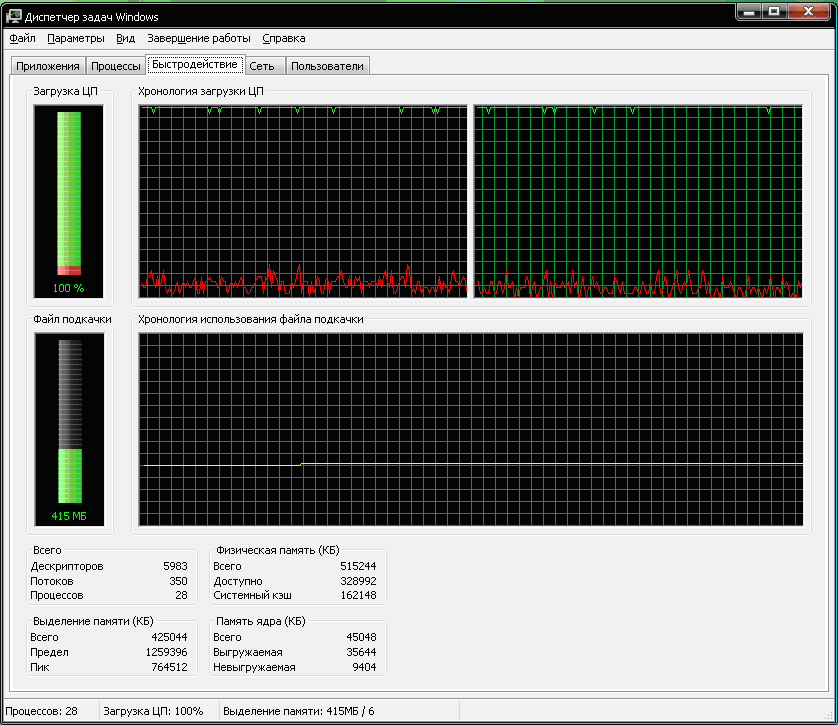
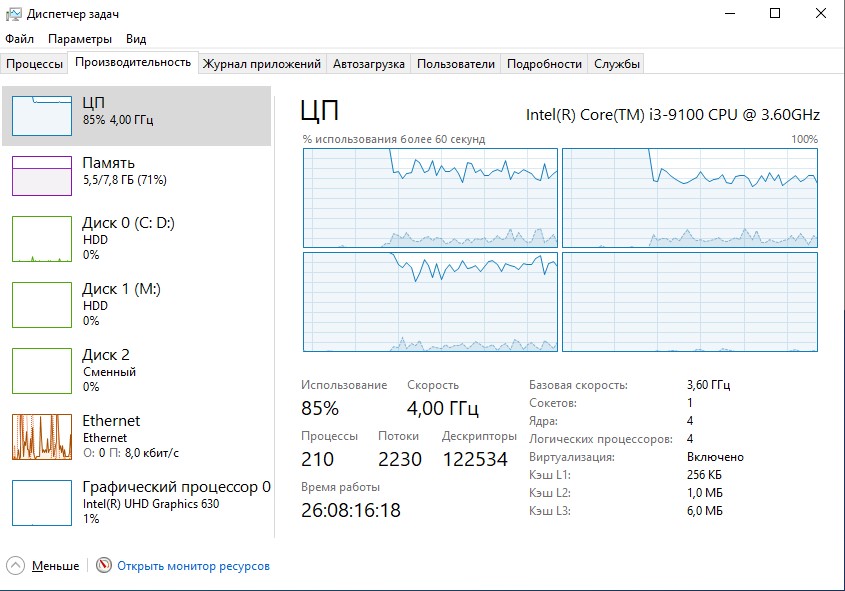
**ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК «МОНИТОРА РЕАЛЬНОГО БЫСТРОДЕЙСТВИЯ И ПРОФИЛИРОВЩИКА ПРОЦЕССОРОВ СЕРИИ X86\_64 SPEEDPROGRAF»**

Монитор реального быстродействия и профилировщик процессоров серии X86\_64 Speedprograf предназначен для мониторинга скорости выполнения операций процессоров серии X86\_64 Intel и AMD, как в режиме ядра (Ring 0 – Kernel Mode), так и в пользовательском режиме, на других уровнях привилегий (Ring 1,2,3 - User Mode) с минимально возможными затратами процессорного времени на его осуществление. Speedprograf профилирует всю систему целиком (операционную систему, гипервизор, приложения) и накапливает информацию на MSR (Model-Specific Registers) регистрах процессора.

Существующие мониторы производительности под названием быстродействие

и производительность

показывают обыкновенную загрузку процессора (% времени работы на временном промежутке).

Аналогов тахометра и спидометра в современных мониторах производительности нет.

Speedprograf исправляет эту несправедливость, показывая и загрузку процессора, и реальную скорость исполнения кода («тахометр» процессора в % пикового быстродействия).

В США существовала программа развития суперкомпьютеров DARPA HPCS (2001-2010 годы), в которой ставилась задача увеличения производительности суперкомпьютеров на реальных приложениях в 10-40 раз (не за счет увеличения количества ядер, а за счет реализации их пикового быстродействия). Задача увеличения реальной производительности не была выполнена, поскольку уперлась в «стену памяти» и перешла для решения в очередную программу США по суперкомпьютерам UHPCS (2011-2020 годы). Но если в 10-40 раз можно увеличить реальную производительность для суперкомпьютеров, то их реальный текущий КПД по быстродействию 2,5-10%. На сегодняшний день эта задача так и не решена. Список ТОП-500 суперкомпьютеров оказался не у дел при майнинге криптовалют – затраты на электричество превысили объем сгенерированной криптовалюты. Для генерации криптовалюты на смену им пришли ASIC майнеры, которые больше ни для чего не пригодны!

В эпоху развития искусственного интеллекта (далее – ИИ), «сапожник» оказался без сапог или, другими словами, суперкомпьютеры не оправдали надежд.

Мое предложение Сбербанку измерить реальное быстродействие его ЦОДов закончилось отказом в ранге заместителя Г.О.Грефа, хотя сам Г.О.Греф сетовал на медленное обучение ИИ Сбербанка - 20% в год. Он может и не знает, что этот процесс можно кардинально ускорить.

Ряд региональных органов власти испытывают проблемы с ЦОДами не из-за их недостаточной мощности, а из-за низкой эффективности их использования.

По мнению автора, существует проблема исполнения кода в процессоре, независимо от того, где он установлен: в суперкомпьютере, центре обработке данных, ПК, гаджете, станке. Это задача размерности 1х1, т.е. 1 процессор на 1 программу! Суперкомпьютер работает медленно потому, что на каждом его процессоре исполняется медленно соответствующая программа. Решив проблему исполнения кода единичного процессора, мы сможем увеличить мощность любых вычислительных систем на порядки.

**Проблема исполнения кода на единичном процессоре состоит из следующего:**

1. Низкая скорость исполнения кода ввиду того, что программный код не учитывает иерархическое построение памяти процессора, код и данные не синхронизированы, код однопроходной (неповторяемый или слабо повторяемый).
2. Однопроходной код перегружает шину памяти чтением программного кода для исполнения.
3. В многоядерном сокете при смене контекста нить (не привязана к ядру) при каждом подключении может менять ядро, что приводит к неоправданному повторному заполнению кэш-памяти 1-го и 2- го уровня из кэша 3-го уровня или из памяти? В результате однонитевое приложение на многоядерном процессоре исполняется медленнее, чем на одноядерном!
4. Большие накладные расходы при исполнении кода, которые существенно превышают долю кода, который исполняет целевую функцию программно-аппаратного комплекса и составляют от 66% до 99% (Эту тему автор впервые поднял в материалах международной конференции «Наука и будущее: идеи, которые изменят мир», 14-16 апреля 2004 года, Москва, ГГМ им. В.И.Вернадского РАН, тезисы доклада Суворова Д.А. «Роль организации вычислений в реализации потенциала современных компьютеров»).

Продуктивность программного кода – это доля кода, которая служит выполнению целевой функции, исполняемой программно-аппаратным комплексом. Продуктивность программного кода вычислительных систем 2 поколения, написанных в машинных кодах, составляла 33%, продуктивность программного кода вычислительных систем 3-го поколения (IBM 360/370, ЕС ЭВМ ряд 1/2), написанных на языках высокого уровня (ALGOL68, FORTRAN, PL1) со стековой организацией памяти вызовов, составляла 25%, продуктивность современного кода систем программирования с ООП (объектно-ориентированное программирование) и стековой организацией памяти составляет 10-20%, продуктивность кода ядра современной операционной системы и гипервизоров составляет 2,5-5%, продуктивность Java -апплетов может быть меньше 1%. Снижение продуктивности приводит к увеличению длины кода при выполнении одной и той же работы! Остальная часть кода, кроме целевой функции, называется накладными расходами. В современных компьютерах они превышают 80-90% и без специальных усилий это приводит к быстрому исчерпанию основного ресурса – процессорного времени.

Специальная организация вычислений позволяет повысить продуктивность кода на этапе исполнения и соответственно повысить производительность, не меняя основного программного обеспечения. Автор располагает актом о промышленной реализации в 4-х АСУ реального времени 2 и 3 поколения, с эффектом от 2-х до 4-х раз, именно за счет повышения продуктивности программного кода.

Speedprograf демонстрирует те же цифры реализации пикового быстродействия на задачах общего назначения на процессорах ПК общего пользования, что и в суперкомпьютерах. В User Mode код исполняется со скоростью 5-15%, а в Kernel Mode - 1-3% (со скоростью памяти и со скоростью перегруженной памяти)!

Вы бы стали ездить на машине, у которой в паспорте написана максимальная скорость 250 км/час, а она ездит не быстрее 7,5 км/час? Наверное, нет!

На старой странице сайта [Speedprograf](https://spgeedprogaf.mirbb.com/) приведены данные повышения производительности технологии гипертрейдинга на первых процессорах NetBurst –Intel Pentium 4 (ядро Willamette): «Например, технология гипертрейдинга (HT) по данным рекламных компаний увеличивает производительность на 25-40%. Мне удалось получить ускорение на двух процессах на 27% при скорости работы процессора в 2,5%. Фактически ускорение произошло на 2,5\*0,27=0,675% в терминах пиковой производительности. Так все-таки на 27% или на 0,675%? Разница в 40 раз? Резерв производительности современных ПК при объективном измерении и соответствующих доработках программного обеспечения измеряется порядками».

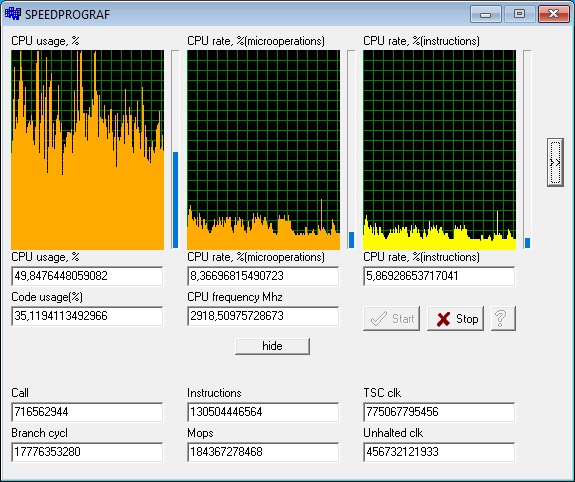
Перед началом профилирования Speedprograf программирует MSR регистры на измерение необходимых характеристик через драйвер msr.sys, который выполняет в режиме ядра привилегированные команды, после окончания профилирования (или периодически) в течение работы Speedprograf периодически считывает значения запрограммированных регистров, также через драйвер msr.sys.

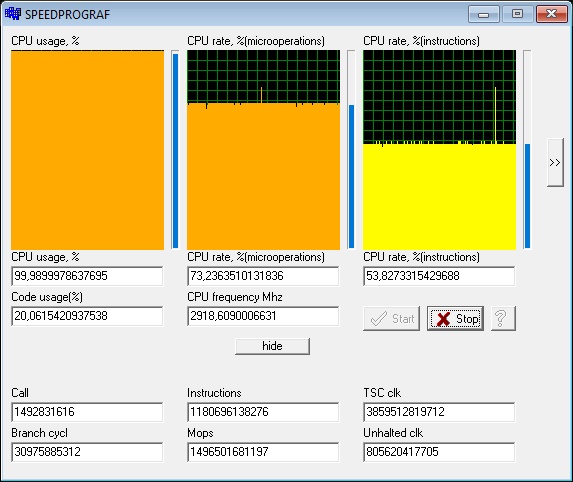
В наиболее полнофункциональной версии (для процессоров Intel Pentium IV – это наиболее полный профилировщик для процессоров X86\_64 серии, а не устаревший, который одновременно может регистрировать до 16 программируемых параметров + TSC) Speedprograf выводит следующие параметры:

1. Процент реализации пикового быстродействия в инструкциях (CPU rate instruction).
2. Загрузка процессора (CPU usage).
3. Частота работы процессора (CPU frequency Mhz).
4. Количество исполненных инструкций (Instructions).
5. Количество тактов процессора, прошедших с начала профилирования (TSС clk).
6. Количество тактов процессора, когда он не простаивал (Unhalted clk).
7. Количество вызовов (команд Call).
8. Количество переходов (Branch cycl).
9. Доля команд процессора, обращающаяся к данным (Code usage).
10. Процент реализации пикового быстродействия в микрооперациях (CPU rate microoperations).
11. Количество исполненных микроопераций (Mops).

Версия профилировщика для AMD поддерживает до 8 первых параметров (из 11) в предыдущем списке (в AMD 4 программируемых регистра + TSC).

Версия профилировщика для Core XX поддерживает только 4 первых параметра (в Core XX 2 программируемых регистра + TSC).

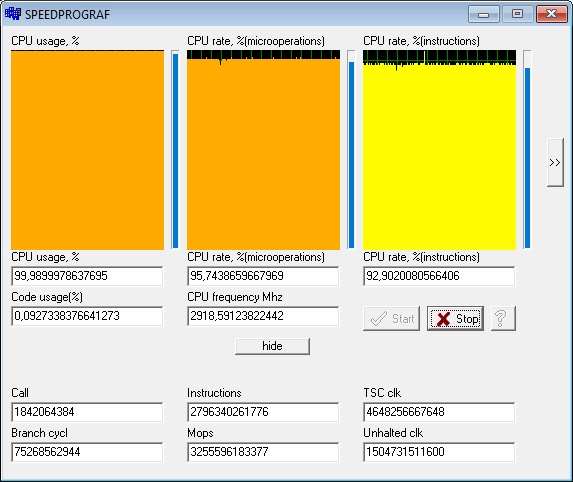
Скорость архивации WinRar выглядит следующим образом:

Скорость на уровне 6% от пиковой, при этом доля команд, обрабатывающая данные Code usage, составляет 35,1%.

.

А это скорость теста, к которой мы должны стремиться после доработок операционных систем! Здесь пиковое быстродействие 54%, а Code usage - 20%.

Скорость в 9 раз больше, чем на предыдущем рисунке. Есть к чему стремиться!



На данном рисунке 92,9% пикового быстродействия! Если сравнить данное быстродействие с быстродействием 6%, получим разницу в 15 раз!

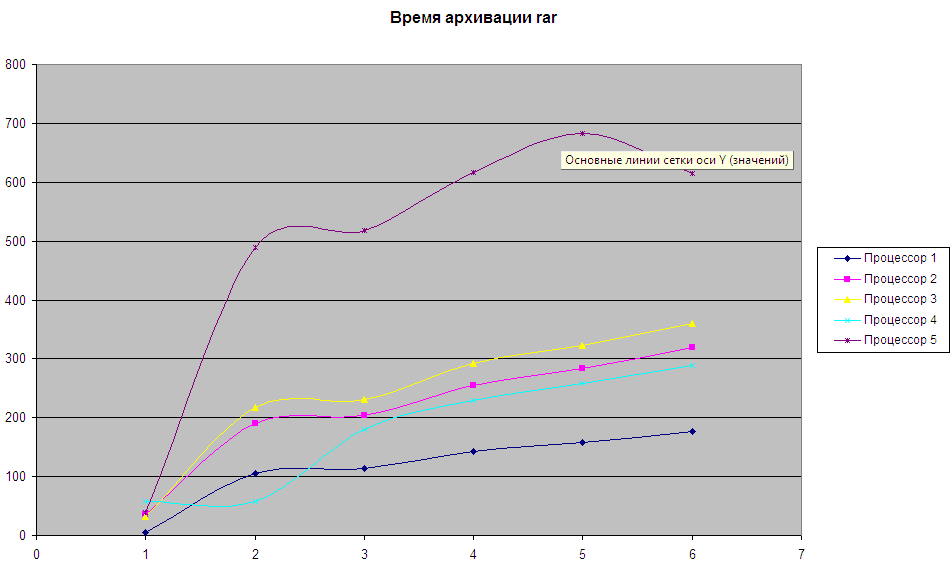
Однако здесь используется нечестный прием тестировщиков. Тест построен на пустых операциях (NOP), где Code usage всего 0,09%. Процессор не обрабатывает данные, с обработкой данных он не может работать с такой скоростью.

Полнофункциональный профилировщик позволяет объективно оценивать реальное быстродействие!

Привожу пример исследования инструментом SPEEDPROGAF однонитевого приложения на 5-ти X86\_64 процессорах(1 - Intel Core2Duo E7400 2,8 Ghz (DDR2-800) , 2 – Intel CeleronD 430 1,8 Ghz (DDR2-800), 3 – AMD Turion 64 TL-50 1,6 Ghz (DDR2-667), 4 – Intel Pentium 4 3Ghz Presscott (DDR2-800), 5 - Intel Pentium 4 1,5Ghz Willamette (DDR-266). Это архивация WinRar  дистрибутива Microsoft Office размером 0,55 Гбайт. Состав команд, используемых процессором – целочисленная арифметика. Архивные файлы представляют собой файлы длиной 50-60% от исходного объема дистрибутива. Степень архивации обозначена цифрами: 1 - архив без сжатия, 2 – скоростная, 3 - быстрая, 4 – обычная, 5 – хорошая, 6 – максимальная. Результирующие архивы от работы разных процессоров совпадают по длине до байта, несмотря на различную длину исполненного кода! Что происходит?

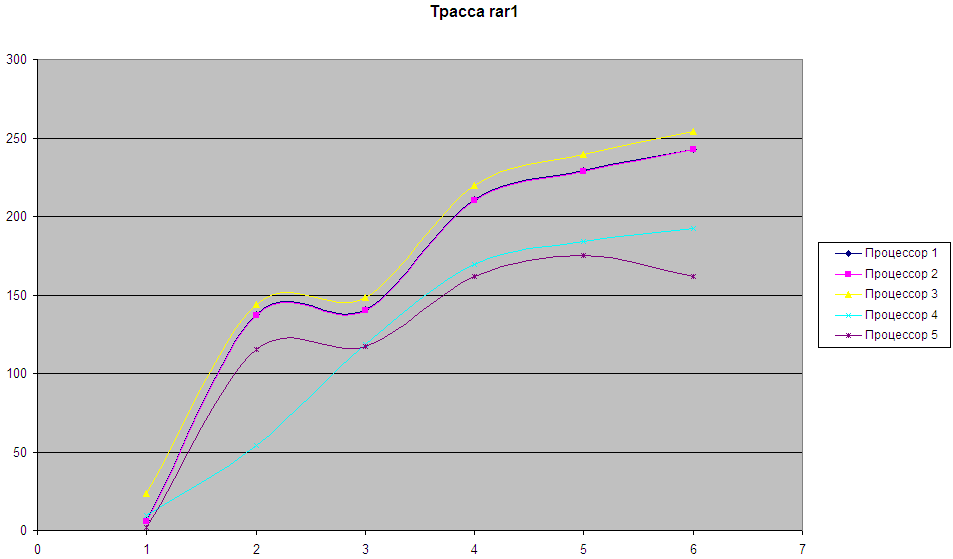
На графике, представленном ниже, показана скорость архивации в GIPS (в миллиардах инструкций в секунду) для различной степени архивации.

Видно, что процессор №1 исполняет код с наибольшей скоростью, процессор №5 – с наименьшей, а процессоры №2,3,4 – примерно с одинаковой.

На графике, представленном ниже, показано время архивации (в секундах).

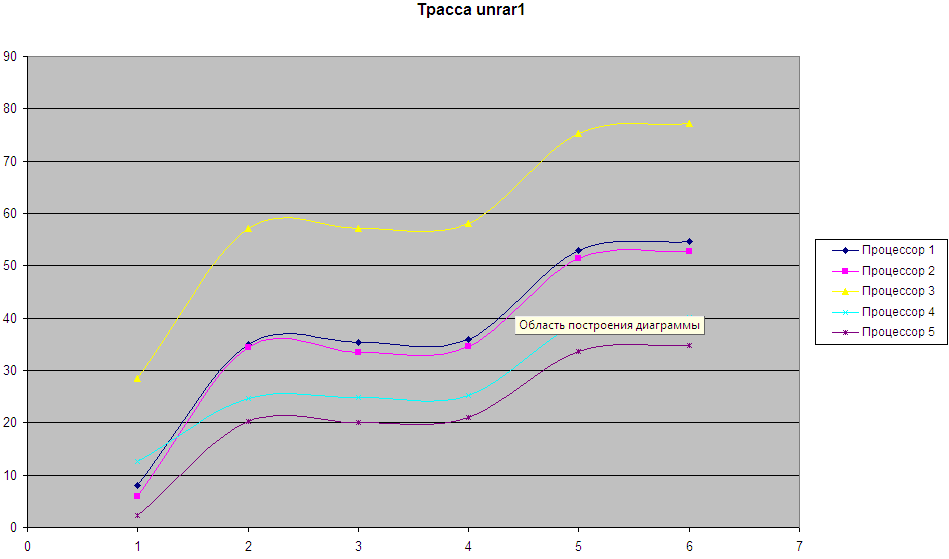
Как мы и ожидали, у процессора №1 минимальное время архивации, у процессора №5 – максимальное, но во 2-ом режиме архивации процессор №4 обогнал лидера производительности. В чем тут дело?

Приведем длину трассы (в миллиардах инструкций х86, исполненных по программе - Retired Instruction). График прилагается ниже.

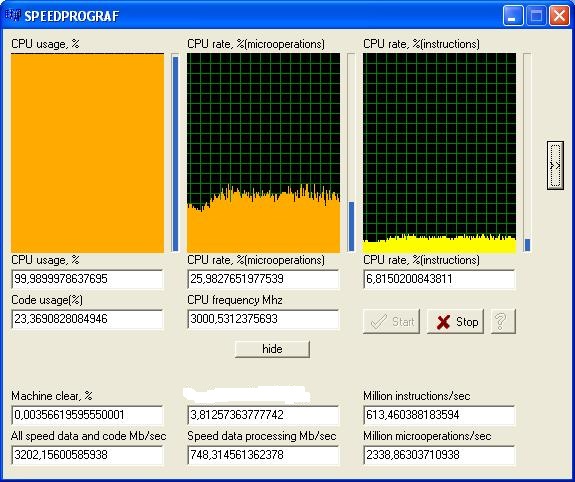


Вот теперь становится понятно, почему «медленный» процессор №4 обогнал процессор №1 с самой высокой производительностью! У него количество выполненных команд во 2-ом режиме архивации в 2 раза меньше, чем у остальных процессоров! А что, разве одну и ту же работу процессор может выполнить различными путями? Оказывается, может! Режимы архивации 3-6 тоже различаются для групп процессоров 4-5 и 1-2-3.

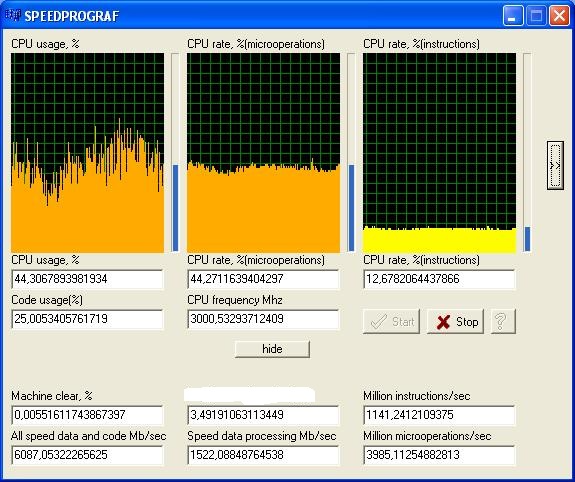
В подтверждение выше сказанного прошу рассмотреть следующий график разархивации WinRar и обратить внимание на то, что длина трассы разархивации на различных процессорах X86\_64 отличается в 2-3 раза.



SPEEDPROGRAF подтвердил наличие накладных расходов. Процессор с более высокой и реальной производительностью исполняет трассу другого порядка длиннее, а медленный - короче.

Пример исследование 2-го режима архивации с использованием программного средства объективного контроля SPEEDPROGRAF на процессоре Intel Pentium 4 3Ghz Presscott (DDR2-800) приведен ниже. На следующем рисунке приведены характеристики вычислительного процесса с выключенным режимом гипертрейдинга 

А на этом рисунке приведены характеристики вычислительного процесса с включенным режимом гипертрейдинга:



Т.е. «выдающиеся» способности на 2-м режиме архивации процессор Intel Pentium 4 3Ghz Presscott (DDR2-800) проявляет при включении режима гипертрейдига (2 медленных процессора, вместо 1 быстрого). Загрузка процессора снижается в 2 раза (трасса короче больше, чем в 2 раза), скорость обработки вырастает практически в 2 раза (с 6,81% до 12,68%), а время архивации сокращается в 4 раза (квадратичная зависимость).

Если в BIOS отключить режим гипертрейдинга, он как и все остальные процессоры из приведенного перечня выше, начинает исполнять на 2-м режиме архивации «длинную» трассу (111 миллиардов инструкций, вместо 54) и в 2 раза медленней.

SPEEDPROGRAF позволяет достоверно оценить правильные направления оптимизации ядра операционной системы: сокращение трассы исполнения (режим трассировки) и рост пиковой производительности.

Автор планирует опубликовать результаты исследования платформы X86\_64 с помощью инструмента объективного контроля SPEEDPROGRAF. Он позволил определить «узкие» места в архитектуре многоядерных процессоров: шина памяти, кэш L3, прерывания. Процессоры без L3 кэша работают намного быстрее. В 4-х ядерном процессоре 50% времени ядра простаивают в ожидании обработки прерывания, поскольку на 1 сокете одновременно прерывания обрабатывает 1 ядро. Сколько будут простаивать 12 логических ядер современных процессоров при обработке прерываний можно будет узнать из будущей публикации. Однонитевые приложения исполняются на многоядерных процессорах медленнее, чем на 1 ядре!

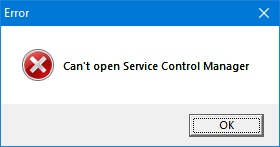
SPEEDPROGRAF позволяет достоверно оценить правильные направления оптимизации ядра операционной системы - сокращение длины «трассы» исполнения (количества выполненных команд по программе) и рост пиковой производительности, а также оценить полученный реальный эффект до и после доработок!

**Инструкция по установке SPEEDPROGRAF**

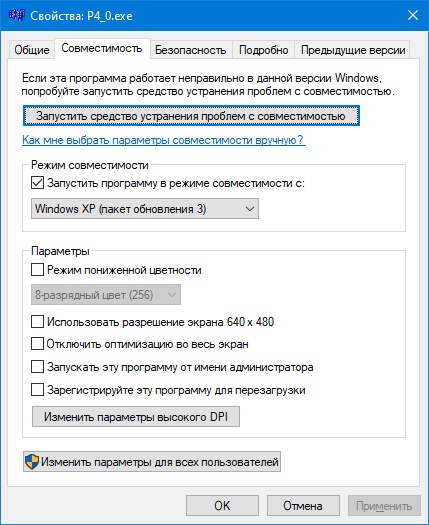
Монитор реального быстродействия и профилировщик процессоров серии X86\_64 Speedprograf поставляется в виде параметричесой версии: в виде \*.exe-файла и файла драйвера режима ядра msr.sys. Многоядерный коммерческий драйвер msr.sys запускается на любом ядре процессора сокета (сокетов).

Для установки SPEEDPROGRAF необходим:

1. Скопировать \*.exe-файл и файл драйвера режима ядра msr.sys в рабочую директорию.
2. Запустить исполняемый файл \*.exe.
3. Если выдается ошибка вида

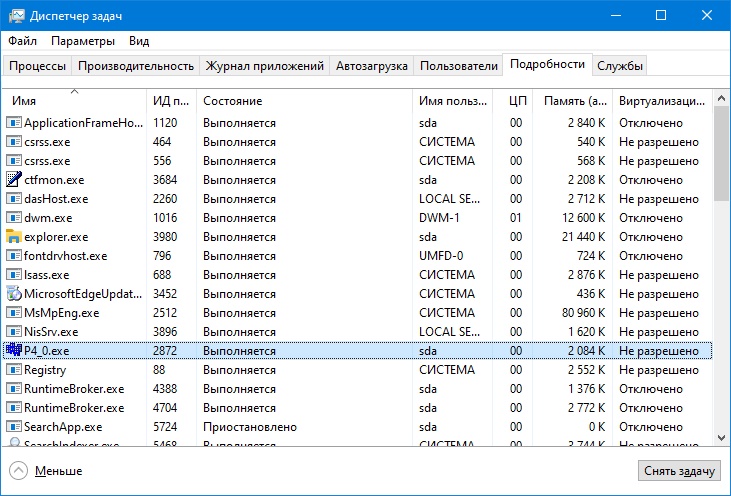


необходимо в «Свойствах» ярлыка исполняемого файла \*.exe

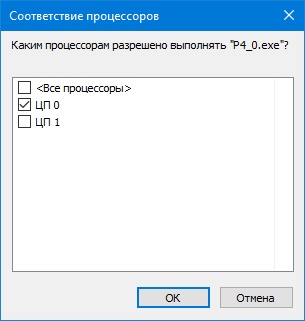


установить совместимость запуска с Windows XP(SP3).

1. Перед началом трассировки необходимо привязать экземпляр \*.exe файла к номеру ядра, который вы собираетесь профилировать. Он программирует и читает MSR регистры процессора, на котором запущен. Сначала выбрать экземпляр \*.exe-файла в Диспетчере задач,

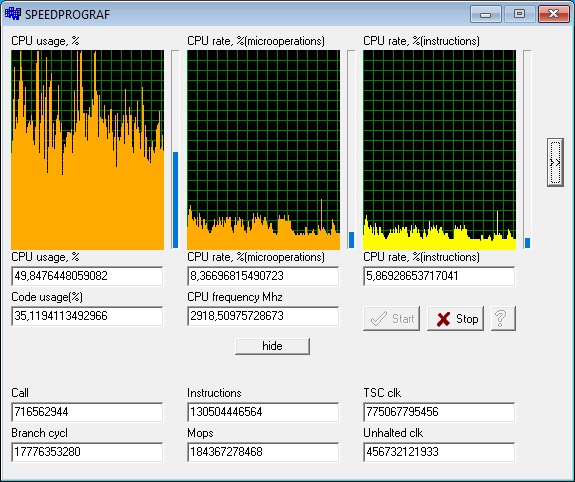
 ,

затем правой кнопкой выбрать «Задать сходство» и оставить галочку в

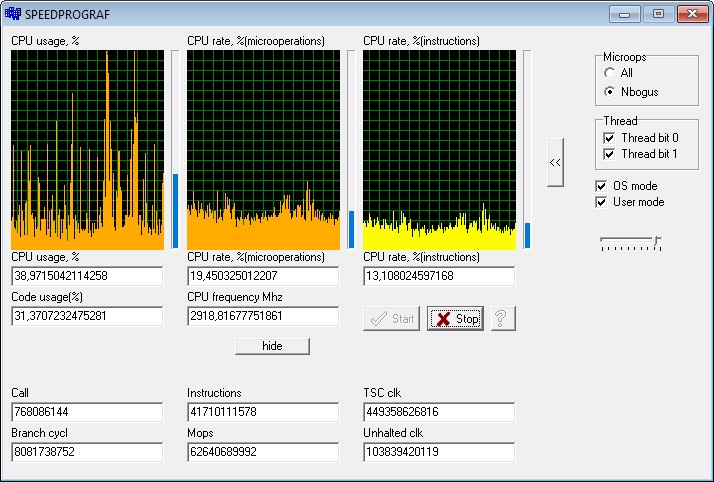


одном нужном процессоре.

1. Запускаем необходимые приложения для профилирования и нажимаем кнопку Start.



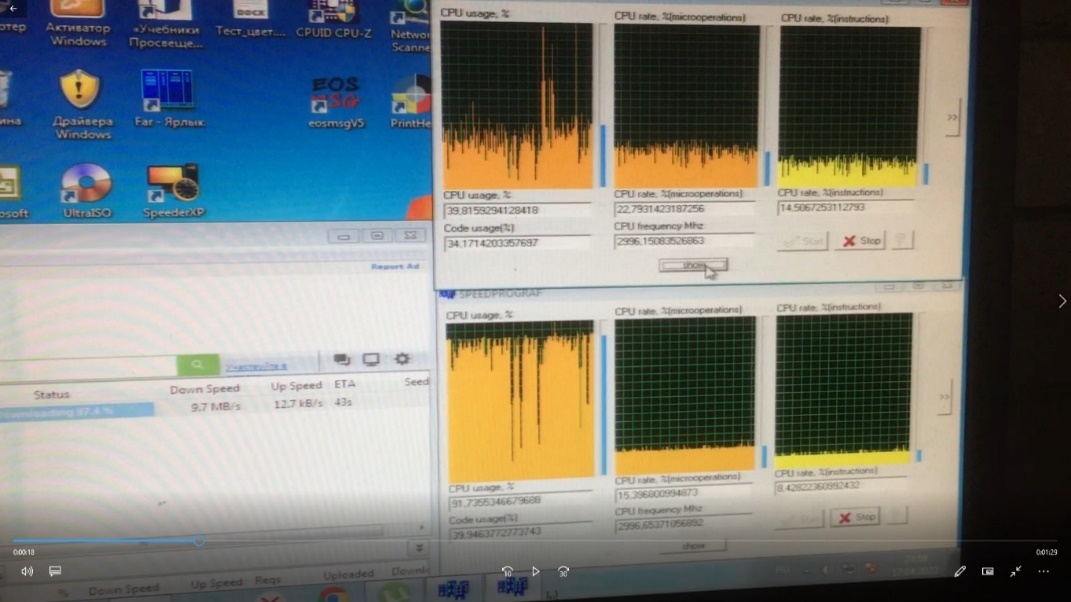
1. После завершения профилирования нажимаем кнопку Stop.
2. Если нажать на кнопку с правой стороны, то можно выбирать галочками режимы профилирования Os\_mode и User\_mode,



раздельно или вместе и регулировать темп выдачи информации на экран. Кнопка hide скрывает нижние 6 параметров при профилировании для уменьшения размера окна приложения.

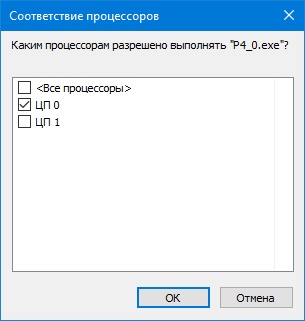
1. Если вы приобрели два (или более) \*.exe-файла с названиями например \*\_0.exe, \*\_1.exe, то пункт 4 нужно выполнить для всех запущенных профилировщиков (привязать каждый к своему ядру). Затем запускать приложения и стартовать все \*\_0.exe, \*\_1.exe файлы последовательно. После завершения профилирования останавливать их надо в том же порядке.

Два запущенных профилировщика выглядят следующим образом:



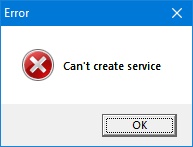
**Инструкция по эксплуатации SPEEDPROGRAF**

1. Если вы по какой-то причине не привязали \*.exe-файл к ядру и нажали клавишу Start, вам может выдаваться в цикле ошибка выполнения привилегированной команды в незащищенном режиме. Для этого необходимо, удерживая клавишу «esc», добиться остановки профилирования

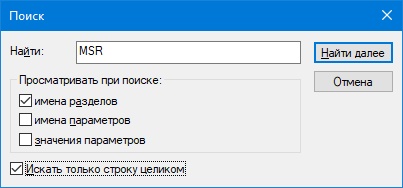


и привязать \*.exe-файл к нужному ядру.

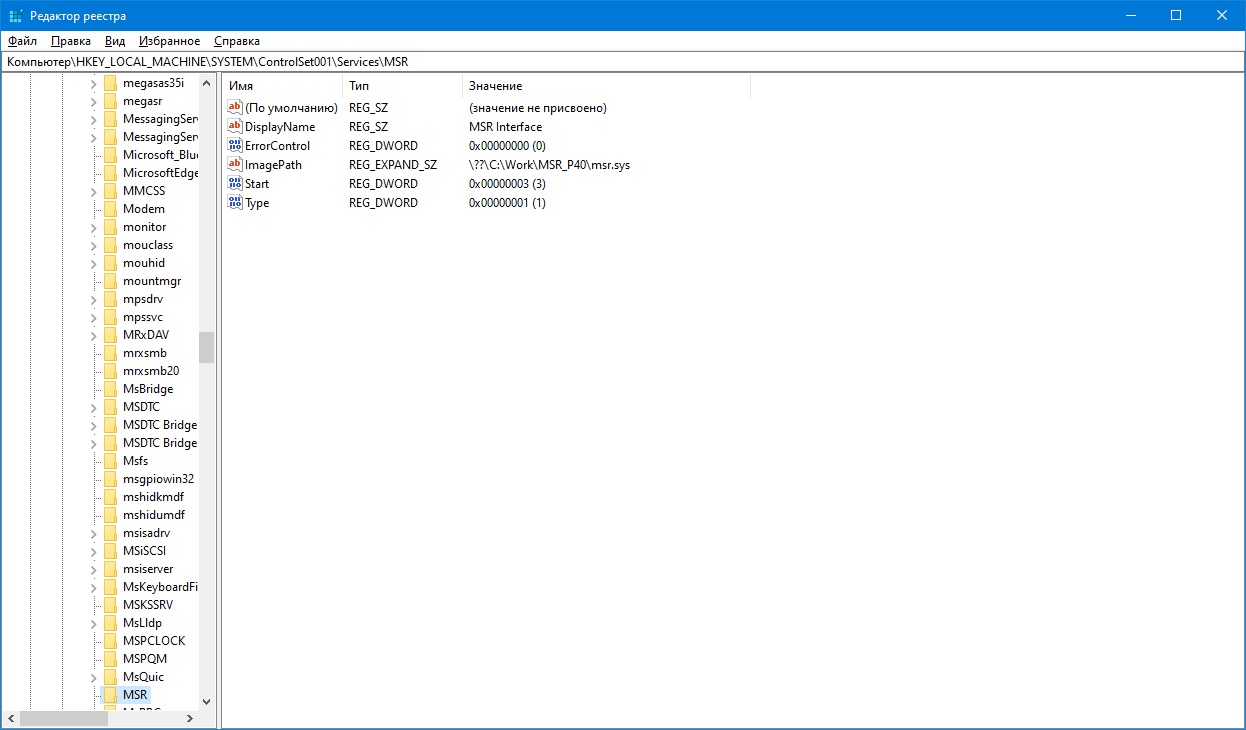
1. Если у вас по какой-то причине выключился ПК во время работы P4\_0.exe (аварийное завершение работы), при повторном запуске вам высветиться следующая ошибка:

.

Speedprograf при запуске создает запись в реестре Windows и удаляет ее при выходе из программы, поэтому нужно запустить программу regedit.exe,



найти строку (F3-повтор поиска) раздела MSR



и полностью удалить ее (если найдет несколько копий, удалить все). После этого обязательно перезагрузить ПК. Работоспособность Speedprograf восстановится.

1. Если вы запускали несколько экземпляров \*.exe-файлов и компьютер завис, в реестре Windows нужно найти разделы MSR, MSR1, MSR2, MSR3 и т.д. и соответственно удалить для восстановления работоспособности.

Сейчас в состоянии разработки находится мультиядерный драйвер режима ядра, который позволит опрашивать счетчики любого процессора с любого. Когда он будет отлажен, появится возможность опрашивать все ядра процессора с одного приложения без привязки к ядрам.

**ПРЕДЛОЖЕНИЕ ИНВЕСТОРУ**

Автор предлагает к реализации способ организации вычислений для реализации пикового быстродействия процессоров при исполнении низко-продуктивного программного кода, который повышает его продуктивность на этапе исполнения, в результате которого длина исполняемого кода сокращается, а скорость исполнения возрастает - проект диспетчера мультиконвейерной обработки (далее – диспетчер МКО) в ядре ОС c функцией автоматического повышения быстродействия. Имеющийся акт о промышленной реализации указанной технологии в 4-х АСУ реального времени может быть предоставлен заинтересованным лицам.

Speedprograf разработан специально для объективной оценки скорости исполнения кода и его длины (сколько команд истрачено на выполнение той или иной работы).

Практически для кода с продуктивностью 1/3-1/100 можно сократить длину (трассу) с 2,5 до 9 раз при 10-кратном повторении целевого кода на этапе исполнения, при этом скорость исполнения целевого кода при повторении из кэш-памяти увеличится до 3-х раз. График представлен ниже.

Суммарная загрузка шины памяти может уменьшиться в 4,5 раза. График представлен ниже.

Итоговый график по ускорению при 2-10 кратном повторении для 64-битных систем будет выглядеть следующим образом:

При использовании диспетчера МКО можно практически полностью восстановить пиковое быстродействие процессоров не только для режима пользователя, но и даже для режима ядра (увеличение производительности в 40 раз).

Диспетчер МКО не относится к средствам распараллеливания вычислений и может применяться на вычислительных системах с любой архитектурой, он работает на каждом процессоре вне зависимости от целей организации взаимодействия процессоров между собой, задаваемым внешним независимым управлением и реализует преимущества иерархии памяти процессора. Диспетчер МКО «разгоняет» каждый процессор независимо друг от друга только на принципе низко-продуктивности программного кода (только улучшая повторяемость).

Разработка опытного образца (макета) диспетчера МКО в ядре ОС с доказанными и повышенными потребительскими свойствами позволит запатентовать указанную технологию и в дальнейшем использовать для реализации на рынках всех сегментов вычислительных устройств.

Решение по диспетчеру МКО будет одинаковым как для 32-битных, так и для 64-битных ОС. Поэтому, на какой разрядности ОС (32 или 64) проводить доработки, не принципиально!

В дальнейшем полученные решения по диспетчеру МКО можно использовать на ПК, в ЦОДах, в суперкомпьютерах, после чего (или параллельно) и на гаджетах.

Что могу предложить партнеру:

Предлагаю на паритетных началах знания (ноу-хау) по технологии диспетчера МКО через заключение партнерского соглашения. В случае положительной реализации предлагаемой технологии рассчитываю на равноценное (50/50) участие в ее патентовании, реализации и дальнейшем использовании.

По договоренности с партнером публикация материалов использования Speedprograf в области низко-продуктивного программного кода может быть отложена до получения положительных результатов по разработке макета или действующего прототипа диспетчера МКО и регистрации его в органах патентной собственности.

**Сейчас на реализованной 32-битной технологии для X86\_64 платформы (при наличии 32-bit версии ОС) есть возможность на моей платформе проводить доработки 32-битных аналогов операционных систем (сред) с целью их дальнейшего переноса на 64-битные прототипы (портации) с использованием собственного средства объективного контроля (speedprograf) , либо на любой платформе и в любой среде с предоставлением соответствующих имеющихся средств объективного контроля вычислительного процесса.**

**Что жду от партнера:**

* заказ на доработку ядра промышленной (коммерческой) операционной системы на любой платформе и в любой среде с соответствующими средствами объективного контроля вычислительного процесса, аналогичному speedprograf, с целью автоматической реализации пикового быстродействия процессоров или на разработку макета диспетчера МКО ядра ОС, с целью подтверждения его высоких потребительских свойств;
* компетенции в анализе состава и построения выбранной операционной системы на уровне исходного кода для оказания консультаций при проведении необходимых работ по модернизации до уровня диспетчера МКО.

Очевидно, партнером для разработки макета диспетчера МКО может быть российский производитель ПО в целях импортозамещения. Российские средства ЭВТ на порядок хуже по быстродействию Intel и AMD, например «Таволга (технология MIPS)», «Эльбрус» и т.п., и технология реализации пикового быстродействия позволит им встать на один уровень или даже обогнать ЭВТ, разработанную на базе процессоров Intel и AMD и других импортных производителей.

Однако мне неизвестно, поддерживают ли отечественные процессоры технологию чтения MSR-регистров. Поэтому макет диспетчера МКО может быть создан и отлажен и на X86\_64 платформе на 32-bit версии ОС, а затем портирован на необходимую платформу и разрядность (64) ЭВТ российского производителя.

**Разработка трассировщика и монитора производительности SPEEDPROGRAF под 64 бит или под другие платформы при необходимости может быть отдельным направлением деятельности или под конкретную архитектуру, с целью их дальнейшей реализации на рынке для широкого круга пользователей** и при возникшей необходимости по доработкам.

**Информация по проекту расположена на сайте** [**https://spgraf.ru**](https://spgraf.ru).

Открытые публикации автора:

Cуворов Д.А. Роль организации вычислений в реализации потенциала современных компьютеров//Матер. междунар. конф. «Наука и наше будущее: идеи, которые изменят мир»/ ГГМ им. В.И.Вернадского РАН. –М., 2004. –С. 186-187 // [Электронный ресурс]. – Режим доступа: URL: <https://spgraf.ru/Publications/136_01.pdf>

Суворов Д.А. Измерение текущего быстродействия процессоров и качества программ. Способы оценки и повышения реальной производительности. Журнал «Открытое образования» №9/2006 // [Электронный ресурс]. – Режим доступа: URL: <https://spgraf.ru/Publications/paper.pdf>

Есть список закрытых публикаций (35), в них 3 авторских свидетельства, которые при необходимости могут быть предоставлены.

E-mail для обращений: [speedprograf@mail.ru](mailto:speedprograf@mail.ru), Суворов Дмитрий.