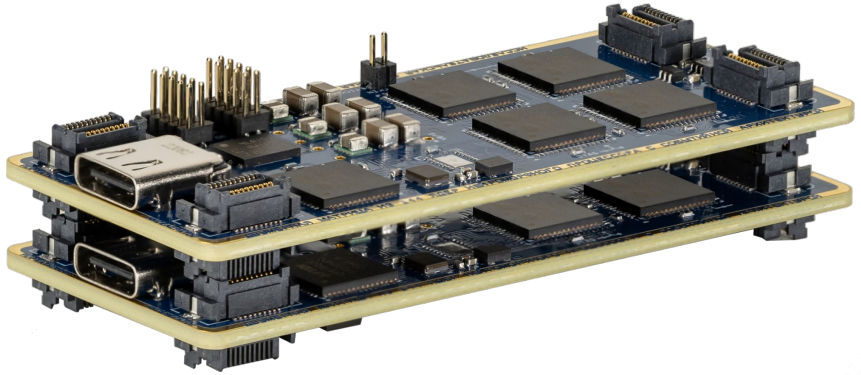
Kaspersky Neuromorphic AI Conference 2025

Kaspersky Neuromorphic AI



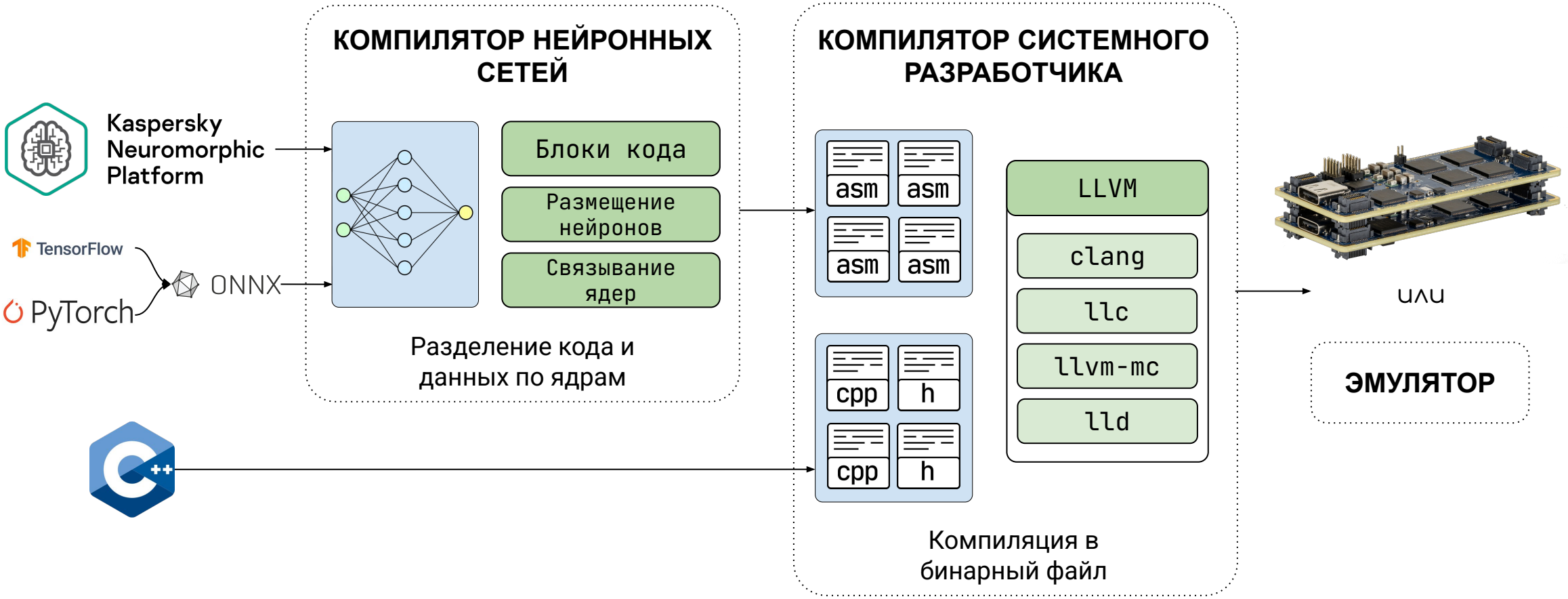
SDK НЕЙРОМОРФНОГО ПРОЦЕССОРА «АЛТАЙ-3»

Кострицын Игорь

ООО «Мотив Нейроморфные Технологии»

ОБЩИЙ ПАЙПЛАЙН КОМПИЛЯЦИИ

Kaspersky Neuromorphic AI



МОДЕЛЬ ЯДРА «АЛТАЙ-3»

Kaspersky Neuromorphic AI

«АЛТАЙ-3» – СОБЫТИЙНЫЙ ПРОЦЕССОР

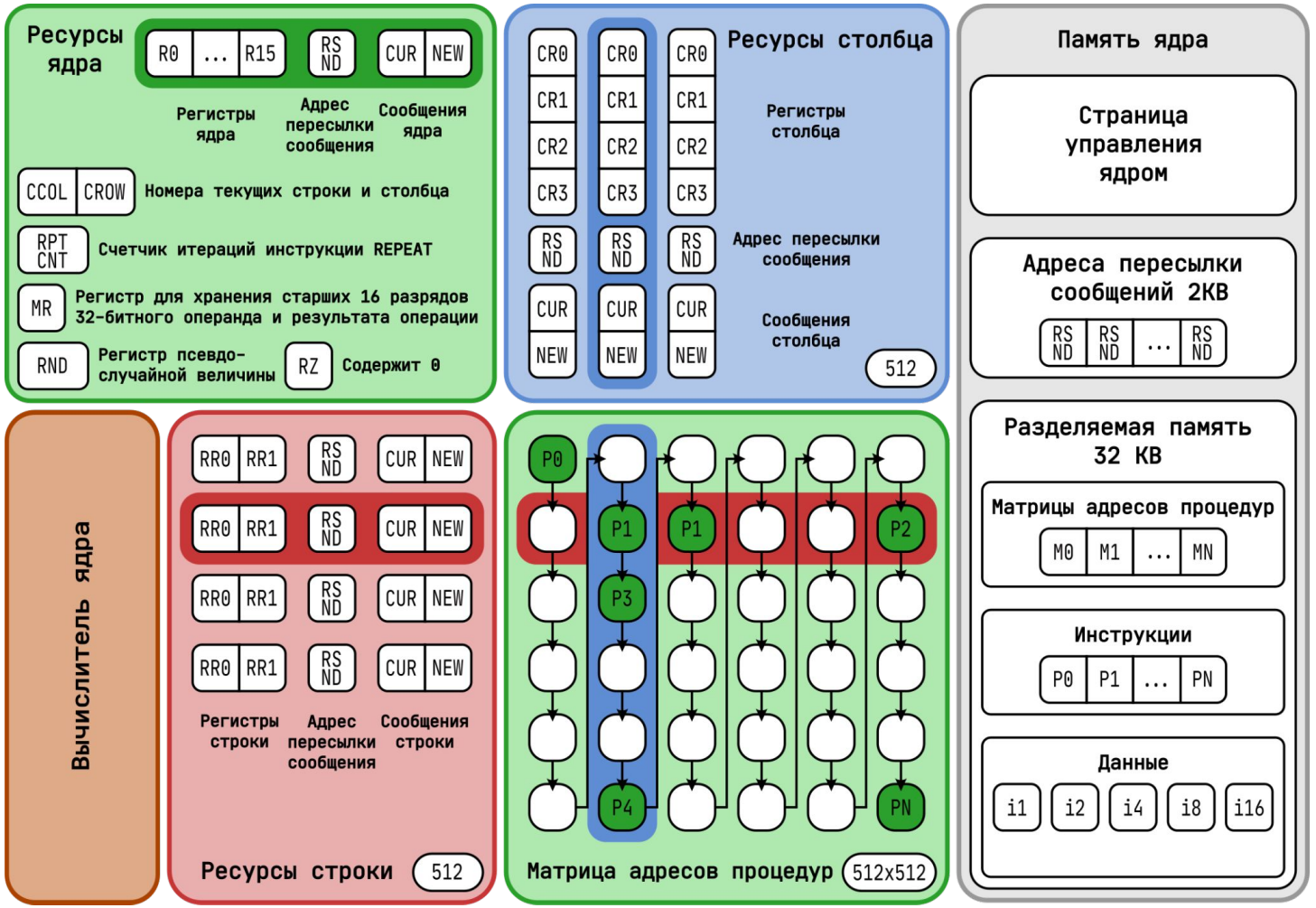
Исполнение процедур может быть пропущено, если нет входных сообщений

КОНТЕКСТ ПРОЦЕДУРЫ ВАЖЕН

Один код исполняется для разных данных. Процедура имеет доступ к определенному набору регистров

ПЕРЕДАЧА ДАННЫХ

Между процедурами данные передаются через контекстные регистры или память, а между ядрами – с помощью сообщений



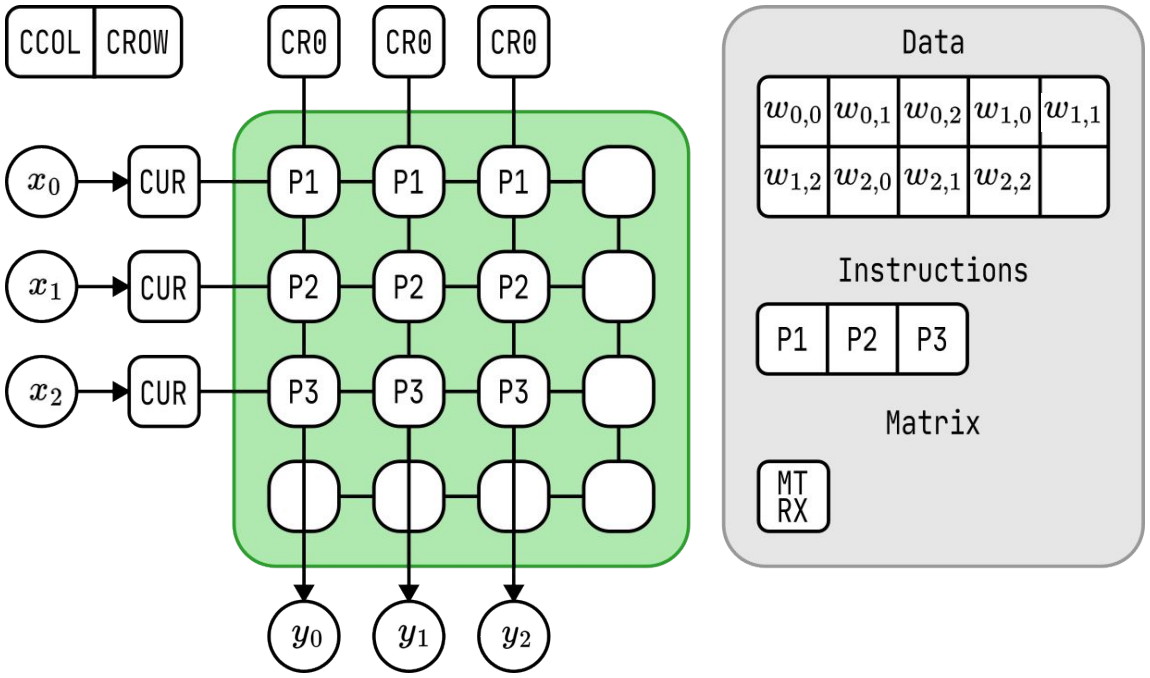
КОМПИЛЯТОР НЕЙРОННЫХ СЕТЕЙ

Kaspersky Neuromorphic AI

УМНОЖЕНИЕ МАТРИЦЫ НА ВЕКТОР

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} w_{0,0} & w_{0,1} & w_{0,2} \\ w_{1,0} & w_{1,1} & w_{1,2} \\ w_{2,0} & w_{2,1} & w_{2,2} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}$$

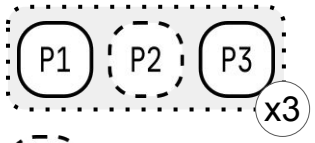
ЭТАП 1. ПЛАНИРОВАНИЕ ВЫЧИСЛЕНИЙ



ЭТАП 2. ПОСТРОЕНИЕ ПРОЦЕДУР НЕЙРОНА

```
P1: add  0,RZ,CR0                          Инициализация
    macb (CROW,CCOL)ROWEPI,CR0             MAC-операция

P2: macb (CROW,CCOL)ROWEPI,CR0             MAC-операция

P3: macb (CROW,CCOL)ROWEPI,CR0             MAC-операция
    post NEW,TGT_ROW,TGT_DXDY,CR0          Отправка результата
```

ЭТАП 3. ФОРМИРОВАНИЕ МАТРИЦЫ ПРОЦЕДУР



Матрица формируется и хранится в сжатом виде

Можно пропустить, если входного сообщения нет ($x_i = 0$)

Пустые ячейки не исполняются

КОМПИЛЯТОР НЕЙРОННЫХ СЕТЕЙ

Kaspersky Neuromorphic AI

ЭТАП 4. ОЦЕНКА РЕСУРСОВ

32KB — Общее ограничение памяти на **данные**, **инструкции**, и **матрицы адресов процедур** на ядро

512 — Максимальное число сообщений на **строки** и на **столбцы** на ядро

НЕСКОЛЬКО ЯДЕР

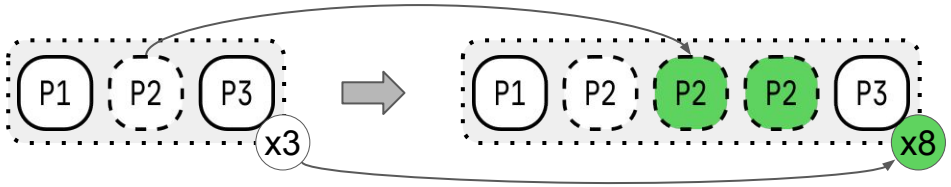
RS ND — Позволяет задать адрес ядра, которому будет ретранслировано входящее сообщение

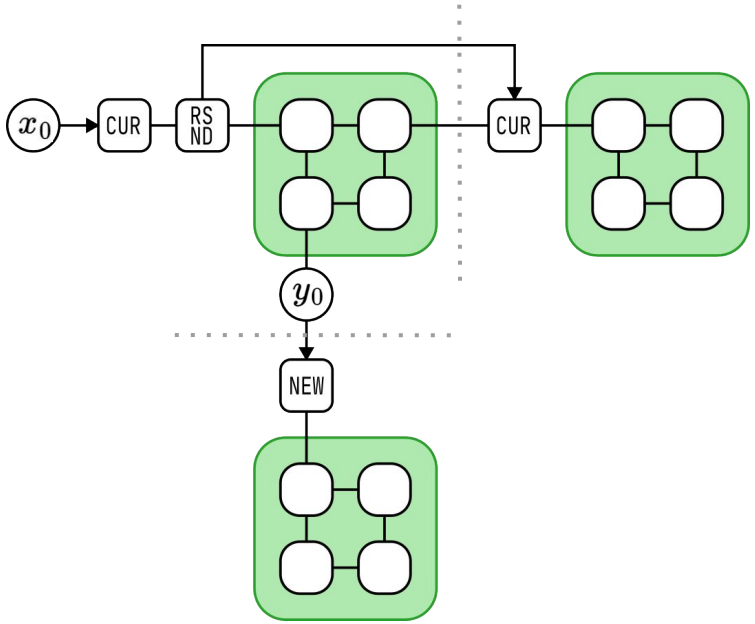
POST — Инструкция позволяет отправить сообщение с данными другому ядру для дальнейшей обработки

ЭТАП 5. МАСШТАБИРОВАНИЕ

ОДНО ЯДРО

В рамках ядра изменение параметров слоя влияет положение процедур в матрице и объем хранимых данных





КОМПИЛЯТОР СИСТЕМНОГО РАЗРАБОТЧИКА

Kaspersky Neuromorphic AI

«АЛТАЙ-3» КАК БЭКЕНД LLVM

Стандартный пайплайн компиляции для нестандартной архитектуры

ПОДДЕРЖКА ЯЗЫКА ВЫСОКОГО УРОВНЯ

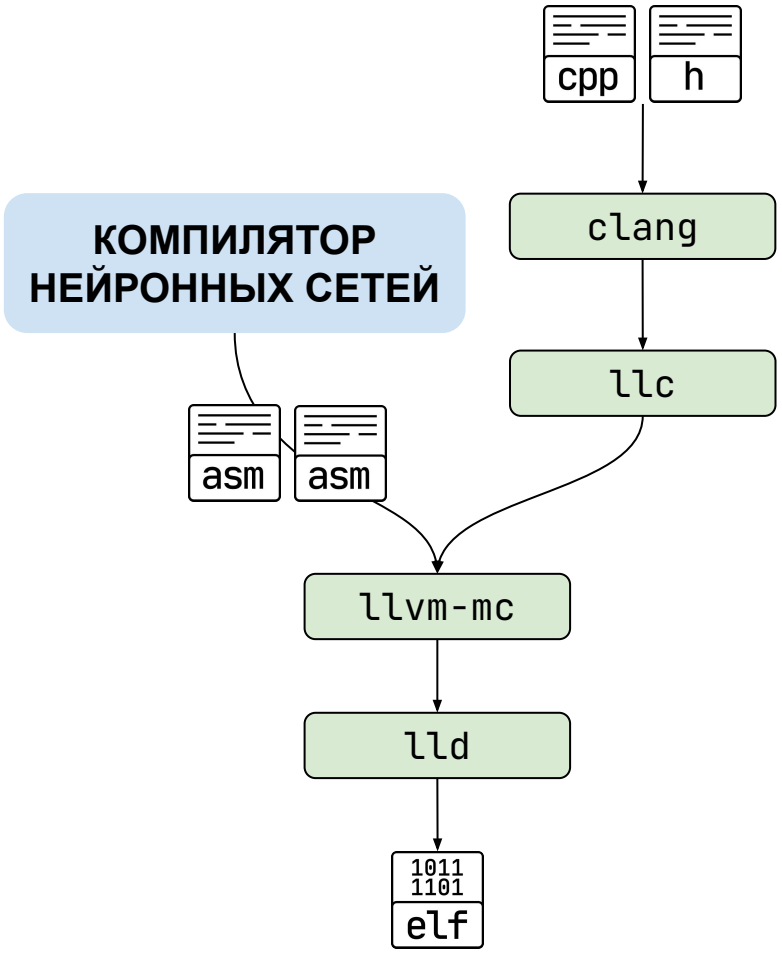
Ускорение разработки при переходе от asm к C/C++

ПРЯМОЕ УПРАВЛЕНИЕ РЕСУРСАМИ ЯДРА

Оптимальное использование контекстных регистров и памяти разработчиком с помощью макросов, атрибутов, builtin-функций

РЕАЛИЗАЦИЯ АЛГОРИТМОВ ОБЩЕГО НАЗНАЧЕНИЯ

Можно использовать «Алтай-3» без хоста и ОС для работы с внешними устройствами и управления другими ядрами



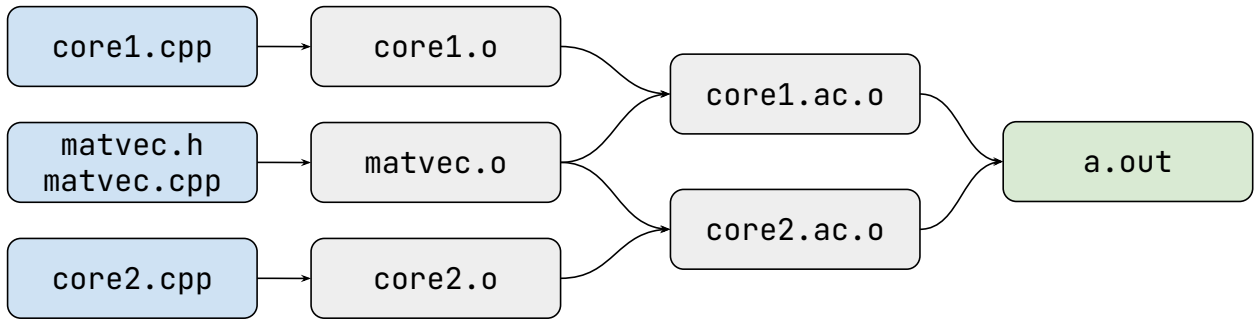
КОМПИЛЯТОР СИСТЕМНОГО РАЗРАБОТЧИКА

Kaspersky Neuromorphic AI

```cpp
// matvec.cpp
#include "altai.h"

extern int weights[];
int sum = 0;
int i = 0;

void P1() {
 // Initialization
 sum = 0;
 i = 0;

 // MAC
 sum += weights[i++] * ROWEPI;
}

void P2() {
 // MAC
 sum += weights[i++] * ROWEPI;
}

void P3() {
 // MAC
 sum += weights[i++] * ROWEPI;

 // Finalization
 int tgt_col = CCOL;
 int tgt_dxdy = BUILD_DXDY(0, 1);

 POST(COL_MSG, COL, tgt_dxdy, tgt_col);
}
```

```cpp
// core1.cpp
#include "altai.h"
#include "matvec/matvec.h"

int weights[] { 1, 2, 3, 4, 5, 6, 7, 8, 9 };

__attribute__((altai_matrix))
void *mtrx;

__attribute__((altai_matrix_init("mtrx")))
void init_mtrx() {
 ALTAI_PROCEDURE(P1, Activation::Always)
 ALTAI_PROCEDURE(P2, Activation::Rowepi)
 ALTAI_PROCEDURE(P3, Activation::Always)
 ALTAI_PROCEDURE_EMPTY(509)
 ALTAI_PROCEDURE(P1, Activation::Always)
 ALTAI_PROCEDURE(P2, Activation::Rowepi)
 ALTAI_PROCEDURE(P3, Activation::Always)
 ALTAI_PROCEDURE_EMPTY(509)
 ALTAI_PROCEDURE(P1, Activation::Always)
 ALTAI_PROCEDURE(P2, Activation::Rowepi)
 ALTAI_PROCEDURE(P3, Activation::Always)
}
```

```cpp
// core2.cpp
#include "altai.h"
#include "matvec/matvec.h"

int weights[] { 2, 2, 1 };

void CORE2_P1() {
 // Initialization
 sum = COLEPI;
 i = 0;

 // MAC
 sum += weights[i++] * ROWEPI;
}

__attribute__((altai_matrix))
void *mtrx;

__attribute__((altai_matrix_init("mtrx")))
void init_mtrx() {
 ALTAI_PROCEDURE(CORE2_P1, Activation::Always)
 ALTAI_PROCEDURE(P2, Activation::Rowepi)
 ALTAI_PROCEDURE(P3, Activation::Always)
}
```



ЭМУЛЯТОР «АЛТАЙ-3»

Kaspersky Neuromorphic AI

МУЛЬТИПРОЦЕССНОСТЬ

Каждый процесс эмулирует работу четверки ядер и одного маршрутизатора

РАСПРЕДЕЛЕННОСТЬ

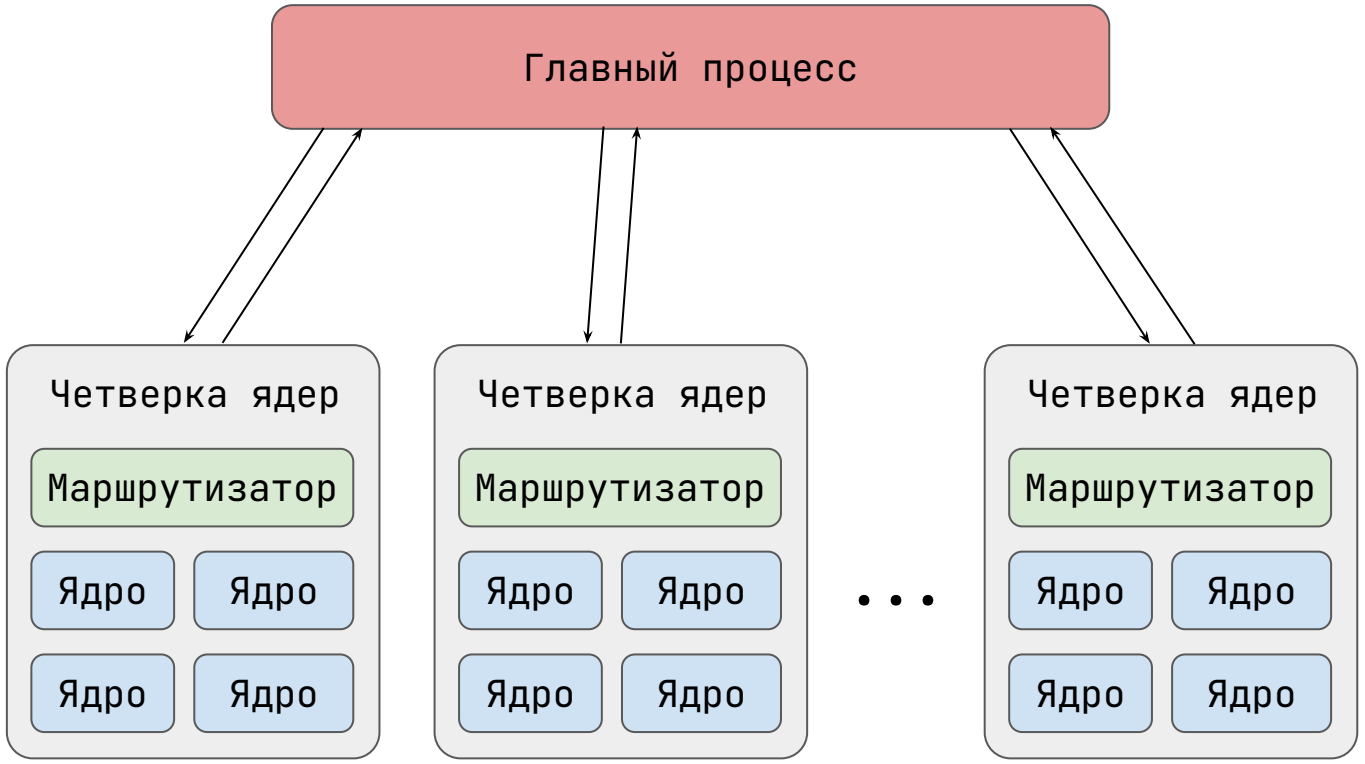
Четверки ядер могут находиться на нескольких хостах

Обмен сообщениями между четверками выполняется через общий менеджер

ЭМУЛЯЦИЯ ПО ВРЕМЕННЫМ КВОТАМ

Четверка ядер эмулирует заданное число тактов процессора независимо от других процессов

Затем идет фаза обмена сообщениями



KASPERSKY NEUROMORPHIC PLATFORM

Kaspersky Neuromorphic AI

ПРОГРАММИРУЕМАЯ МОДЕЛЬ НЕЙРОНА

Переход от фиксированной модели LIF-нейрона с линейной утечкой к программируемой модели нейрона, поддержка BLIFAT-нейрона

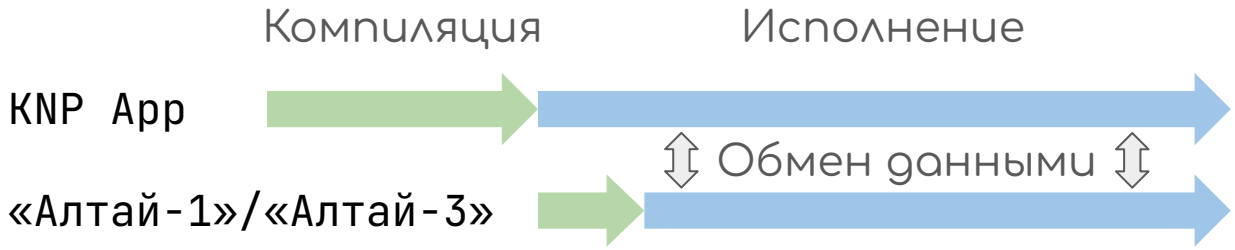
ОБУЧЕНИЕ

Обучение нейросетей в KNP. Возможность включать новые ядра с нейронами в сеть во время исполнения

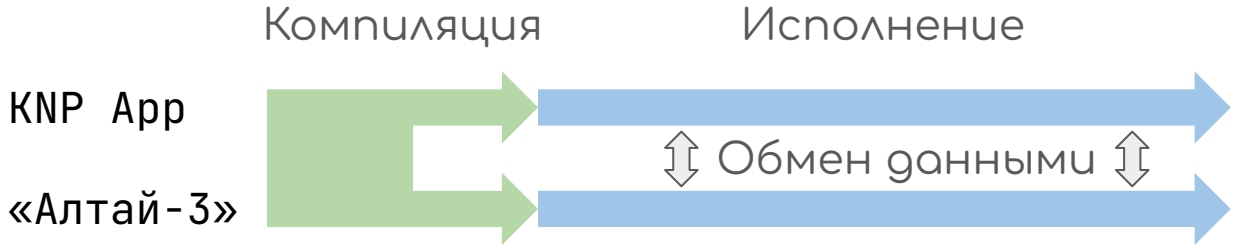
УГЛУБЛЕННАЯ ИНТЕГРАЦИЯ С KNP

Прорабатывается возможность отделения кода, специфичного для «Алтай-3» на этапе компиляции программы, использующей KNP

Текущий пайплайн



Перспективный пайплайн



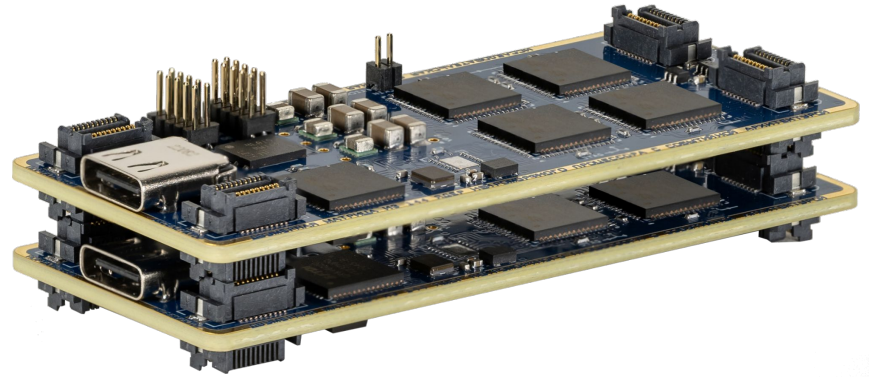
СПАСИБО ЗА ВНИМАНИЕ!

Kaspersky Neuromorphic AI

Кострицын Игорь

Ведущий разработчик инструментального ПО
ООО «Мотив Нейроморфные Технологии»

ikostritsyn@motivnt.ru



Сайт

Telegram

Дзен

KNP