МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР

ODA+ML подходы для непрерывного анализа производительности суперкомпьютеров: обзор и наши подходы

Личманов Д.И., Воеводин Вад.В.,
Матвеев В.А., Антонов А.С.

Научно-исследовательский вычислительный центр МГУ имени М.В. Ломоносова

Исследование выполнено за счет гранта Российского научного фонда №25-21-00137

Operational data analytics (ODA)

- Эффективная работа СКЦ невозможна без постоянного контроля его состояния

- **Цель ODA в сфере HPC:** обеспечить постоянный сбор, хранение и анализ информации о различных аспектах поведения суперкомпьютеров для оперативного принятия решений

- Сбор и хранение – ОК. Но анализ недостаточно развит
  - 2020(*): «Анализ ODA данных по-прежнему в основном выполняется вручную»

* Ott M., Shin W., et al. (2020). Global experiences with HPC operational data measurement, collection and analysis.

ODA: исходная классификация задач и подходов

Ввиду масштабности HPC-систем, выделяют следующие направления ODA, объединённые по целевым компонентам кластеров:

- Уровень инженерной инфраструктуры
- Уровень вычислительной аппаратуры
- Уровень системного программного обеспечения
- Уровень приложений

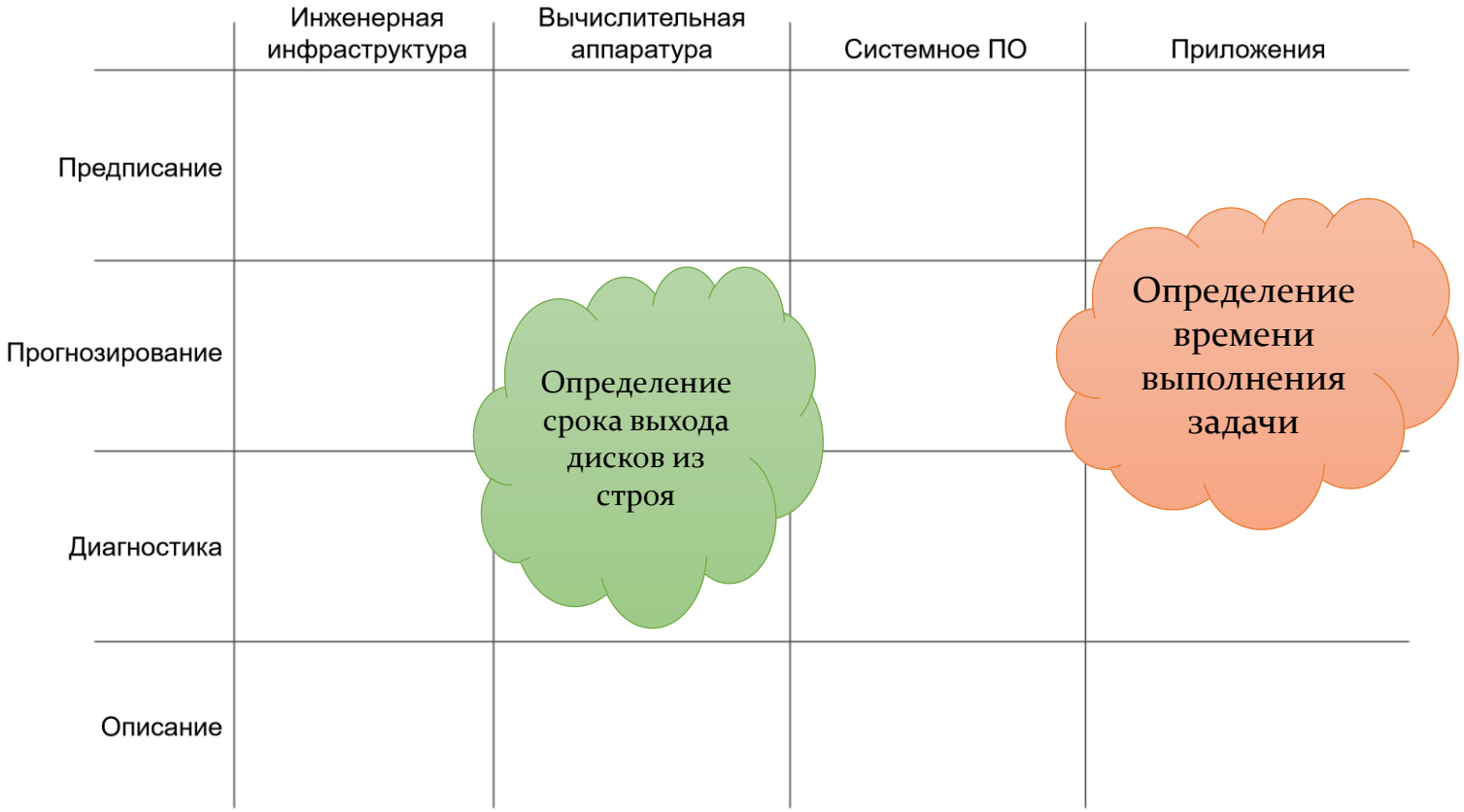


Схема классификации направлений по ODA

Обзор работ в ODA, использующих ML

	Инженерная инфраструктура	Вычислительная аппаратура	Системное ПО	Приложения
Supervised				
Unsupervised				
Reinforcement				

Число работ

- Нет
- Мало
- Достаточно
- Много

- Все направления используют Supervised, поскольку это традиционно наиболее точные методы
- Отсутствие работ по обучению с подкреплением для вычислительной аппаратуры обусловлен тем, что в основном рассматривается пара состояний “нормальное” и “аномальное”, поэтому система наград неприменима
- Отказ от unsupervised для инженерной инфраструктуры связан как с наличием большого числа размеченных исторических данных, так и со стоимостью ошибки

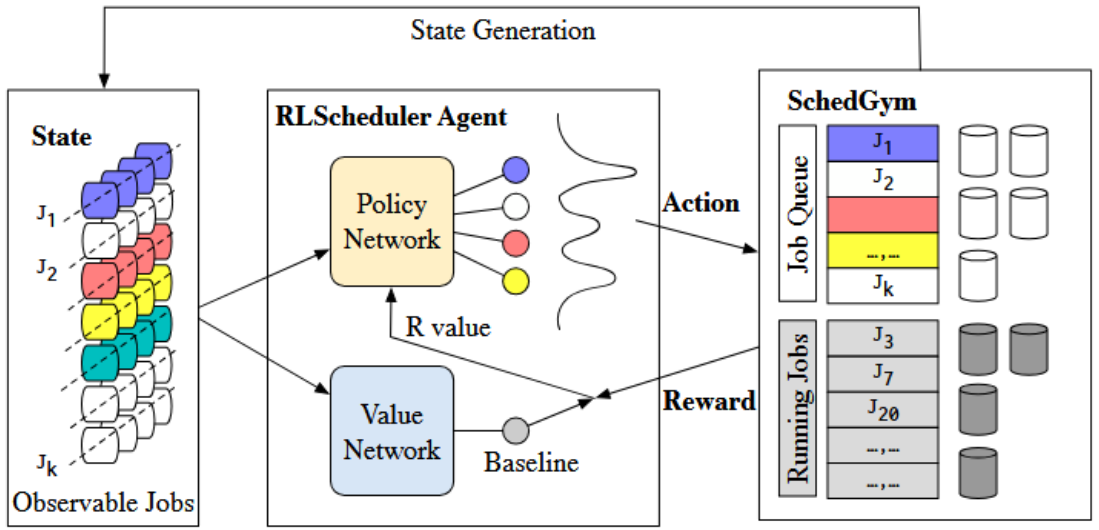
Актуальные задачи в рамках ODA

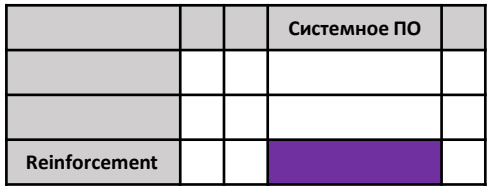
В рамках ODA для HPC в текущих работах можно выделить следующие направления, в которых использование ML наиболее распространено:

- Оптимизации планировщика заданий
- Управление системой охлаждения на инженерном уровне
- Определение системных и программных аномалий на вычислительных узлах
- Предсказание энергопотребления приложения
- Предсказание времени выполнения заданий и времени ожидания заданий в очереди

Применение обучения с подкреплением для оптимизации планировщика заданий

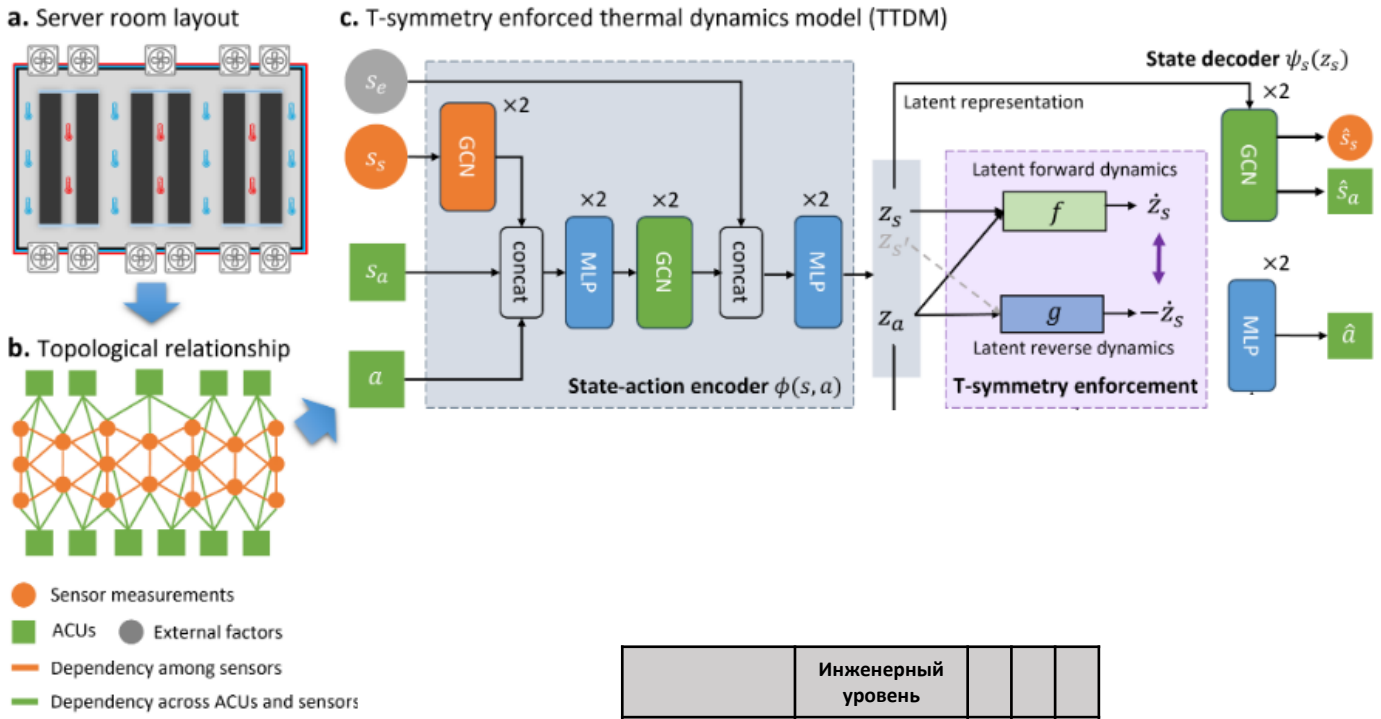
- На основе небольшого множества характеристик выполняющихся задач строится планировщик заданий, нацеленный на автоматическую адаптацию без ручной подстройки
- В качестве модели выбрана Actor-Critic модель обучения RL-агента
- За целевую метрику, являвшуюся также наградой при обучении RL-агента, брался средний показатель замедления времени выполнения задачи из-за простаивания в очереди
- Учитывался механизм Backfilling
- На реальных и синтетических пулах входных задач показывает в среднем на 20% меньшее значение замедления, чем эвристические и классические ML-методы





Zhang, D., Dai, D., He, Y., Bao, F. S., & Xie, B., 2020. RLScheduler: an automated HPC batch job scheduler using reinforcement

Оптимизация охладительного контура

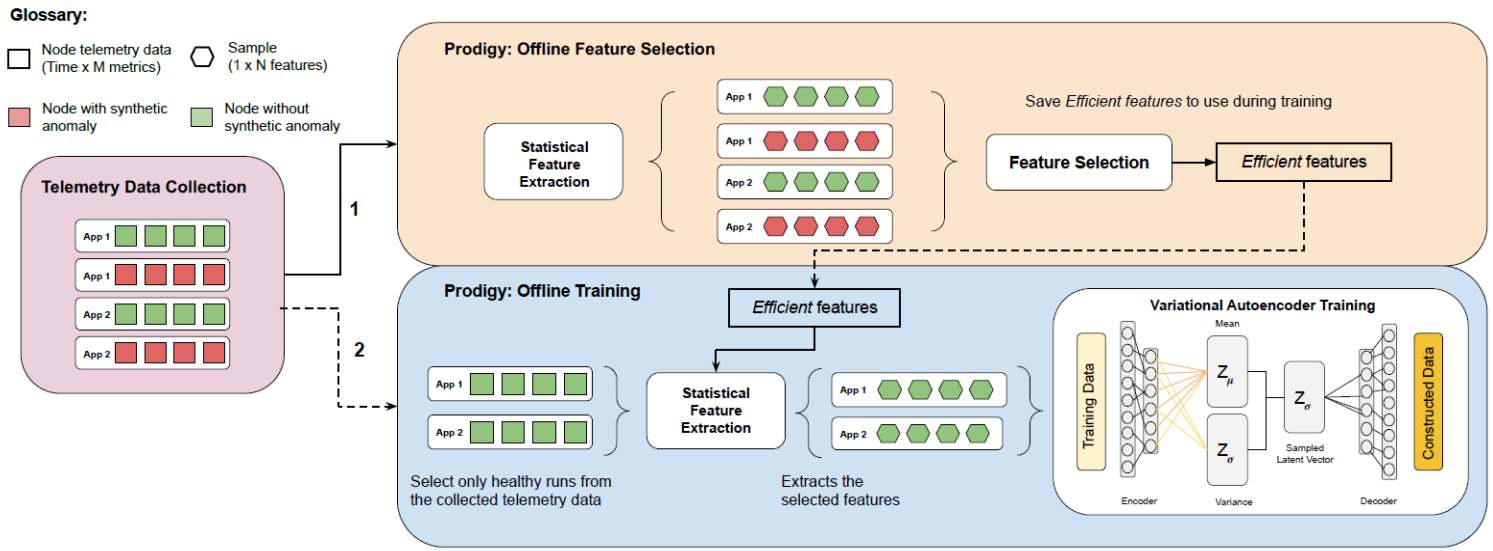


	Инженерный уровень			
Reinforcement				

- Для каждого охладительного элемента настраивается его скорость (fan speed) и степень открытости водного клапана
- Используется представление датчиков и охладительных элементов в виде графовой структуры, обрабатываемой графовой нейронной сетью
- Формула для награды учитывает инженерные ограничения на температуру входящего в контур воздуха и на рабочую температуру сервера
- На реальной системе удалось снизить энергопотребление системы охлаждения на ≈15%

Zhan, X., Zhu, X., Cheng, P., Hu, X., He, Z., Geng, H., ... & Zhao, F., 2025. Data center cooling system optimization using offline reinforcement learning

Использование вариационных автокодировщиков для определения аномалий на узлах

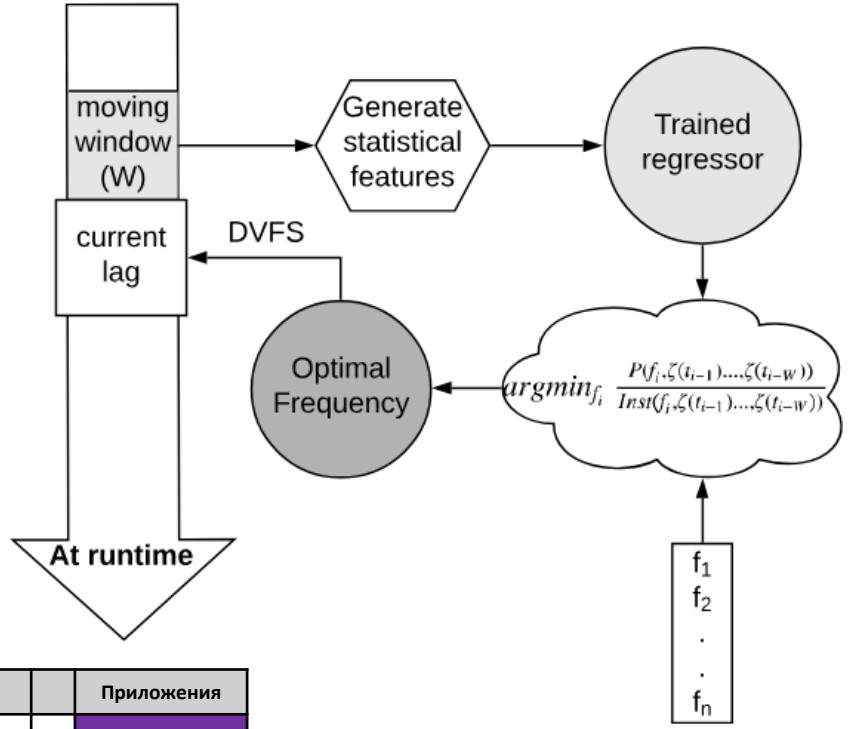


		Вычислительная аппаратура	Системное ПО	
Unsupervised				

- Изначально автокодировщик обучается на здоровых узлах, не имеющих аномалий
- Обнаружение аномалий проводится через сравнение reconstruction error с пороговым значением
- Для тестирования использовали синтетические аномалии: memleak, membw, cpuoccupy, cachecopy. Из бенчмарков - популярные BT, CG, FT и т.д.
- Используются статистические методы, позволяющие выбирать для обучения и инференса ограниченное число характеристик
- На основе малого числа данных (≈60 запусков на здоровых узлах) получилось добиться F1-score=0.96, реальные запуски на ≈1500 узлах показали F1-score=0.9

Aksar, B., Sencan, E., Schwaller, B., Aaziz, O., Leung, V. J., Brandt, J., and Coskun, A. K., 2023. Prodigy: Towards unsupervised anomaly detection in production hpc systems

Предсказание энергопотребления приложения



				Приложения
Supervised				

- Цель фреймворка – предсказывать мощность CPU и InstRetired, чтобы в рантайме выбирать частоту CPU (DVFS), минимизируя энергию
- Для сбора характеристик используют системы мониторинга DCDB и GEOPM
- Для полученных признаков по скользящему окну подсчитывается набор статистик
- Для обучения используют мультиклассовый классификатор случайный лес
- На известных приложениях получилось добиться хорошей точности по мощности, точность относительно базового подхода выросла на 0.25
- Нет апробации на реальных данных

Ozer, G., Garg, S., Davoudi, N., Poerwawinata, G., Maiterth, M., Netti, A., & Tafani, D., 2019. Towards a predictive energy model for HPC runtime systems using supervised learning.

Статус по направлению Applications

- Несмотря на большое количество работ по направлению приложений, подавляющее большинство работ по этой теме в качестве основной задачи ставят эффективность работы HPC-системы в целом
- Почти отсутствуют работы, посвящённые эффективности приложения как такового

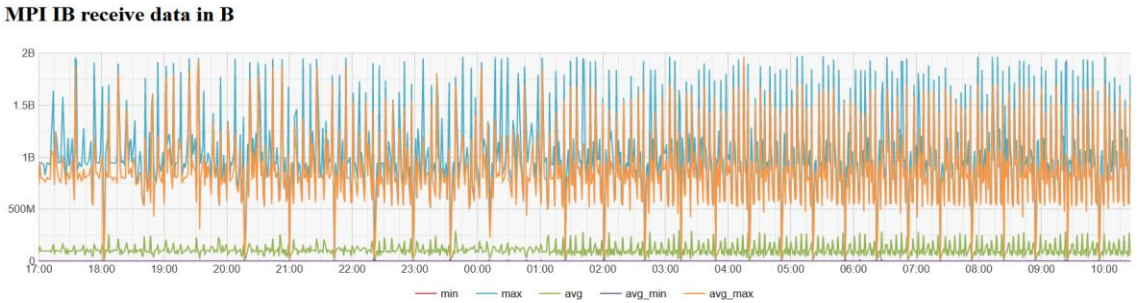
Наш подход к ODA+ML

- Мы хотим автоматически выявлять актуальные конкретные проблемы с производительностью в пользовательских заданиях
  - Актуальные проблемы: ошибки при предсказании переходов, использование сложной арифметики, насыщение канала к DRAM памяти
  - “Конкретные” проблемы означает, что мы понимаем, какие дальнейшие шаги по их устранению можно предпринять
  - Предполагается использование методов ML
- Конечная цель: создание универсальной рекомендательной системы для пользователей суперкомпьютера

Входные данные для предлагаемого подхода

- Основной источник данных: система мониторинга на вычислительных узлах
- Каждое задание представляется многомерным временным рядом
- Основная сложность – требуются данные от процессорных датчиков (PMU), но одновременно можно собирать только 4 аппаратных датчика(на СК Ломоносов-2)
- Использование мультиплексирования привносит накладные расходы и снижает точность

Число промахов в LLC кэш-память в секунду



Объём данных (в байтах), полученных по сети

Основная идея предлагаемого подхода

- Разработка точных формул для обнаружения определённой проблемы производительности на основании метрик, полученных от средства мониторинга
  - Для некоторых случаев есть готовые формулы (Intel Vtune, pmu-tools, …)
- Апробация на бенчмарках с целью проверки корректности формул и замера точности
- Генерация обучающего множества
  - Моделирование различной интенсивности целевой проблемы
  - Создание разнообразного фона для неэффективного приложения
- Обучение классификатора, работающего на меньшем наборе аппаратных событий, но предоставляющего приемлемую точность

Примеры формул

False sharing rate

$$= \frac{\text{L2\_RQSTS:DEMAND\_RFO\_MISS}}{OFFCORE\_RESPONSE\_0:DMND\_DATA\_RD + MEM\_LOAD\_UOPS\_RETIRED:L1\_MISS} \times$$

$$\frac{OFFCORE\_REQUESTS\_OUTSTANDING:DEMAND\_RFO\_CYCLES}{CYCLE\_ACTIVITY:STALLS\_MEM\_ANY} \times$$

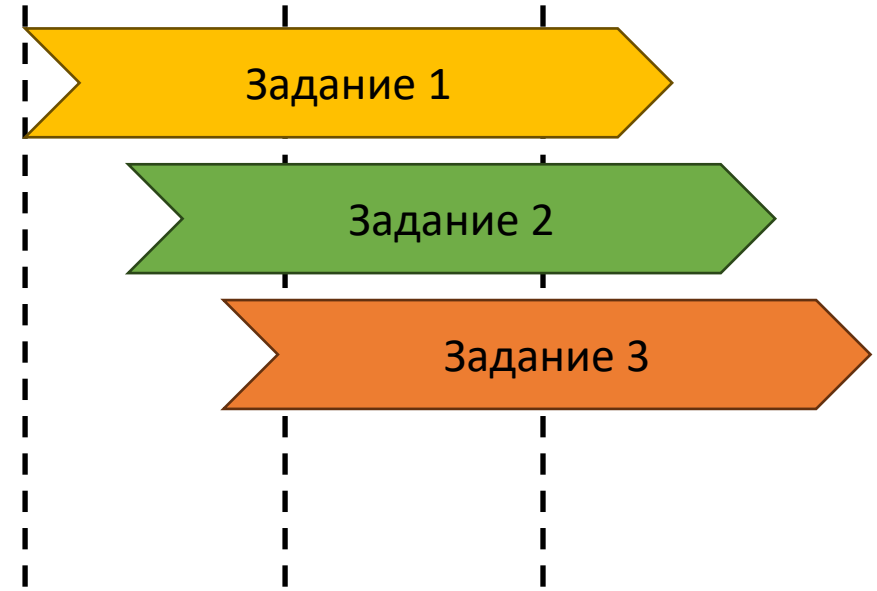
$$\frac{OFFCORE\_RESPONSE\_0\_REMOTE\_DRAM}{(OFFCORE\_RESPONSE\_0\_LOCAL\_DRAM + OFFCORE\_RESPONSE\_0\_REMOTE\_DRAM + OFFCORE\_RESPONSE\_0\_DMND\_DATA\_RD\_ANY)}$$

Branch misprediction rate

$$= \frac{BR\_MISP\_RETIRED.ALL\_BRANCHES\_PS}{BR\_MISP\_RETIRED.ALL\_BRANCHES\_PS + MACHINE\_CLEARS.COUNT} \times$$

$$\frac{UOPS\_ISSUED.ANY - UOPS\_RETIRED.RETIRE\_SLOTS + 4 * INT\_MISC.RECOVERY\_CYCLES}{2 * CPU\_CLK\_UNHALTED.THREAD\_ANY}$$

Что хотим разработать в итоге



Сбор и агрегация данных мониторинга

Определение наличия проблемы №1 с помощью ML

Определение наличия проблемы №2 с помощью ML

...

Если вероятность наличия проблемы больше порога

Job_ID XXX	Доля времени на ошибочное предсказание переходов слишком высоко	Рекомендуется: - Поставить наиболее вероятные условные операторы первыми - Переместить условные операторы для более раннего выполнения