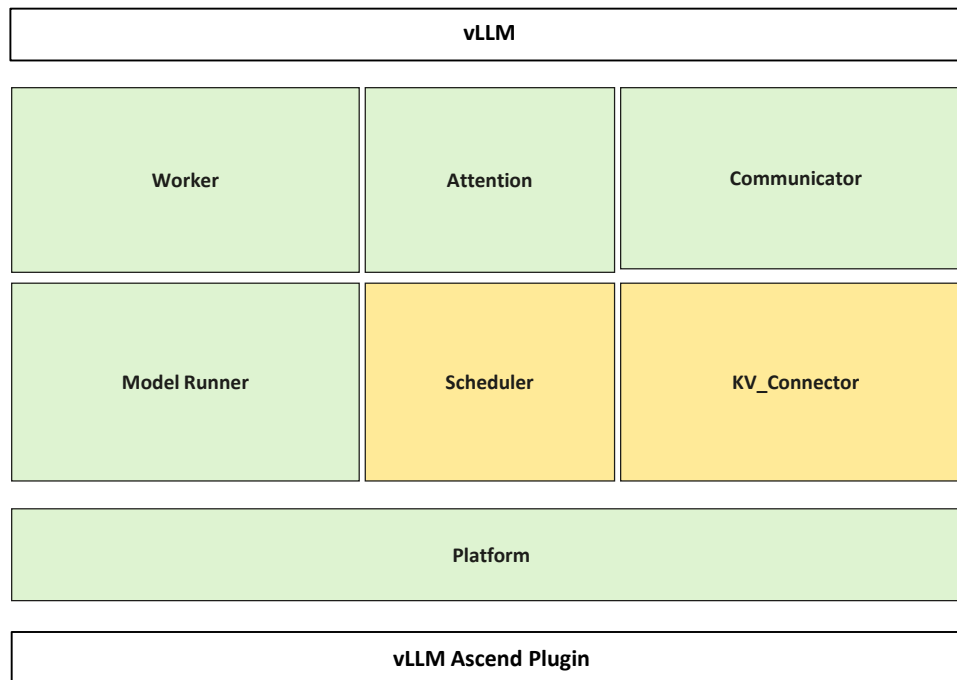
vLLM Ascend Integration

User experience

`pip install vllm vllm-ascend`

- vLLM community **official** project
- Install and enable Ascend with vLLM in **one command**
- **No change** for user code
- Run on Ascend **automatically**

Technical architecture



Platform register

in vLLM

1. Load platform from python entry_points

2. Call plugin function, set global var

in vLLM Ascend

1. Set python entry_points

2. Implement register function

```
from importlib.metadata import entry_points
discovered_plugins = entry_points(group='vllm.general_plugins')
for plugin in discovered_plugins:
    func = plugin.load()
    func()
```

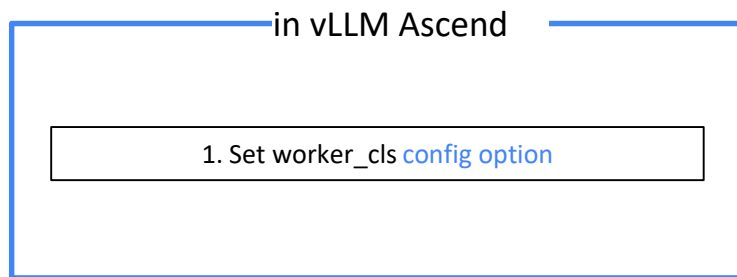
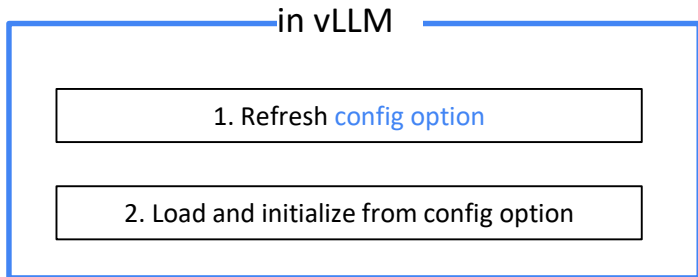
vLLM

```
from setuptools import setup
setup(entry_points={
    'vllm.general_plugins':
        ["register_dummy_model = vllm_add_dummy_model:register"]
})
```

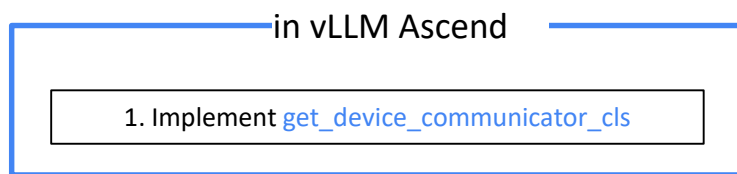
out-of-tree plugin

Python EntryPoint Mechanism

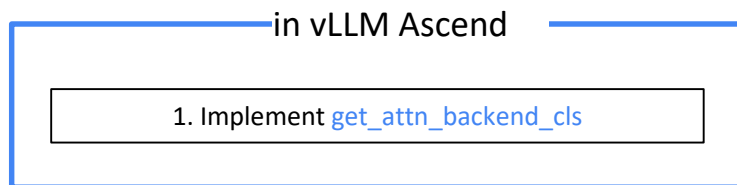
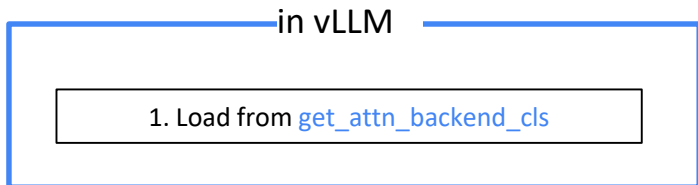
Worker register



Communicator register



Attention register



Pytorch Ops register

in vLLM

1. Call `forward_oop` function in each op

in vLLM Ascend

1. Implement `forward_oop` for each op

Custom Ops register

in vLLM

1. Build ops from c++/cu

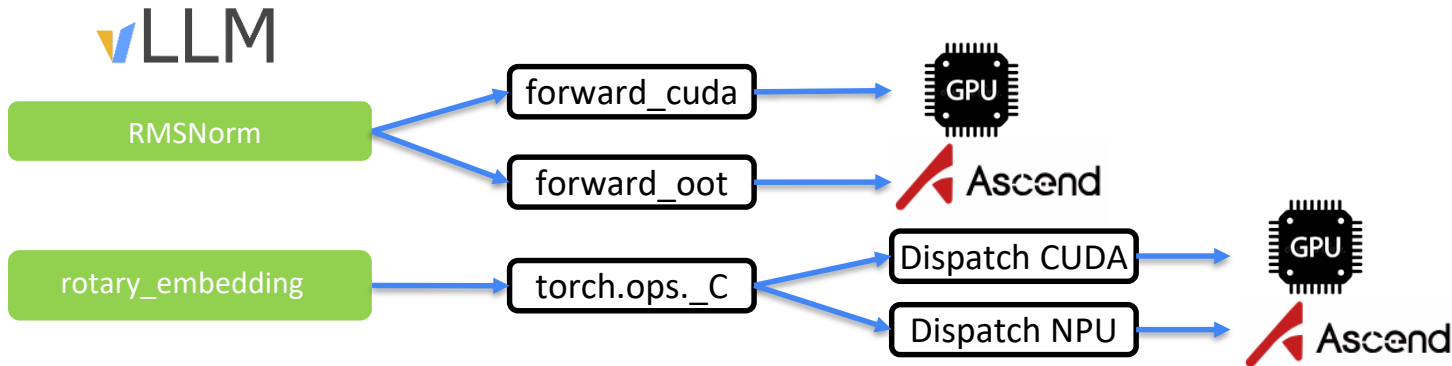
2. Load ops to torch namespace

3. Call ops from torch namespace

in vLLM Ascend

1. Implement ops from c++ with different dispatch key

2. Bind to the same namespace



vLLM Ascend Support matrix



Feature Support

Feature	Supported	Note
Chunked Prefill		Plan in 2025 Q1
Automatic Prefix Caching		Plan in 2025 Q1
LoRA	X	Plan in 2025 Q1
Prompt adapter	X	Plan in 2025 Q1
Speculative decoding	✓	
Pooling	✓	The accuracy is not correct, it'll be fixed in 2025 Q2
Enc-dec	X	Plan in 2025 Q2
Multi Modality	✓ (LLaVA/Qwen2-vl/Qwen2-audio/internVL)	Add more model support in 2025 Q2
LogProbs	✓	
Prompt logProbs	✓	
Async output	✓	
Multi step scheduler	✓	
Best of	✓	
Beam search	✓	
Guided Decoding	✓	Find more details at the issue
Tensor Parallel	✓	
Pipeline Parallel	✓	

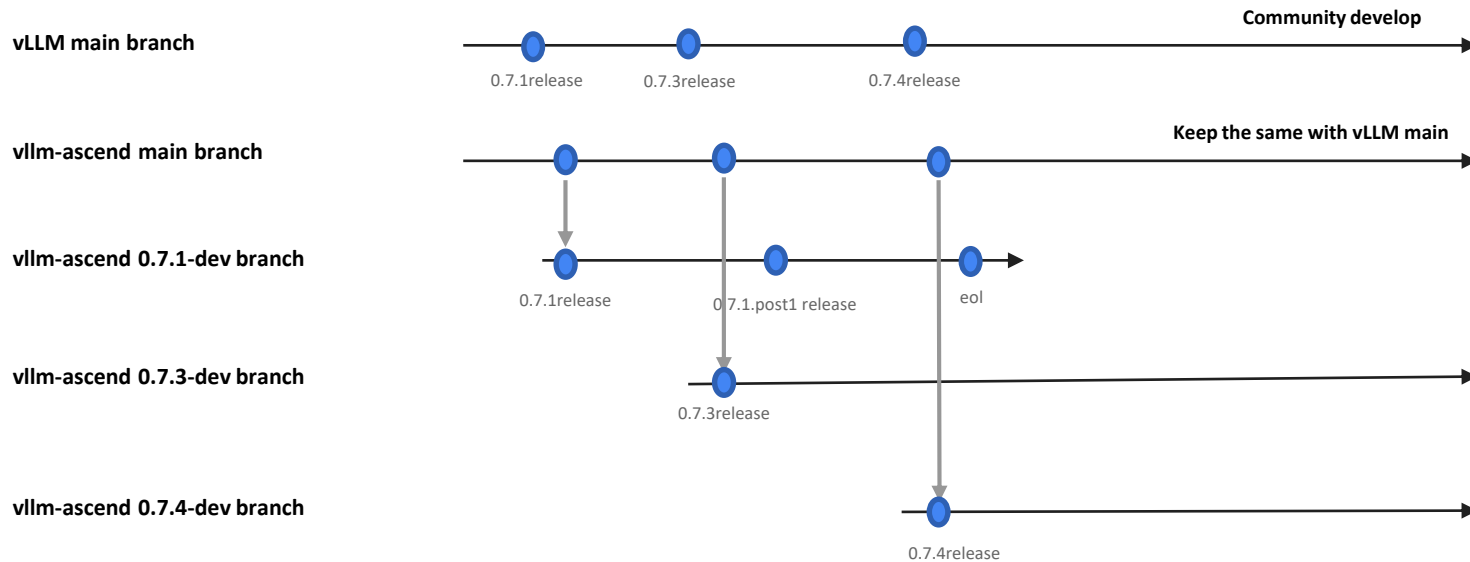
Supported Models

Model	Supported	Note
DeepSeek v3	✓	
DeepSeek R1	✓	
DeepSeek Distill (Qwen/LLama)	✓	
Qwen2-VL	✓	
Qwen2-Audio	✓	
Qwen2.5	✓	
Qwen2.5-VL	✓	
MiniCPM	✓	
LLama3.1/3.2	✓	
Mistral		Need test
DeepSeek v2.5		Need test
Gemma-2		Need test
Baichuan		Need test
Internlm	✓	
ChatGLM	✗	Plan in Q2
InternVL2.5	✓	
GLM-4v		Need test
Molomo	✓	

<https://vllm-ascend.readthedocs.io/en/latest/>

Release Policy

evolving in sync with the vLLM community



vllm-ascend	vLLM	Python	Stable CANN	PyTorch/torch_npu
v0.7.3rc1	v0.7.3	3.9 - 3.12	8.0.0	2.5.1 / 2.5.1.dev20250308
v0.7.1rc1	v0.7.1	3.9 - 3.12	8.0.0	2.5.1 / 2.5.1.dev20250218

Future plan

Feature Plan

- **Feature complete(Chunked prefill etc)**
- **V1 Engine support**
- **Scheduler Plugin**
- **Prefilling Disaggregated Plugin**
- **Model support in Day1**

Performance Plan

- **Custom Ops support and implementation**
- **TP, PP, EP and DP improvement**
- **Benchmark per PR and per day**

Quality Plan

- **UT coverage**
- **Model daily Test CI**
- **rc release and post release strategy**

vLLM Ascend v0.7.3rc1 release



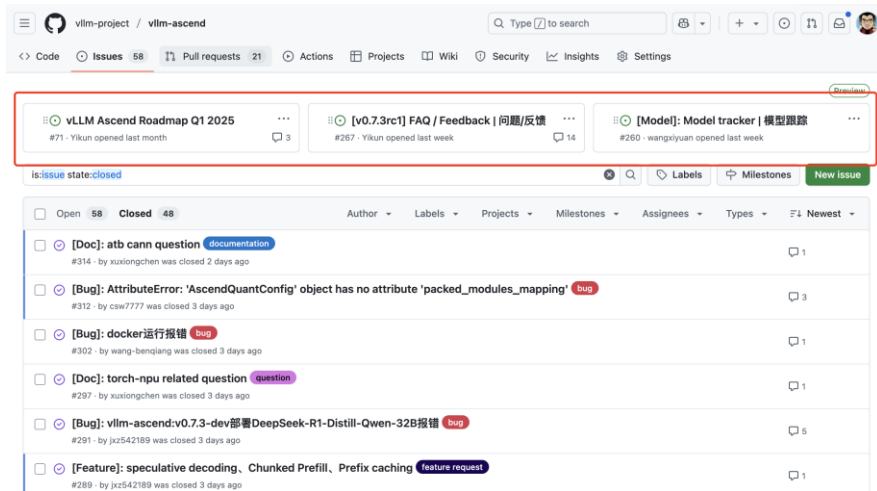
vLLM Ascend First RC Release for vLLM v0.7.3

```
$ docker pull quay.io/ascend/vllm-ascend:v0.7.3rc1
```

```
$ pip install vllm vllm-ascend
```

Doc: <https://vllm-ascend.readthedocs.io>

Feedback: github.com/vllm-project/vllm-ascend/issues



We welcome and value any contributions and collaborations

- vLLM Team (rfc review, PR contributor and reviewer):

@akeshet, @DarkLight1337, @jeejeelee, @njhill, @mgoin @simon-mo, @yannicks1, @youkaichao

- vLLM Ascend Contributor/Reviewer (27):

@Angazenn @ApsarasX @ganyi1996ppo @ji-huazhong
@kunpengW-code @MengqingCao @new-TonyWang @noemotiovon @Potabk
@ranjiewen @shen-shanshan @shink @ShiyaNiu @Sidaoy @wangxiyuan @whx-sjtu
@wuhuikx @xiemingda-1002 @Yikun @zouyida2002 @Yaphets24 @simon-mo @yiz-liu
@mengwei805 @rjg-lyh @wwfu109

-- vLLM Ascend Issue Feedback (90+)

@a-flying-crow, @ahutkai, @AIR-hl, @baymax591, @caijijuhe, @caolicaoli,
@chenqi123, @ColdeZhang, @csw7777, @dawnranger, @dependabot[bot],
@fengzx99, @ffanyt, @flying632, @gameofdimension, @gebing, @geekchen007,
@GenerallyCovetous, @gyr-kdgc, @h7878778h, @huangwei-xy, @huowang-li,
@hzOne, @imsatoshi, @invokerbyxv, @Jial5588, @jiayi-1994, @Jozenn, @jrcyyzb,
@junming-yang, @jxz542189, @Kangzf1996, @liaoyanqing666, @Luo-Jinyan, @man-
in-sky, @maxupeng, @mhqmh, @micheleamarzollo, @myliangchengyu, @niejingwei,
@nk1888, @onehaitao, @phellonchen, @pjgao, @qsunyyy, @Qukka0914, @rickywu,
@RongRongStudio, @ryys1122, @shuowoshishui, @SHYuanBest, @staugust, @tcye,
@w1051868626, @wang-benqiang, @whu-dft, @wzb1005, @Xinteny, @xinyang920,
@xuxiongchen, @XuyaoWang, @yimuu, @YuanEZHou, @YuanJZhang, @zhuo97,
@Ziang-Zack-Gao, @ZRJ026

Welcome to Contribute

- **Submit issue/Answer question**
- **Fix bug/Add feature/Write Doc**
- **Implement Ops/Benchmark/Accuracy improvement**
- **.....**

https://vllm-ascend.readthedocs.io/en/latest/developer_guide/contributing.html



Building the **fastest** and **easiest-to-use**
open-source LLM inference & serving
engine!



<https://github.com/vllm-project/vllm>
<https://github.com/vllm-project/vllm-ascend>



<https://slack.vllm.ai> sig-ascend channel



<https://www.linkedin.com/company/vllm-project>



https://twitter.com/vllm_project



<https://opencollective.com/vllm>



Wechat Group