COURSE-AS1 124365, г. Москва, Зеленоград, ул. Радио, д. 23, стр. 2. http://www.course-as.ru

Язык Параллельного Программирования

(The Parallel Programming Language)

"CAPER 5.0"

Москва 2016/2018г.

Язык Параллельного Программирования
(The Parallel Programming Language)
CAPER

Прямое управление процессорными ядрами средствами языка CAPER 5.0 (2017)

Язык параллельного программирования **Caper** получил новое развитие - создана новая версия его виртуальной машины **ИВМ** (***И**нтегральная **В**иртуальная **М**ашина*), которая является интегратором локализованных виртуальных машин **ЛВМ** (***Л**окальная **В**иртуальная **М**ашина*), запускаемых в качестве самостоятельных потоков **ОС** (*Операционная Система*) на избираемых ядрах процессора.

Каждая локальная виртуальная машина **ЛВМ** обеспечивает псевдопараллельное исполнение множества процедур, записанных на языке.

Интегральная виртуальная машина позволяет использовать различные режимы запуска локальных виртуальных машин:

1. С указанием ядер процессора, на которых должны быть запущены **ЛВМ**,
2. Без указания ядер, но самостоятельными потоками **ОС**,
3. С первичной загрузкой **ЛВМ** до получения ими заданий на исполнение параллельных процессов языка, или без оного - т.е. со стартами **ЛВМ** по необходимости.

Соответственно, **Caper** снабжен средствами выгрузки **ЛВМ** и/или их перераспределения по другим ядрам процессора.

В связи с новыми возможностями виртуальных машин в языке модифицированы инструкции императивного запуска множеств параллельных процессов и добавлены функции параметризации управления локальными виртуальными машинами.

Проведены тесты, которые показали ускорение исполнений параллельных процессов **Caper** от 3 до 5 раз. Тестирование проводилось имитацией клеточных автоматов и нейронных сетей.

Был запущен 1-н миллион постоянно исполняемых параллельных процессов на восьмиядерном персональном компьютере при 2-х Гб-ом адресном пространстве.

Тестирование показало возможность доведения количества параллельных процессов до (1.3 – 1.5) миллиона для восьмиядерной машины.

Планируется дальнейшая разработка технологии распределения виртуальных машин языка **Caper** в среде многомашинных кластеров.

На картинке внизу изображен интерфейс тестового приложения, написанный на языке CAPER (v.5.0), использующий параллельную многоядерную обработку ангиографической серии в DSA моде для получения интегрального DSA изображения прохождения контраста.

На изображении результат обработки DICOM серии (формат 1024*1024) функцией Roadmapping", реализованной средствами языка параллельного программирования **CAPER v.5.0**.

На возможностях языка **Caper 5.0** будут базироваться медицинские программные модули проекта **"Микросекундная Рентгенология"**, что обеспечит обработку реального времени изображений сверхвысокого разрешения в потоке **1500 MB/s** и выше!

Аннотация с возможностями языка **CAPER 5.0** 2017 года **Что такое CAPER ?** (PDF).
Базовая информация по языку **CAPER** приведена ниже.

CAPER - Язык Императивного Параллельного Программирования

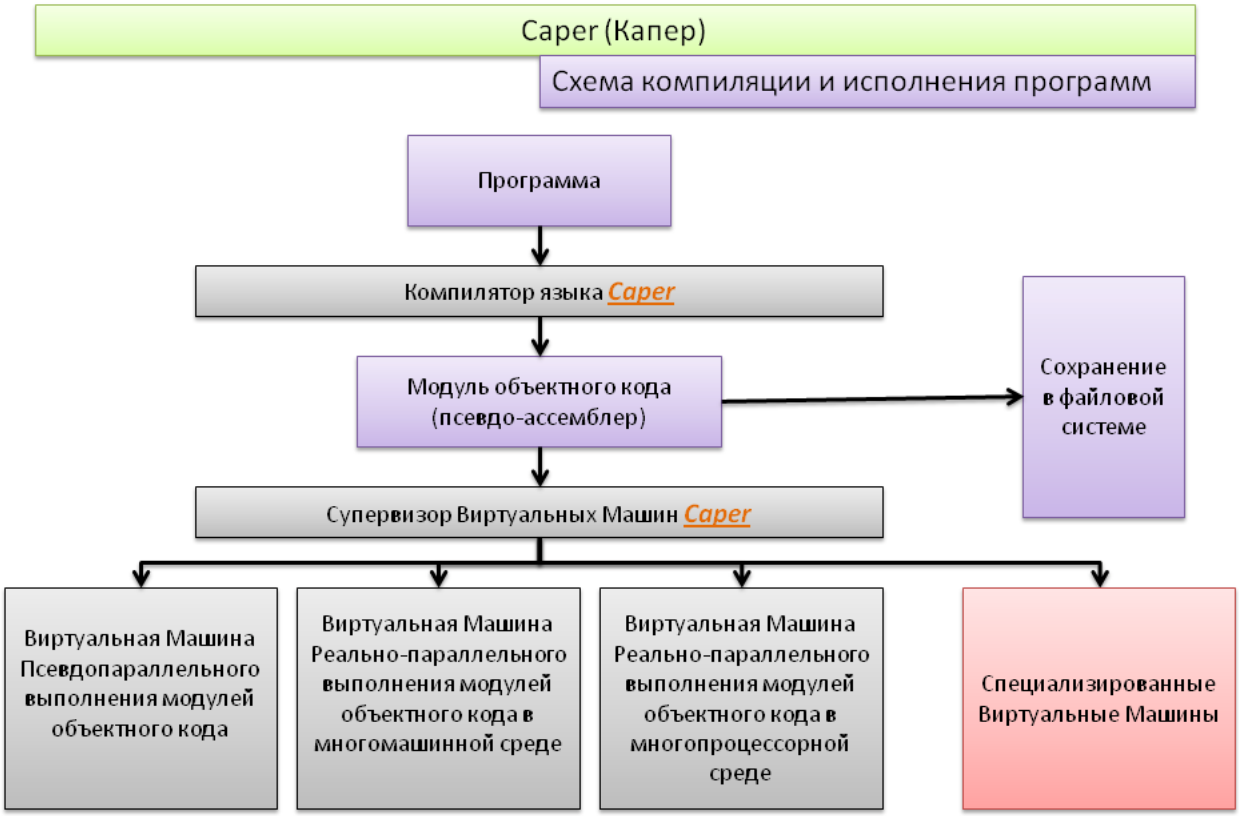
Вартанов Сергей Рубенович, к.ф.-м.н. *

Язык **CAPER** создавался с целью разрешения следующих концептуальных положений:

- структурированность с возможностью динамического изменения структуры программы в целях самоорганизации;
- возможность динамической компиляции и исполнения программы;
- параллельное исполнение элементов структуры программы;
- управление процессом вычислений на основе событий (программирование событиями).

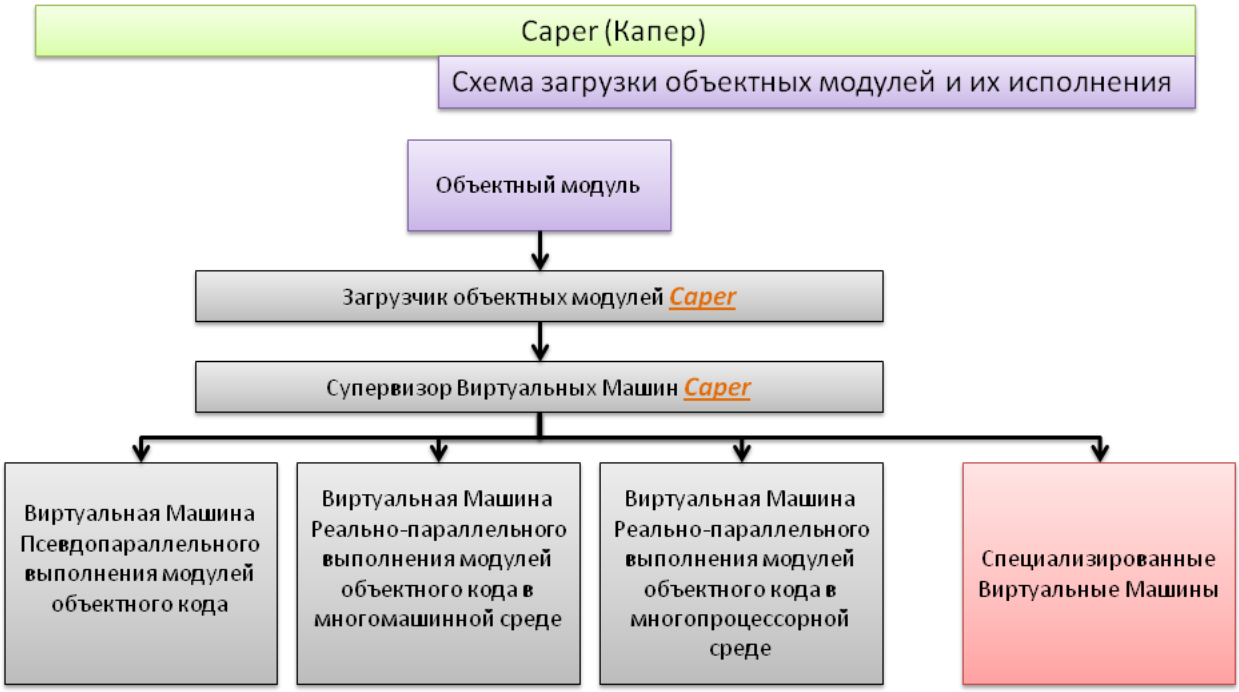
Презентация по основам языка **CAPER** (2016 год)

Программа является таким же ресурсом, как и все прочие атрибуты вычислительной установки и ее операционной среды, а потому ее структурное (организационное) состояние должно быть адекватным целям задачи, а самое важное текущим потребностям вычислительного процесса.

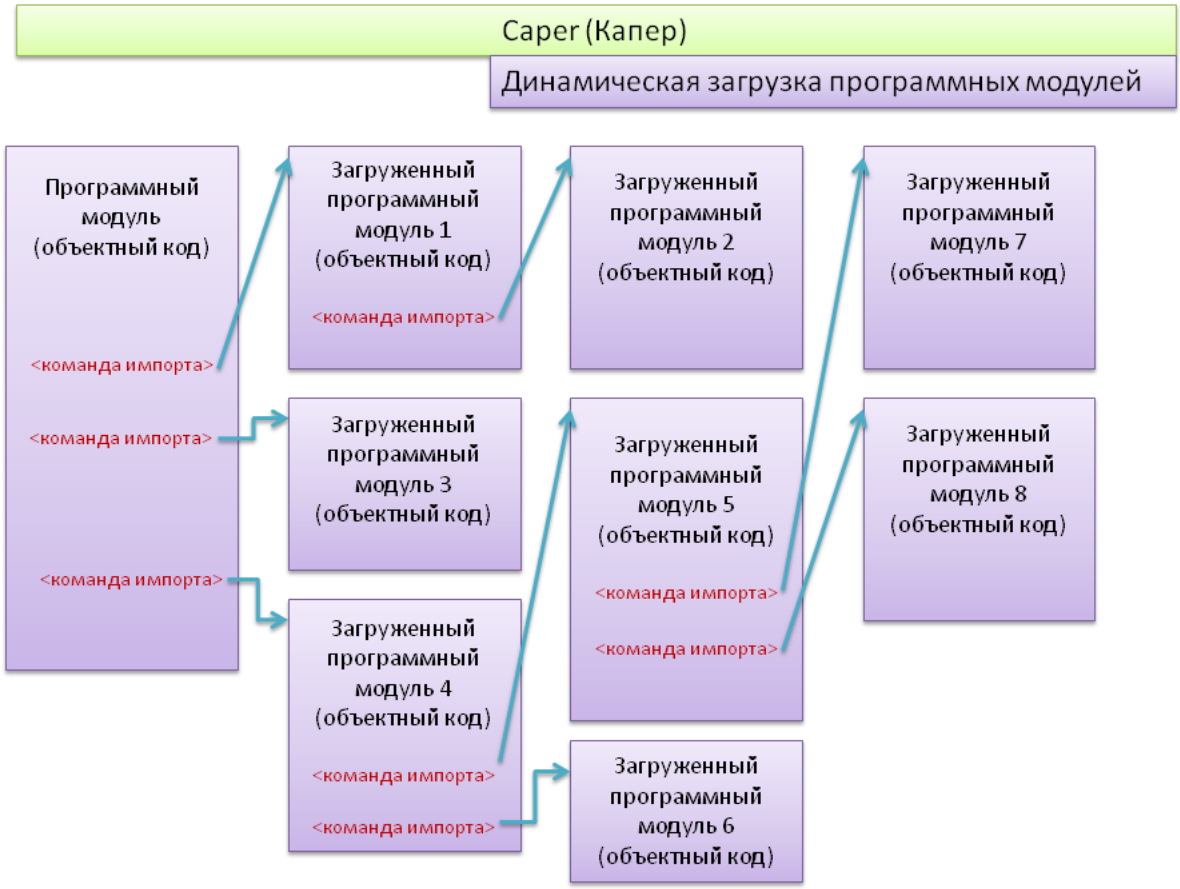


Реорганизация программ может осуществляться удалением собственных компонент, их изменением или же привнесением новых компонент извне. Внешние компоненты могут быть фрагментами исходного текста или объектными модулями. Т.е. речь идет о реструктуризации программы самой программой с целью создания такой ее структуры, которая отвечала бы текущим, сложившимся в процессе вычисления, потребностям решаемой задачи.

Следующим положением языка **CAPER** является параллельное исполнение структурных элементов программы синхронным или асинхронным способом, с квантованием времени, на множестве исполняющих установок - вычислительных машинах или на процессорах в многопроцессорной системе. В комплексе с предыдущими возможностями программирование в **CAPER** приобретает особые свойства по транспортированию фрагментов программы на различные вычислительные установки и их реализация.

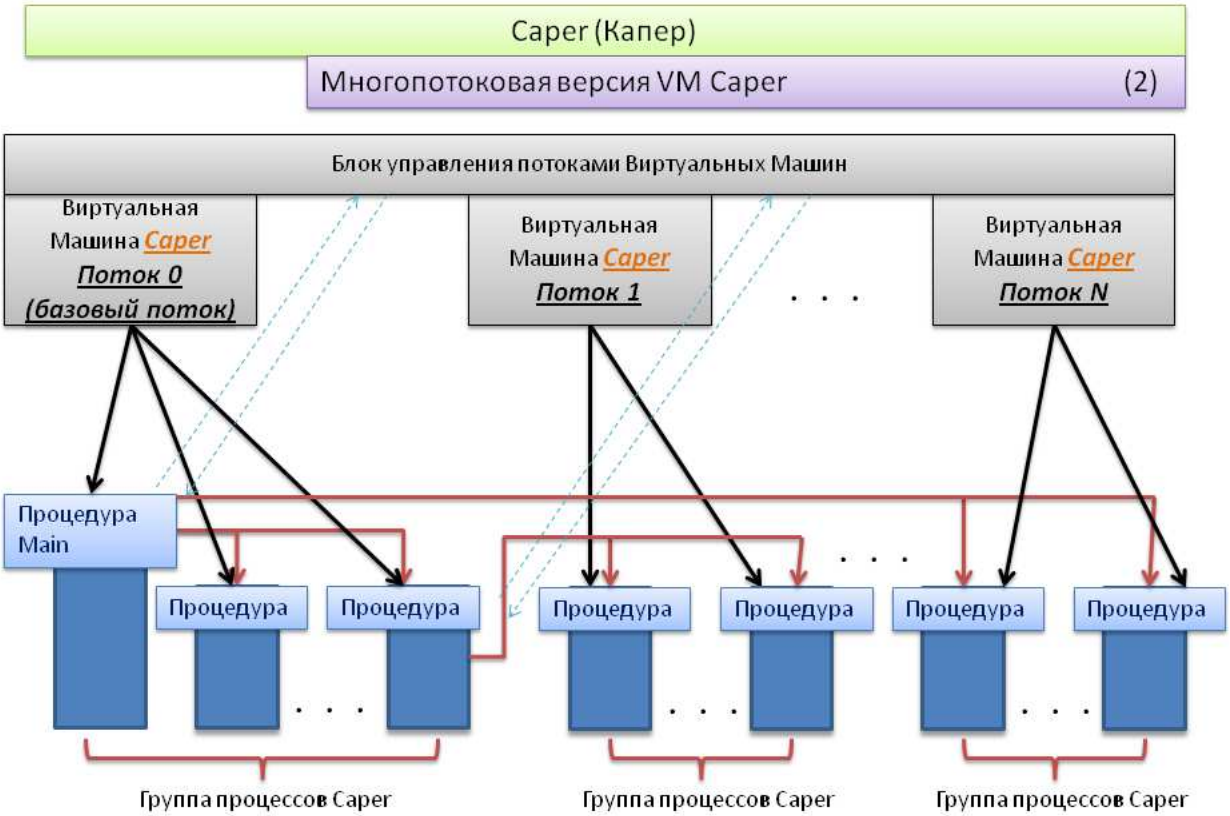


Наконец, последнее принципиальное свойство, реализованное в языке, относится к проблеме программирования реакций на асинхронные события, происходящие как "внутри" программы, так и "снаружи". Неэффективность большинства существующих практических языков достаточно очевидна - как правило, программисту приходится организовывать довольно рутинные процедуры отслеживания событий, которые загромождают программу, и часто весьма искусственны и субъективны, - в то время как было бы желательным и эффективным описывать множество событий и связанных с ними процедур, немедленно исполняемых при возникновении таких событий.



Кроме перечисленных основных положений подход, принятый в определении структуры программы, ее переменных и принципы ее исполнения позволяют эффективно программировать рекурсивные процедуры, непосредственно, без затрат по программированию описывать реентерабельные рутины (routine). Столь же эффективно в **CAPER**е решается проблема обработки

ошибок, происходящих в процессе вычисления: допускается анализ и исправление команд, целых фрагментов программы, приведших к ошибке.

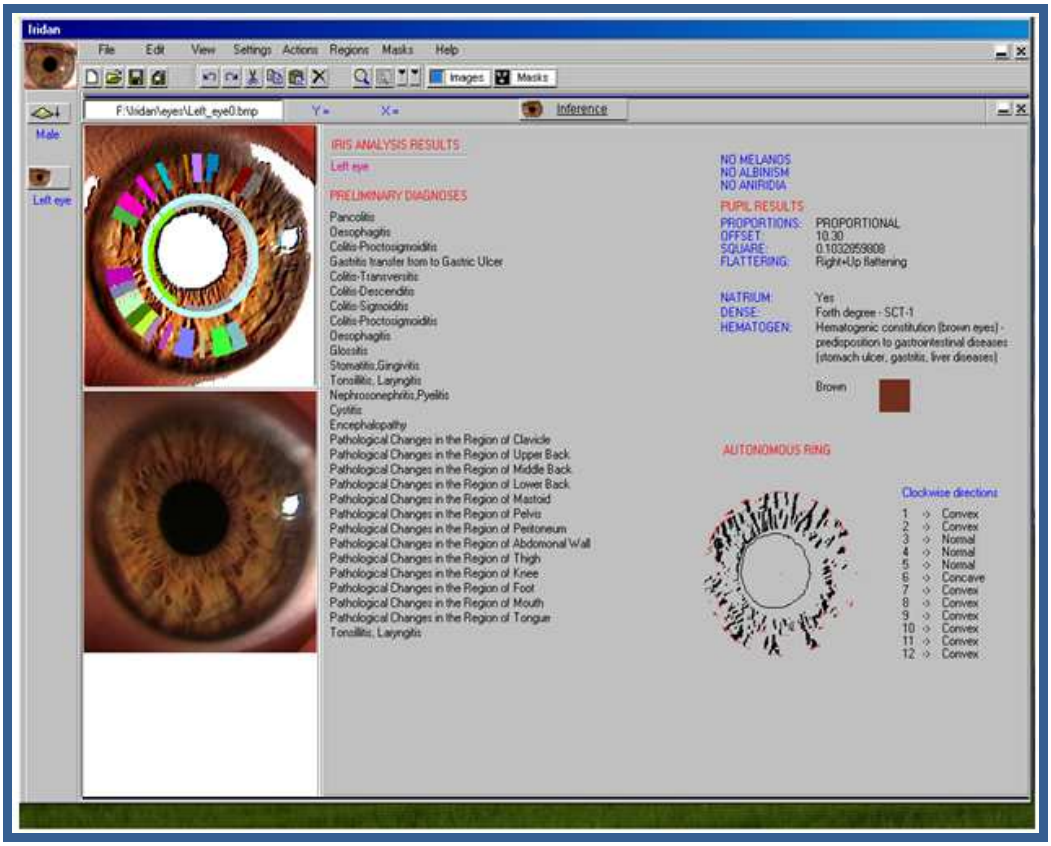


CAPER имеет собственный компилятор, а также реализована собственная виртуальная машина поддержки параллельного исполнения программ.

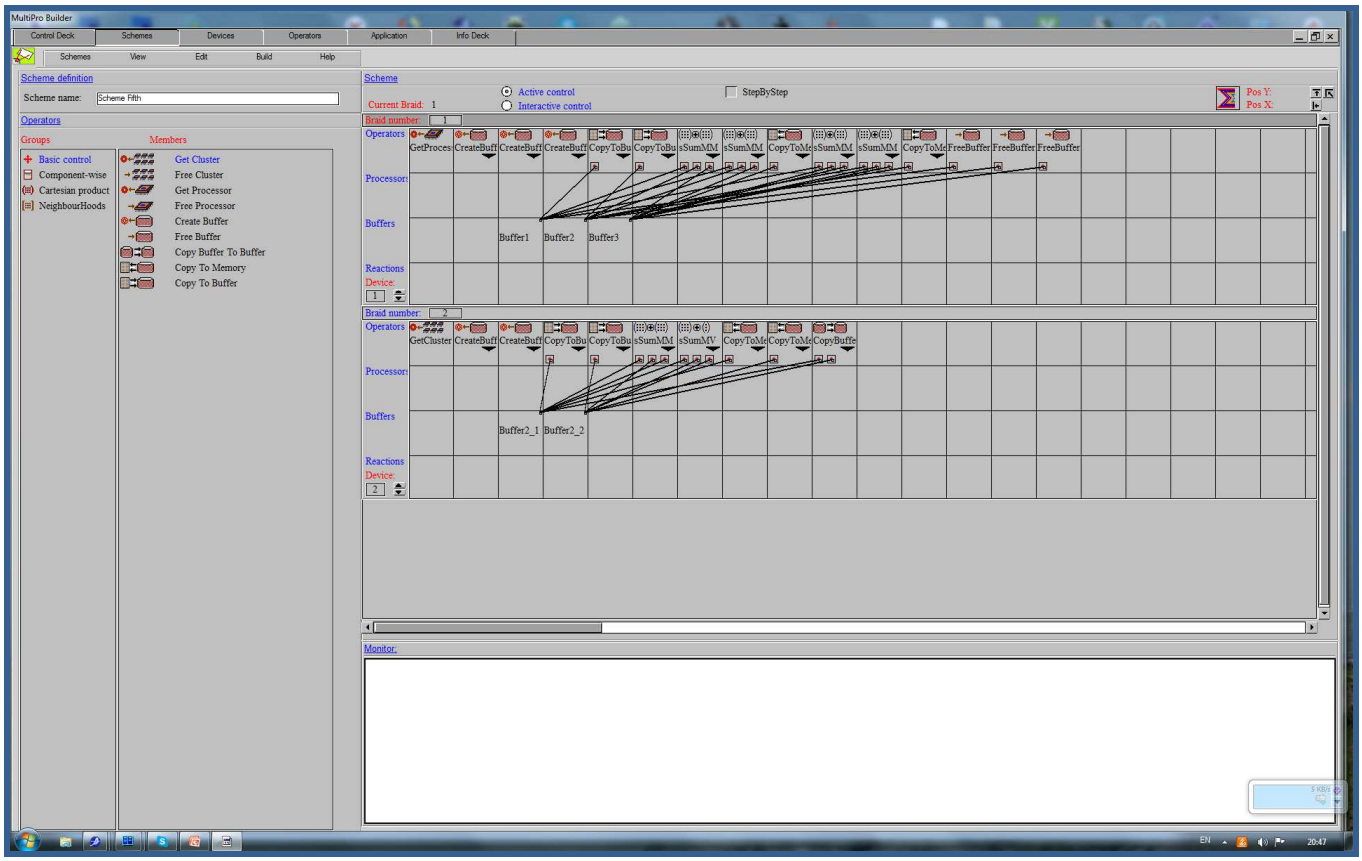
Реализация приложений на языке CAPER

На языке CAPER реализован значительный набор приложений в разных областях деятельности, где задачи имеют ярко выраженный параллелизм, например некоторые из них:

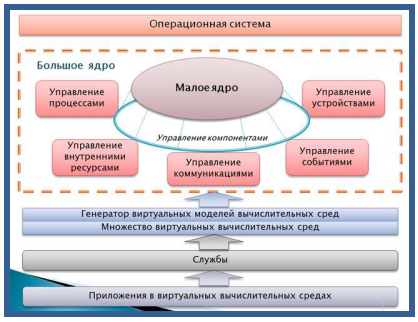
- *Распознавание образов с большим количеством областей интереса в параллельной обработке, например параллельное распознание более 1500 диагностических зон радужной оболочки глаза;*



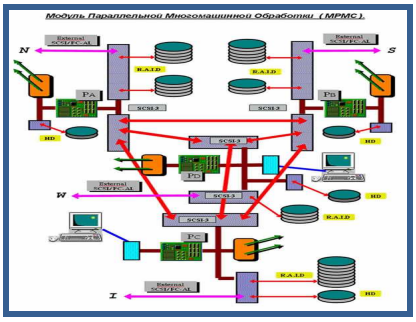
- *Симуляция мультипроцессорных систем (до 400-500 тыс. процессирующих элементов и более) для моделирования электронных схем – четырехкратный выигрыш по скорости симуляции **;*
- *Программы информационно-аналитического поиска на базе алгоритмов параллельной многомашинной обработки и поступлением в реальном времени результатов поиска;*
- *Аналитическая банковская система финансового состояния банков реального времени с обновлением актуальной графической информации каждые 30 сек. на мониторах специалистов;*
- *CAD подготовки программ для многопроцессорных устройств на базе многопроцессорных видеокарт AMD/ NVideo, с помощью которого создается архитектура многопоточных многопроцессорных (кластерных) устройств для обработки изображений или параллельных вычислений.*



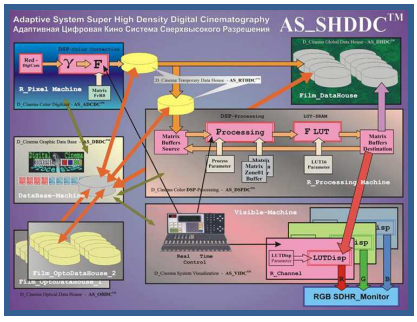
Язык CAPER и его методологическая база стали основой создания ряда проектов:



[Операционная Система Реального Времени **OS-RTRU**]



[МногоМашинное Параллельное Процессирование (обработка)]



[МногоМашинные Системы Сверхвысокого Разрешения]

**** Сопоставление скоростей симуляции СБИС на языках CAPER и SystemC**

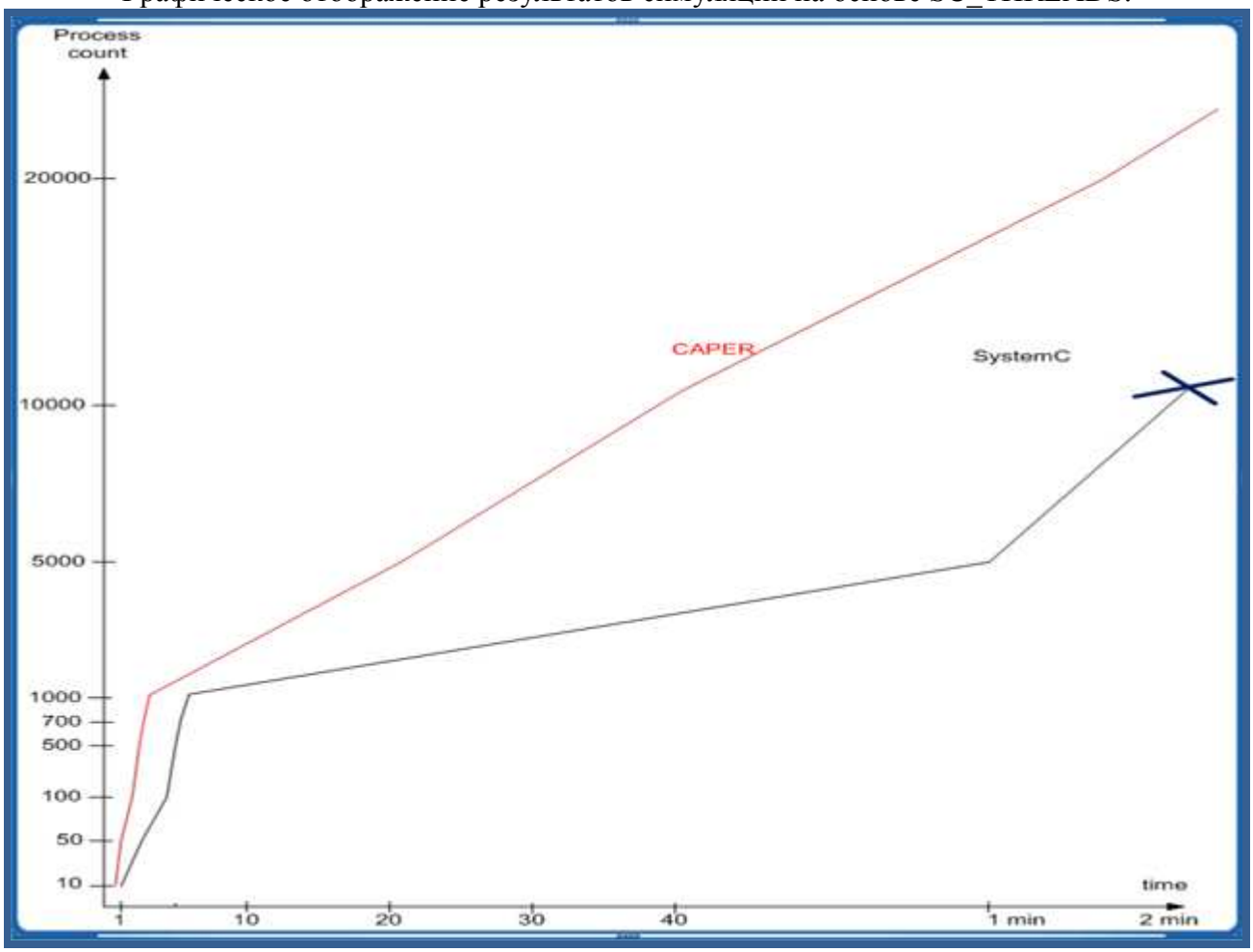
С полным текстом описания моделирования СБИС с помощью языка параллельного программирования CAPER можно познакомиться в **PDF** документе:

Язык параллельного программирования CAPER как средство для эффективного архитектурного моделирования современных СБИС.

Результаты симуляции СБИС на языках **CAPER4** и **SystemC** в табличном формате:

Количество параллельных процессов	SystemC Время симуляции в секундах	CAPER Время симуляции в секундах
10	1,145s	0,849s
50	2,656s	1,67s
100	3,99s	2,004s
500	4,65s	2,135s
700	5,04s	2,407s
1000	5,416s	3,006s
5000	1min, 1s	20,91s
10000	2min	40,7 s
20000	Hang of simulation	2 min

Графическое отображение результатов симуляции на основе SC_THREADS.



* *Крестиком обозначена **точка аварийного завершения** симулятора на SystemC.*

Как видно из графика, при увеличении количества параллельных процессов скорость симуляции на SystemC резко падает, а после 10000 параллельных процессов программа симуляции в SystemC завершается аварийно, в то время как программа на CAPER продолжает симуляцию.

Количество процессов доводилось до 200000, практика показывает, что предел определен архитектурными размерами оперативной памяти компьютера.

*** Основные публикации по языку параллельного программирования CAPER**

1. Вартанов С.Р. Методы конструирования и сопровождения данных средствами языка параллельного программирования Caper. Труды 4-ой международной конференции “Параллельные вычисления и задачи управления”. PACO-2008, Москва, 2008.
2. Вартанов С.Р. Средства динамической компоновки программ и данных в языке Caper. Годичная научная конференция Российско-Армянского (Славянского) Университета, Ереван, 2007.
3. Vartanov S.R. A Mass Parallel Starts In Parallel Programming Language Caper. Информационные технологии и управление. Том 4-1, Ереван, 2006.
4. Вартанов С.Р. Основания к концепции мультиязыков. Информационные технологии и управление. Том 4-3, Ереван, 2005.
5. Vartanov S.R. Parallel Programming Methods in Caper Language and its Application in Image Processing. SCI2002/ISAS2002, Orlando, USA, 2002, vol. XI.
6. Vartanov S.R. On Parallel Programming Language Caper. Lect. Notes in Computer Sci., HCPN-2001, 565-568.
7. Vartanov S.R. Parallel Programming in Caper. Mathematical Questions of Cybernetics and Computing Technique. Vol.22, Yerevan, 2001.
8. Вартанов С.Р. Язык программирования CAPER. Препринт 97-5. Национальная Академия Наук Украины, Институт Кибернетики им. Глушкова. Киев, 1997.