

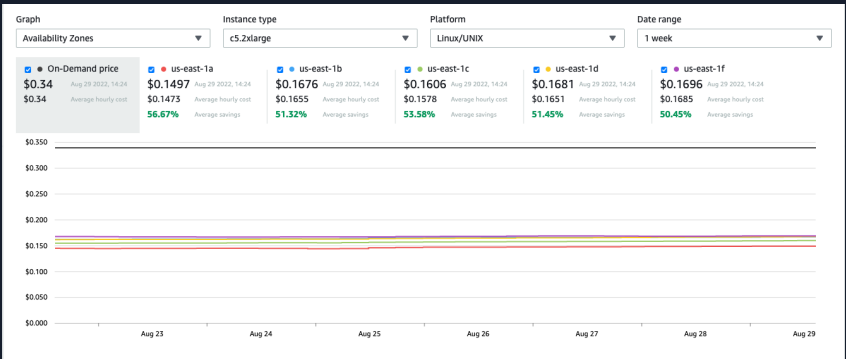
# GPU Spot Instance

解决成本问题

Spot 实例是**空闲**EC2 容量，可作为购买选项提供给客户，并享受大幅折扣



**空闲 EC2 容量**  
与按需部署相同的基础架构



**高达 90% 的按需价格折扣**  
Spot 价格基于 EC2 实例的长期供需趋势 **(无竞价)**



**可中断**  
如果 EC2 特定容量池需要恢复容量，Spot 实例可能会中断，并在中断前2分钟发出警告

# CPU 和 GPU 的调度区别

## CPU 实例

- 容量相对充足
- 不同代间差别不明显
- 实例类型能明显反映资源数量
- 多个应用能共享同一个资源池，增加资源供给可调度更多应用，可将不同类型（CPU/内存密集型）工作负载混布

## GPU 实例

- 容量相对不足
- 不同代间差别非常明显（例：只有 g6e 的48G显存能放下24B模型）
- 实例类型无法反映资源数量（例：g5.12x 有4块GPU， g5.16x 只有1块GPU）
- 应用独占 GPU，无法与其他应用共享。除GPU外，其他资源供给不直接影响可调度的应用数

# Spot 中断最佳实践根本原则 – 灵活性

实例灵活



使用尽可能多且深的容量池

- 不同实例类型
- 不同实例大小
- 不同可用区

C4	AZ1	AZ2	AZ3	On-Demand
8XL	\$0.28	\$0.27	\$0.29	\$1.76
4XL	\$0.21	\$0.19	\$0.16	\$0.88
2XL	\$0.08	\$0.07	\$0.08	\$0.44
XL	\$0.04	\$0.05	\$0.04	\$0.22
L	\$0.01	\$0.01	\$0.02	\$0.11

地区灵活

时间灵活



时间灵活和/或价格灵活可以进一步降低中断率，提高应用程序正常运行的时间

价格灵活



# 查看 Spot 实例的中断率

Spot Instance Advisor

目前只在global区域 可用

Spot instance advisor 工具  
提供过去30 天内某个实例机型在  
某个region的中断率和折扣率的数据。  
(5%-20%)

区域：

美国西部（俄勒冈）

操作系统：

Linux/UNIX

实例类型筛选条件：

vCPU 数量下限：

4

最小内存 (GiB):

8

☐ EMR 支持的实例类型

实例类型	vCPU	内存 (GiB)	按需可节省额*	中断频次 ▾
r5n.4xlarge	16	128	63%	<5% □□□□□
d3en.2xlarge	8	32	70%	<5% □□□□□
r5n.2xlarge	8	64	59%	<5% □□□□□
i3en.12xlarge	48	384	70%	<5% □□□□□
r5n.16xlarge	64	512	73%	<5% □□□□□
m5zn.xlarge	4	16	77%	<5% □□□□□
c7g.4xlarge	16	32	51%	<5% □□□□□

# 判断所选的 Spot 资源池的健壮性

## Spot Placement Score

### 黄金经验法则

- 对于每种工作负载，灵活地在至少 10 种实例类型之间进行选择
- 确保所有可用区配置为在 VPC 中使用

Global区域和中国区均支持

### spot placement score (1-9分)

- 更准确的判断所选资源池当前的健壮性，分数越高代表获得所需资源的可能性越大
- 每个region或AZ的SPS是根据目标容量、实例类型的组成、历史和当前Spot使用趋势以及请求的时间来计算的。
- 正确参考SPS：
  - ✓ 使用跟SPS相同的配置
  - ✓ 使用 **capacity-optimized** 分配策略
  - ✓ 立即根据分数采取行动

SPS 仅可作为建议使用，不能提供在可用容量或中断风险方面的任何保证。

EC2 > Spot requests > Spot placement score

Spot placement score

Spot placement score helps you to select optimal Regions or Availability Zones to run workloads that can use multiple instance types.

Target capacity and instance type requirements

Edit

Target capacity	vCPUs	Memory (GiB)	CPU architecture	Additional attribute filters
500 vCPUs	4 minimum 12 maximum	No minimum No maximum	arm64	Instance virtualization type HVM

Placement scores

Calculate placement scores

We calculate placement scores based on factors such as the number and composition of the instance types, the target capacity, the Spot usage trends, and the time of the request. Scores serve as a guideline, and no score guarantees that your Spot request will be fully or partially fulfilled. A score of 10 means that your Spot capacity request is highly likely to succeed in that Region or Availability Zone at the time of the request. A score of 1 means that your Spot capacity request is not likely to succeed.

Regions to evaluate

Regions to score

Clear filters

☐ Provide placement scores per Availability Zone

Region	Placement score
Europe (Frankfurt) eu-central-1	9
Asia Pacific (Mumbai) ap-south-1	9
Asia Pacific (Tokyo) ap-northeast-1	9
Canada (Central) ca-central-1	9

e.g. 指定所用的实例属性，针对单个区域

# 追踪资源池变化趋势并调整

## Spot Placement Score

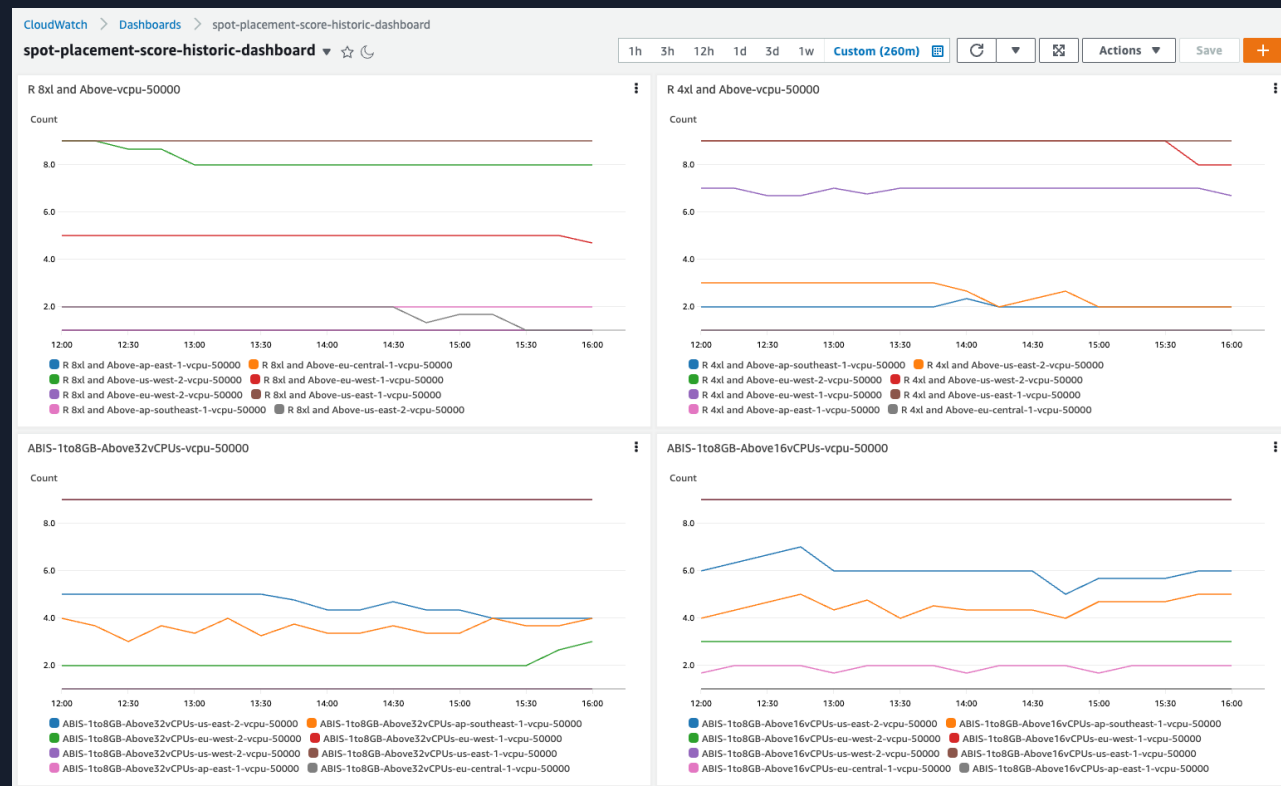
方法一：

定期调用API [GetSpotPlacementScores](#) 查询机型/AZ/Region组合的分数，把1-2等低分资源向8-9高分AZ/region主动切换

方法二：

通过github 项目部署 [EC2 Spot Placement Score Tracker](#)

- 此项目可自动捕获SPS，并将 SPS 指标存储在 CloudWatch 中。然后可以使用 CloudWatch 控制面板可视化历史指标。
- CloudWatch 还可用于触发警报和事件自动化，例如将工作负载移动到容量可用的区域



Spot Placement Score Tracker Dashboard

# 优选 Price-Capacity-Optimized 分配策略

该策略发布于2022年11月11日

示例来自[price-capacity-optimized](#) 发布博客

## Price-capacity-optimized 分配策略

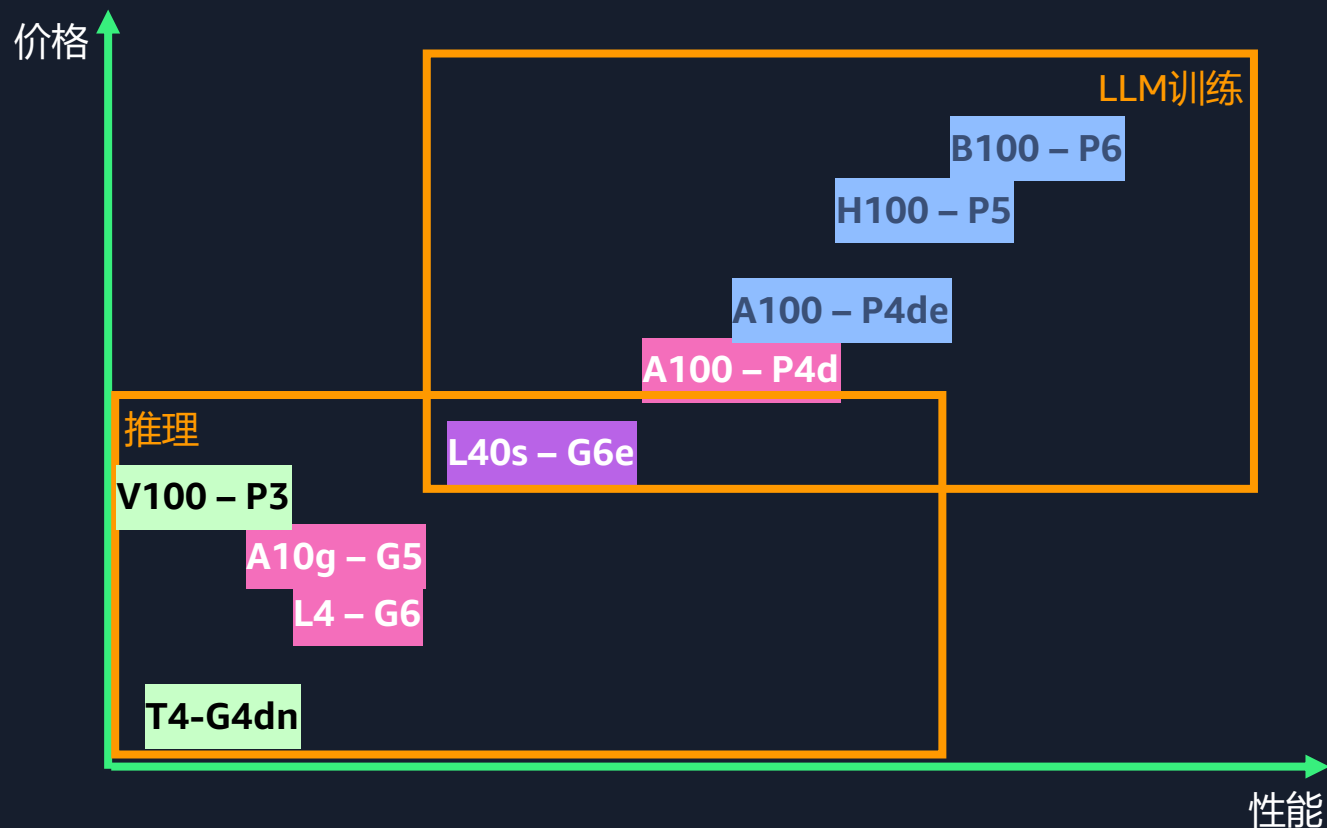
- 平衡Spot价格和容量
- 尝试在具有高容量可用性的多个低价池中启动Spot实例以实现多样化
- 推荐作为大多数Spot workloads的默认策略
  - 无状态和容错工作负载
  - 中断成本高的工作负载但是配置过checkpointing
  - 短时工作负载（最大程度降本的同时考虑容量可用性）
- 也是 Karpenter 上唯一支持的Spot 分配策略

Allocation strategy	Instance allocation	Cost of Auto Scaling group	Spot interruptions rate
price-capacity-optimized	40 c6i.xlarge 20 c5.xlarge	\$4.80/hour	3%
capacity-optimized	60 c5.xlarge	\$5.00/hour	2%
lowest-price	30 c5a.xlarge 30 m5n.xlarge	\$4.75/hour	20%

EC2 Fleet/Spot Fleet 支持的策略有

Price-capacity-optimized, Capacity Optimized, CapacityOptimizedPrioritized, Diversified, Lowest price

# Amazon EC2 加速计算实例概览



- 对于图像生成类工作负载，除SD 1.5可使用g4dn实例类型外，其他工作负载均建议使用g5, g6, g6e等实例类型。
- 性能对比：p4d > g6e >> g5 > g6
- 目前解决方案在海外区域默认使用g6和g5混合实例类型，在中国区域使用g5实例类型。
- 实例大小建议选择2xlarge及以上，以确保有足够空间加载模型到内存。



# Amazon EC2 g6e 实例

- G6e 实例配备了**多达 8 个 NVIDIA L40S Tensor Core GPU** (每个 GPU 内存为 **48 GB**) 和第三代 AMD EPYC 处理器。
- 支持最多 192 个 vCPU、最高 400Gbps 的网络带宽和最多 1.536 TB 的系统内存，以及最多 7.6 TB 的本地 NVMe SSD 存储。
- 与 G6 实例相比，G6e 提供了高出 2 倍的显存 (48 GB) 和 2.9 倍的显存带宽。G6e 实例的性能与 G5 实例相比最多可**提升 2.5 倍**。
- g6e 虽然较 p4d (A100 40G) 显存更大，但 g6e **不支持 NVLink** 技术，多卡之间数据传输需要通过操作系统内存，相对不适合多卡场景，但**对于单卡场景（如推理）工作更为出色**。



采用Ada Lovelace 架构  
(4090同款)



推理性能最高可达A100  
的1.2倍，A10的2.5倍



提供48GB超大显存，可  
承载大参数量模型



推理性价比最高为A10  
的1.5倍

# Amazon EC2 g6 实例

- G6 实例配备**多达 8 个 NVIDIA L4 Tensor Core GPU**（每个 GPU 内存为 **24 GB**）和第三代 AMD EPYC 处理器。
- 支持最多 192 个 vCPU、最高 100Gbps 的网络带宽和最多 768 GiB 的系统内存，以及最多 7.52 TB 的本地 NVMe SSD 存储。
- 与 g4dn 实例相比，g6 实例可提供高达 2.5 倍的深度学习推理性能，性价比提升1.3倍。与 g5 实例相比**性能相当，成本降低20%**



采用Ada Lovelace 架构  
(4090同款)



推理性能最高可达T4的  
2.5倍，与A10相当



提供24GB显存，可承载  
中等参数量模型



利用Spot实例可有效降  
低成本

# 推荐的GPU实例 – G6e

L40S

## L40s 相较于 A10

- 3倍左右的算力提升
- 2倍的GPU memory
- 更强的GPU memory bandwidth

GPU	A10g	A10	L40s
Instance Type	AWS G5	Aliyun gn7i-c8g1	AWS G6e
FP32	35 TF	31.2 TF	91.6 TF
TF32 Tensor Core	35 TF   70 TF	62.5 TF   125 TF*	183 366 TF*
BFLOAT16 Tensor Core	70 TF   140 TF*	125 TF   250 TF*	362.05 733 TF*
FP16 Tensor Core	70 TF   140 TF*	125 TF   250 TF*	362.05 733 TF*
INT8 Tensor Core	140 TOPS   280 TOPS*	250 TOPS   500 TOPS*	733 1466 TOPs*
Memory Bandwidth	600GB/s	600GB/s	864GB/s
GPU memory	24GB	24GB	48GB
RT Core	80	72	209 TF