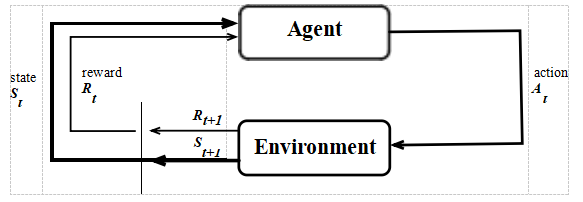
Тема: RL-Планировщик

RL-Планировщик:

Автоматизированный планировщик пакетных заданий высокопроизводительных вычислений, использующий обучение с подкреплением

Что такой RL:

Обучение с подкреплением (RL) - это группа методов машинного обучения, которые позволяют агентам самостоятельно учиться в интерактивной среде путем проб и ошибок .



На рис. 1 показана общая структура RL. На каждом шаге t агент наблюдает за соответствующим состоянием St и выполняет действие At. Следовательно, действие переведет состояние среды из St в St + 1, и агент получит вознаграждение Rt + 1.

На современных платформах высокопроизводительных вычислений (HPC) по-прежнему преобладают пакетные задания.

Теоретически планирование пакетных заданий несложно . На практике планировщики HPC принимают решения о планировании с помощью эвристических функций приоритета, которые присваивают каждому заданию приоритет на основе его атрибутов.

Однако независимо от того, как построена функция приоритета (например, с помощью тщательного анализа рабочей нагрузки или годового опыта экспертов), вышеупомянутые планировщики имеют один и тот же недостаток: они фиксированы и не могут автоматически адаптироваться к изменениям в целевом объекте. окружающая обстановка. На типичных платформах высокопроизводительных вычислений рабочие нагрузки могут меняться от месяца к месяцу, а цели оптимизации также могут меняться со временем.

Однако на практике перед использованием RL в планировании пакетных заданий HPC необходимо ответить на несколько ключевых вопросов:

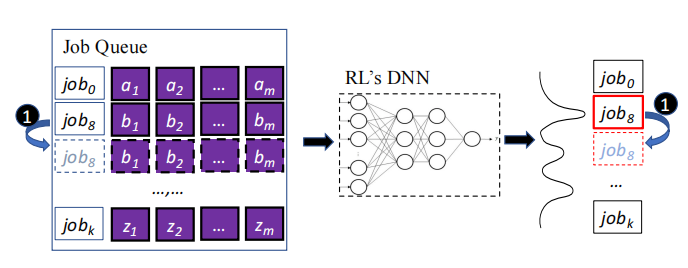
1. Может ли RL обеспечить высококачественную политику планирования, которая сравнима или даже лучше, чем точно настроенные современные политики планирования, для различных рабочих нагрузок и различных целей оптимизации?

2. Применяется ли политика планирования на основе RL только к учебной рабочей нагрузке или вообще применима к различным рабочим нагрузкам? Другими словами, будет ли политика на основе RL эффективно планировать задания для новых рабочих нагрузок, которые это никогда не видел раньше?

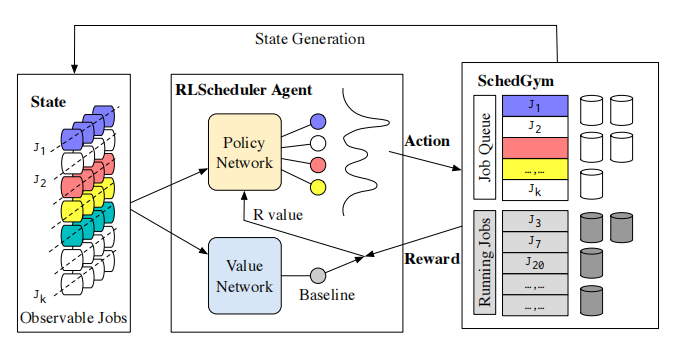
3. Какие ключевые факторы влияют на эффективность обучения планировщиков заданий на основе RL?

во-первых, при правильном проектировании RLScheduler способен изучать политику планирования высокого качества, которая сравнима с современными планировщиками или даже превосходит их, на различных (как синтетических, так и реальных условиях). ) рабочих нагрузок или с совершенно разными целями оптимизации. Во-вторых, модель, изученная RLScheduler, в целом хорошо работает даже с рабочими нагрузками, которые он никогда раньше не видел, что делает ее достаточно стабильной для использования на практике.

два ключевых фактора, которые влияют на производительность планировщика пакетных заданий на основе RL: 1) структура нейронной сети агента; и 2) дисперсия наборов обучающих данных.

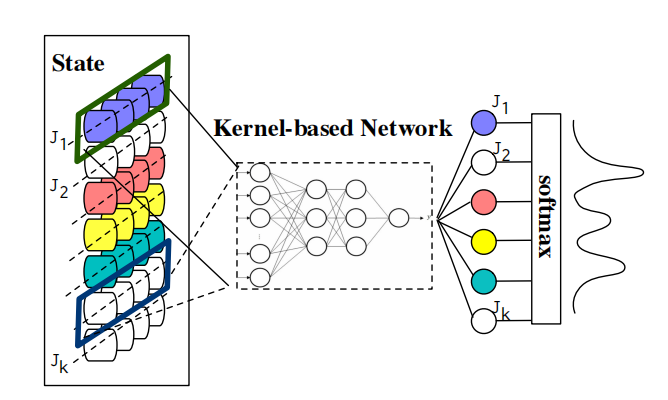


На рис. как агент RL на основе DNN принимает решения о планировании: он принимает ожидающие задания и их характеристики (например, a1 → m) в качестве входного вектора и выводит распределение вероятностей каждого следующего запланированного задания.



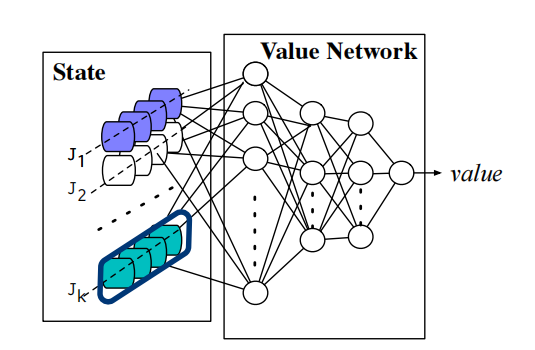
Общая архитектура RLScheduler.

На рис. показана архитектура RLScheduler и три ее основных компонента: агент, среда планирования заданий и состояние среды.



Структура сети политик RLScheduler. Его ядро - нейронная сеть на основе ядра.

На рис. подробно показана сеть. Сама сеть ядра представляет собой трехуровневую полностью связанную сеть, структурированную так же, как трехуровневый персептрон (MLP). Разница в том, что сеть на основе ядра будет применяться к каждому ожидающему заданию одно за другим, как окно.



Структура сети создания ценности RLScheduler. Его ядро - трехуровневая многослойная сеть персептронов (MLP).

На рис. 6 показана внутренняя сеть значений, которая представляет собой трехуровневую сеть MLP, но не имеет механизма ядра. Для работы с MLP векторы всех заданий будут объединены и проверены перед подачей в сеть.