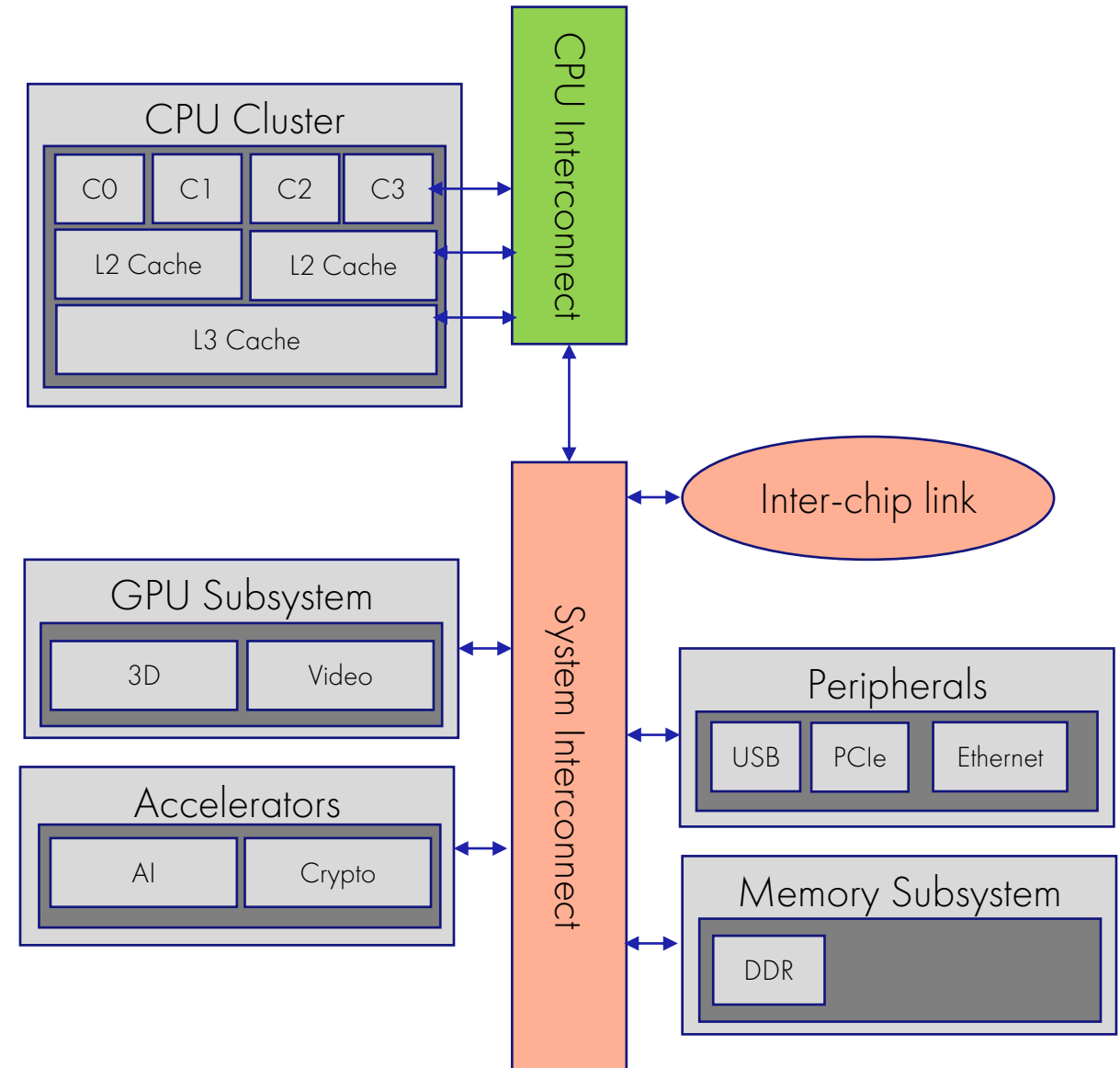


Мониторинг  
производительности в RISC-V  
2024

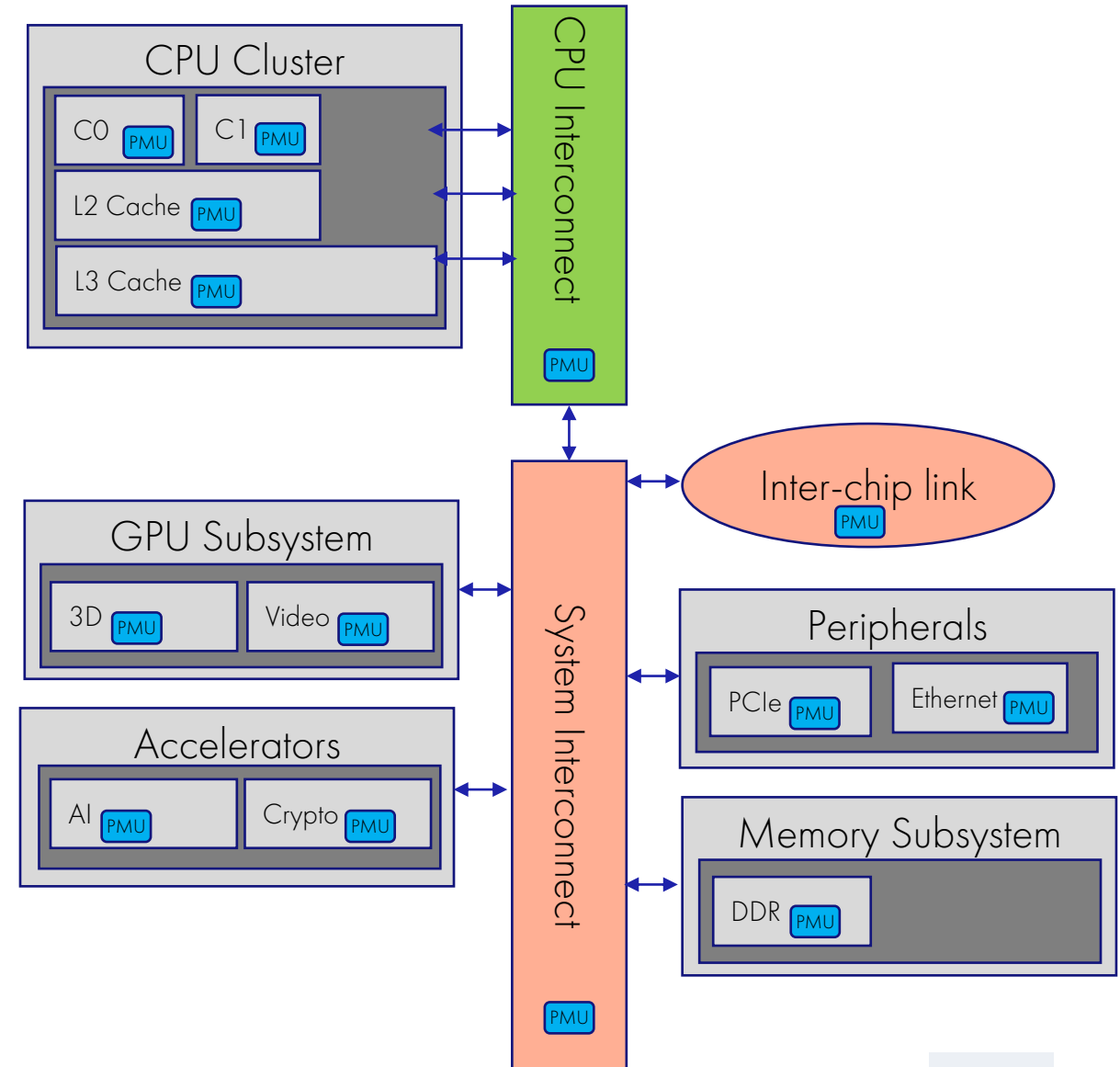
## Зачем нужен мониторинг?

- **Современные SoC – очень сложные системы со множеством связанных компонент**
- **Где находятся узкие места по производительности?**
- **Как сильно их влияние?**
- **Как их исправить?**



# Аппаратные средства мониторинга

- Performance Monitoring Units (PMU) во многих (желательно во всех) компонентах
  - Решается проблема поиска узких мест
  - Новая проблема – зоопарк систем мониторинга от разных производителей



# Стандартизация средств мониторинга в RISC-V

- **Цель – спецификации, определяющие что и как мы можем измерять**
  - Существенно облегчает задачу анализа производительности
  - Позволяет создать стандартные инструменты, работающие на всех RISC-V системах
- **Основные группы в RVI:**
  - **Performance Analysis SIG**
  - **Debug, Trace, and Performance Monitoring (DTPM) SIG**

# Performance Analysis SIG

<https://lists.riscv.org/g/sig-perf-analysis>

**Начало работы – Апрель 2022**

## **Спецификации и рабочие группы**

- **Zicntr, Zihpm, Sscofpmf, Smcdeleg, Ssccfg, Control Transfer Records (CTR), Smcntrpmf, Performance Events TG, Performance Event Sampling TG**

# Debug, Trace, and Performance Monitoring (DTPM) SIG

<https://lists.riscv.org/g/sig-debug-trace-perf-mon>

Начало работы – Август 2022

Основной фокус – аппаратный трэйсинг

- Спецификации и рабочие группы
  - N-Trace, E-Trace, Self Hosted Trace TG

# Zicntr, Zihpm, Sscofpmf

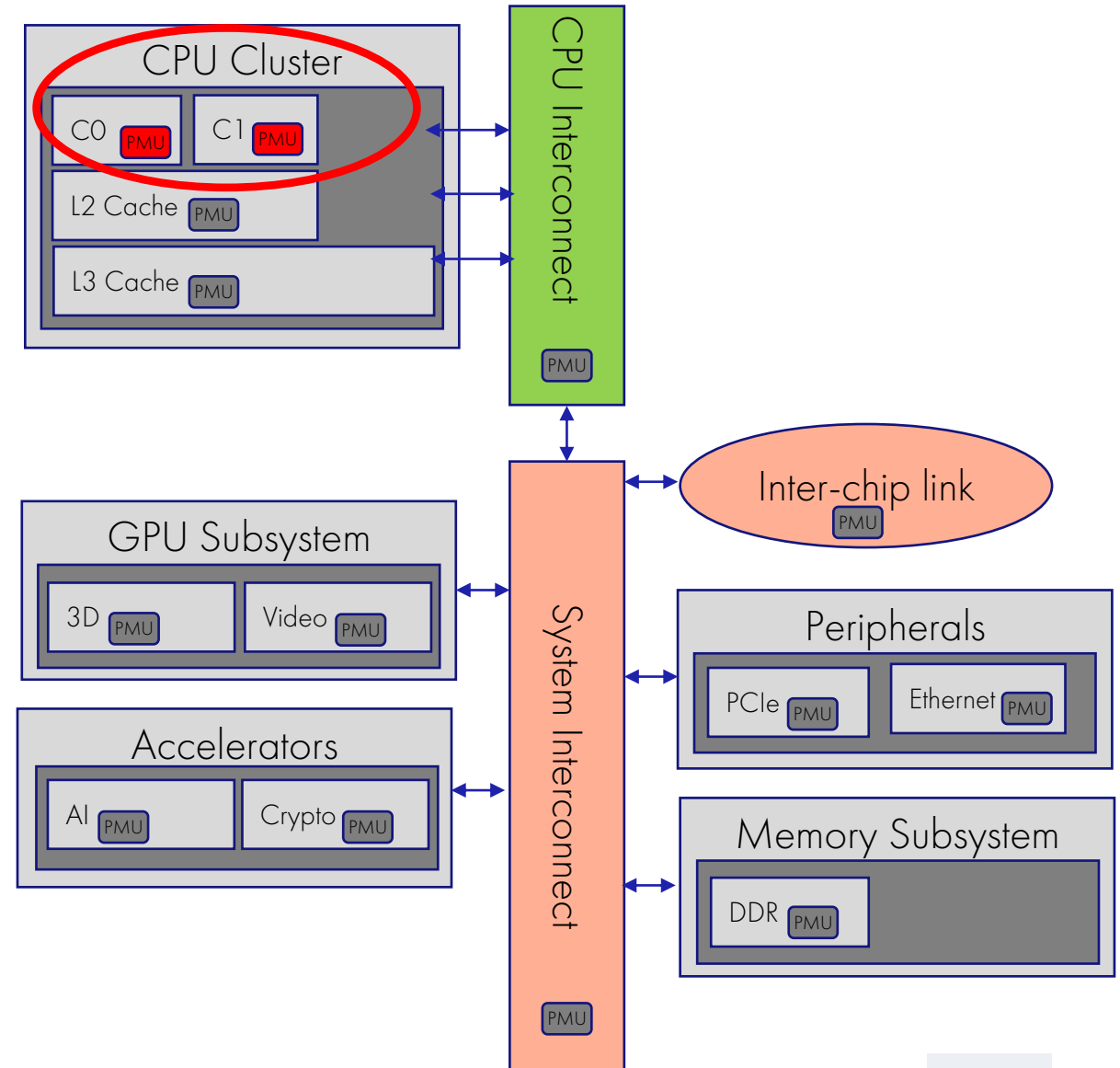
3 фиксированных и до 29 программируемых счетчиков для каждого RISC-V hart

Прерывания при переполнении

- Самплинг – связь событий с инструкциями

Базовые, критически важные расширения для анализа производительности RISC-V процессоров

- Полная поддержка в Linux perf



# Control Transfer Records (CTR)

- Кольцевой массив CSR-регистров, в котором сохраняются N последних переходов
  - Каждая запись содержит:
    - Source IP – адрес откуда был переход
    - Target IP – адрес куда был переход
    - Дополнительная (опциональная) мета-информация
- Основная цель – сбор профилей для PGO (profile guided optimization) в компиляторах. Например – Google AutoFDO.
- Статус – Ratification Ready



# Performance Events TG

- Zihpm определяет механизм сбора событий CPU - сам набор событий никак не регламентирован
  - Каждый производитель изобретает что-то свое
- Performance Events TG определяет стандартный набор событий со строго заданными именами и описаниями
  - Позволяет создавать единые, работающие на всех RISC-V платформах инструменты анализа
- Статус – Work In Progress
  - Работа ведется с начала 2024 года, высокая степень готовности
  - Freeze – конец 2024/начало 2025

# Performance Event Sampling TG

## 3 новых расширения

### 1. Precise Local Counter Overflow Interrupts (Precise LCOFI)

- LCOFI прерывания гарантируются на инструкции, следующей той, что вызвала переполнение

### 2. Precise Event Sample Attribution

- Собирает и предоставляет точный адрес инструкции (IP) вместе в LCOFI
- Для имплементаций, которые не могут предоставить #1

# Performance Event Sampling TG (продолжение)

## 3. Instruction Sampling

- Собирает и предоставляет дополнительную мета-информацию вместе с сэмплом – latency, data address и т.п.
- Статус – Work In Progress
  - Работа началась совсем недавно
  - Ожидаемый срок готовности – середина 2025 года

# Применение аппаратной трассировки

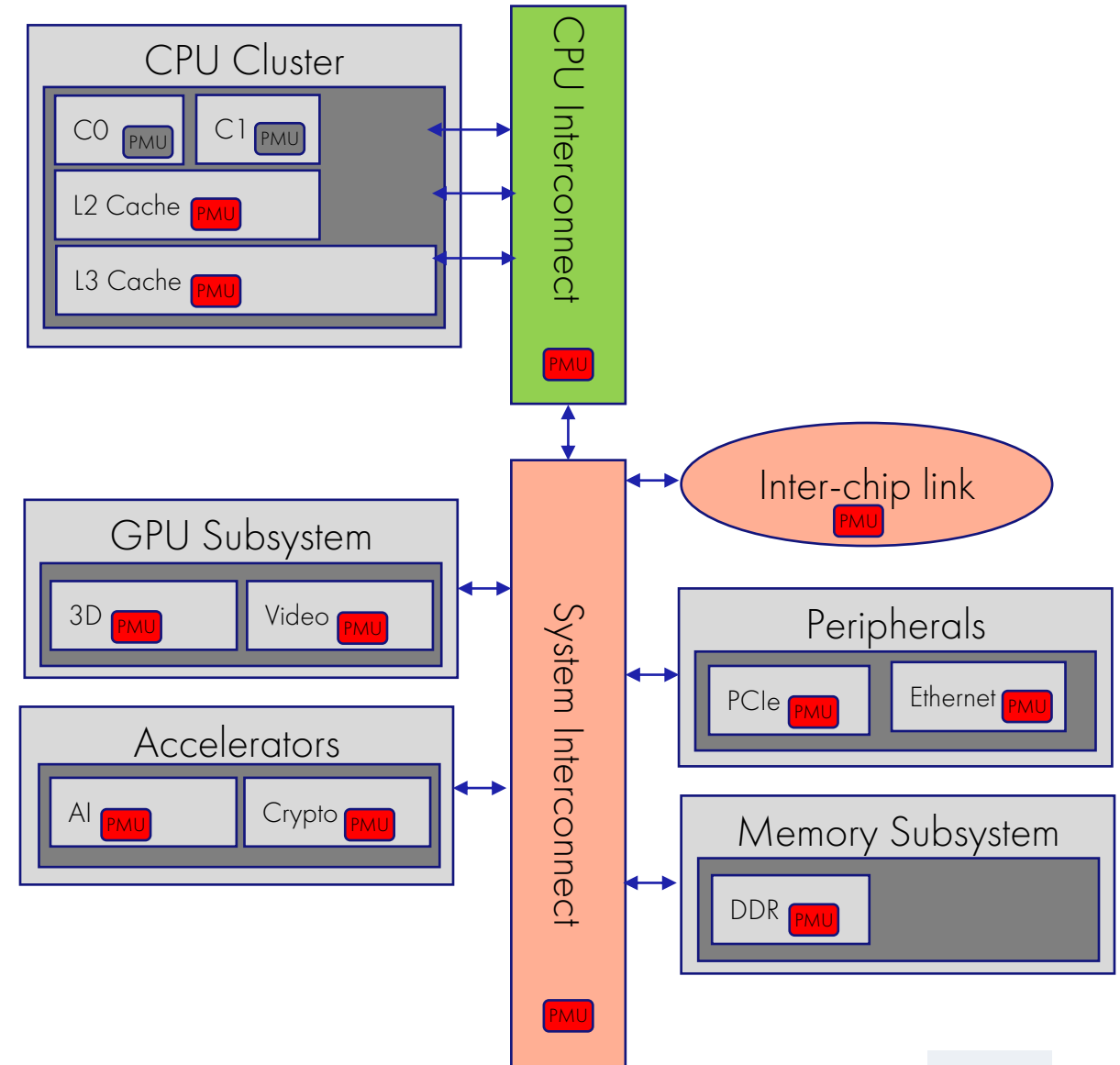
- **Анализ аномалий**
  - **Пример: короткие ( $< 1\text{ ms}$ ) транзакции, периодически некоторые транзакции работают медленнее ожидаемого**
  - **Традиционные подходы (сэмплирование) работают плохо**
  - **Решение – использование аппаратной трассировки**
    - **Сбор трассы в режиме кольцевого буфера**
    - **Остановка в момент появления аномалии**
    - **Сравнение и анализ потока исполнения в аномальных и нормальных случаях**

## N-Trace, E-Trace, Self Hosted Trace TG

- **N-Trace, E-Trace** – позволяют делать **branch tracing**
  - В трассу записываются только переходы, которые не восстановимы из кода
  - Позволяет полностью восстановить порядок выполнения инструкций
  - Приемлемый размер трасс и **runtime overhead**
- **Self Hosted Trace TG** – новое расширение поверх N-Trace, E-Trace
  - Фокус – программный сбор трассы “изнутри” системы
  - Удобный CSR интерфейс
  - Трасса собирается в буфере в памяти
  - Статус - **Work In Progress**

# Uncore PMU Interface

- НРМ покрывает только hart-ы
- Необходимо разработать спецификацию для остальных частей SoC
  - Работа еще не начата



## Итоги

- Базовые возможности мониторинга стандартизованы и успешно применяются
  - Как аппаратные так и программные
- Активная работа в нескольких RVI группах по расширению

Контакты: [dmitriy.ryabtsev@syntacore.com](mailto:dmitriy.ryabtsev@syntacore.com)