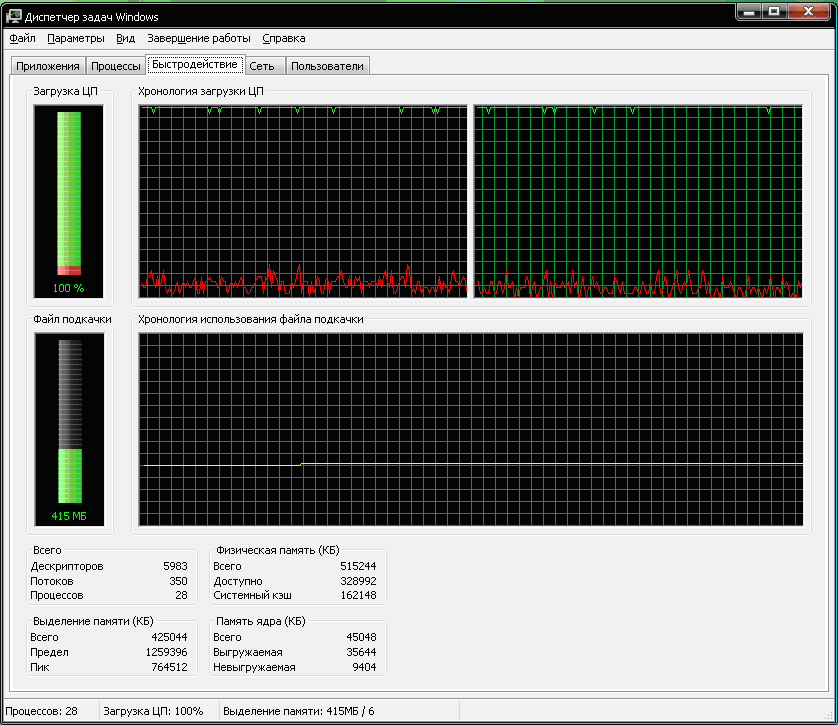
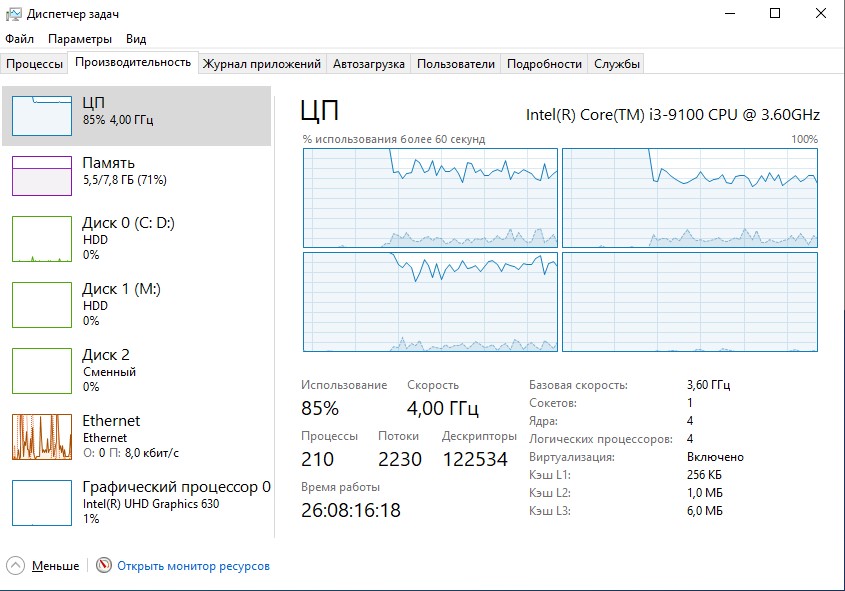
**ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК «МОНИТОРА РЕАЛЬНОГО БЫСТРОДЕЙСТВИЯ И ПРОФИЛИРОВЩИКА ПРОЦЕССОРОВ СЕРИИ X86\_64 SPEEDPROGRAF»**

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».

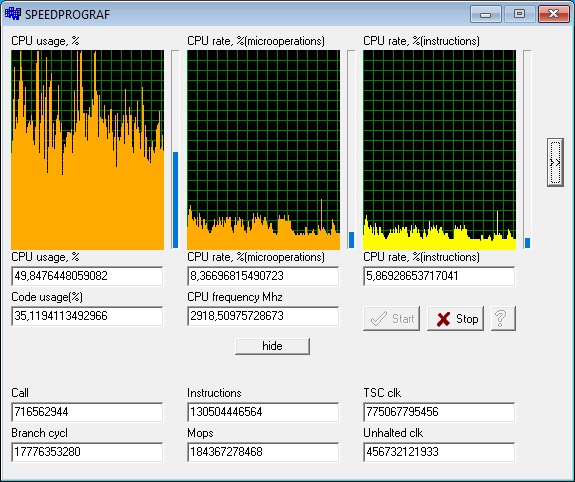
Перед началом профилирования Speedprograf программирует MSR регистры на измерение необходимых характеристик через драйвер msr.sys, который выполняет в режиме ядра привилегированные команды, после окончания профилирования (или периодически) в течение работы Speedprograf периодически считывает значения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

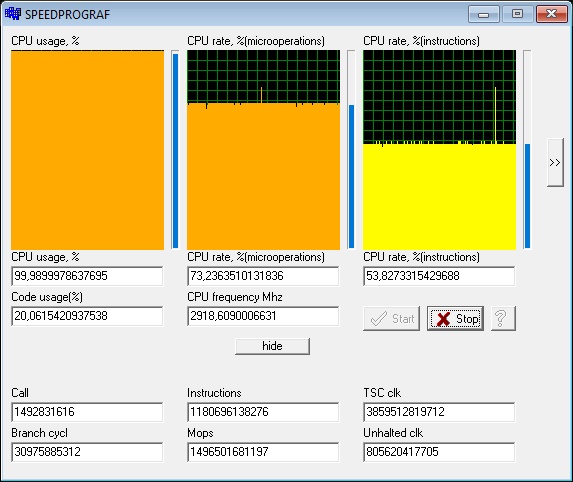
В наиболее полнофункциональной версии (для процессоров Intel Pentium IV – это наиболее полный профилировщик для процессоров X86\_64 серии, а не устаревший, который одновременно может регистрировать до 16 программируемых параметров + TSC) Speedprograf выводит следующие параметры:

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).

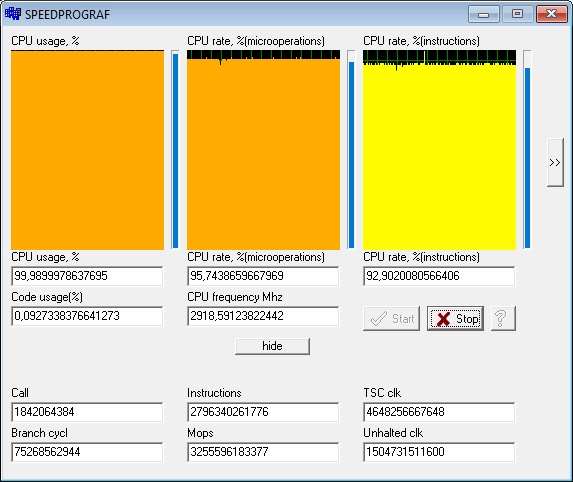
Скорость архивации WinRar выглядит следующим образом:

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!

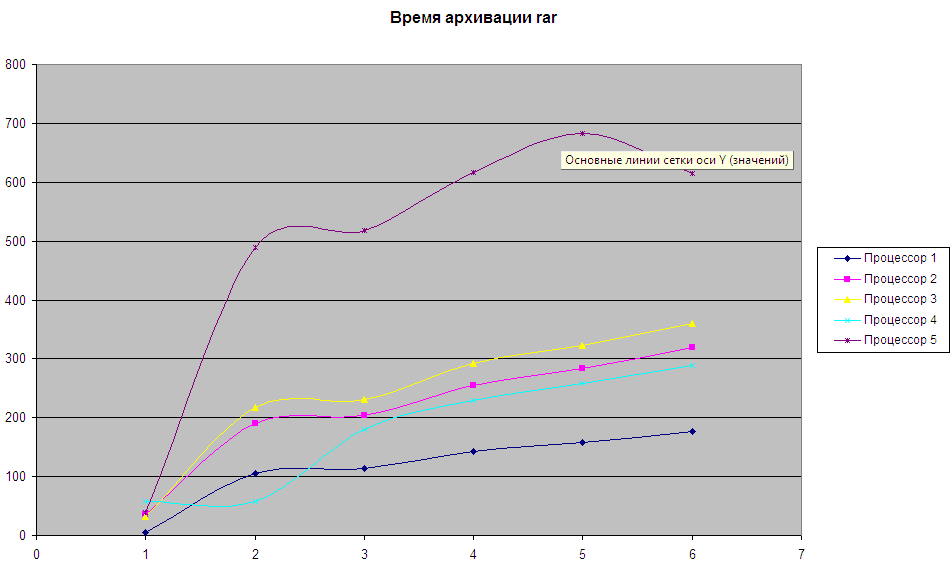
Однако здесь используется нечестный прием тестировщиков. Тест построен на пустых операциях (NOP), где Code usage всего 0,09%. Процессор не обрабатывает данные, с обработкой данных он не может работать с такой скоростью.

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?

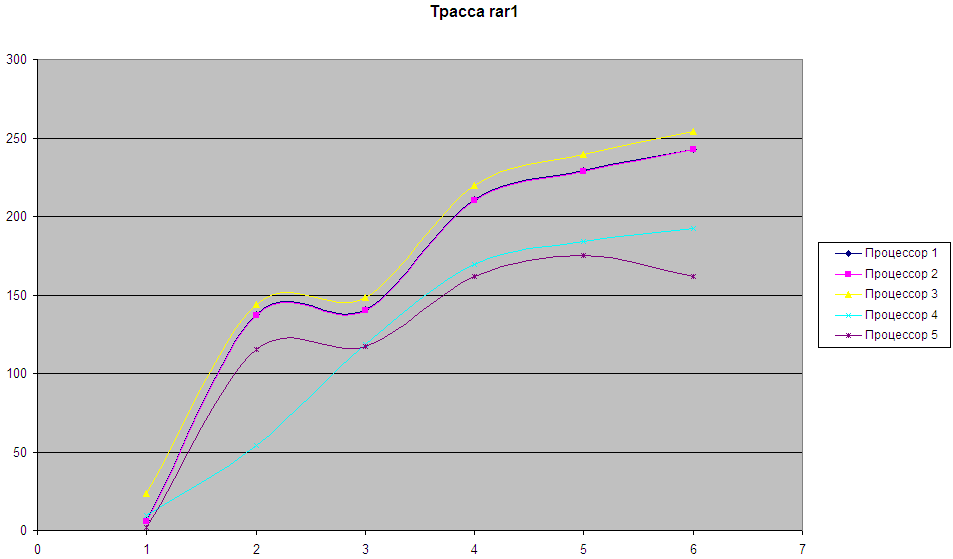
На графике, представленном ниже, показана скорость архивации в GIPS (в миллиардах инструкций в секунду) для различной степени архивации.

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).

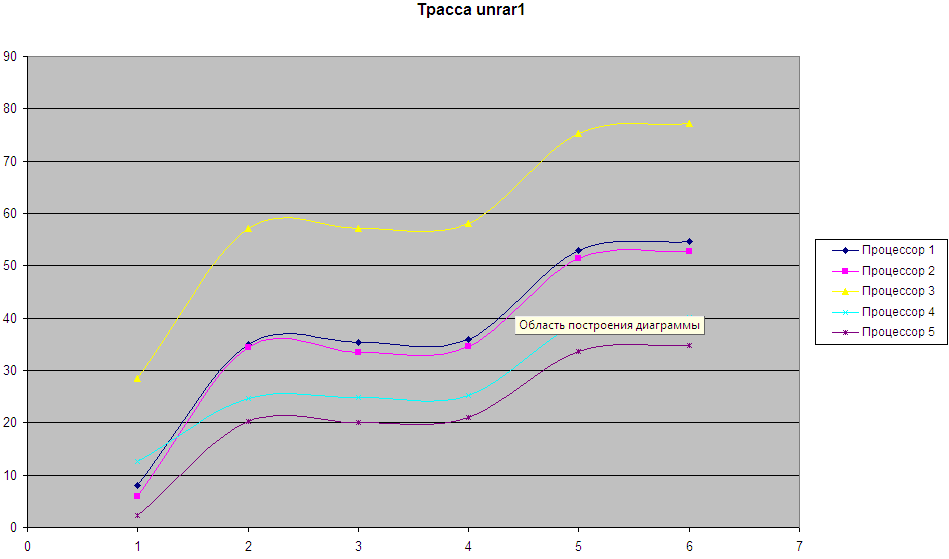
Как мы и ожидали, у процессора №1 минимальное время архивации, у процессора №5 – максимальное, но во 2-ом режиме архивации процессор №4 обогнал лидера производительности. В чем тут дело?

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



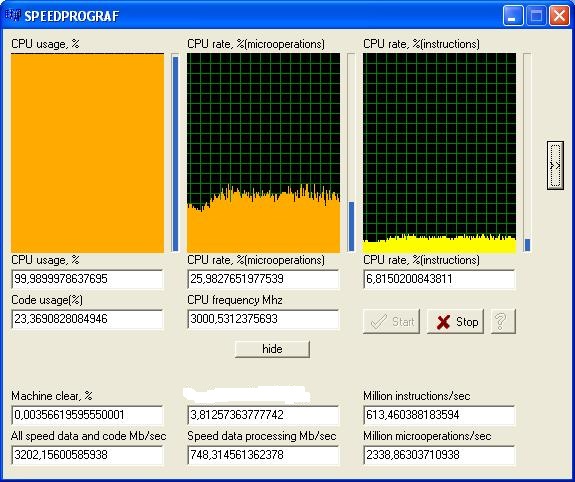
Вот теперь становится понятно, почему «медленный» процессор №4 обогнал процессор №1 с самой высокой производительностью! У него количество выполненных команд во 2-ом режиме архивации в 2 раза меньше, чем у остальных процессоров! А что, разве одну и ту же работу процессор может выполнить различными путями?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

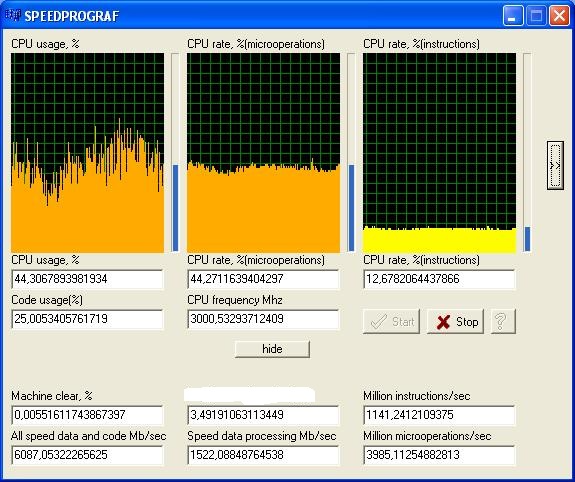


SPEEDPROGRAF подтвердил наличие накладных расходов. Процессор с более высокой и реальной производительностью исполняет трассу другого порядка длиннее, а медленный - короче.

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!