

# **Использование гибридных потоков с целью повышения производительности программ в SMP-системах**

*Студент группы МИ-65М Сарычев Валентин*

## **Предисловие**

Тема моей диссертации «Исследование и разработка моделей многопоточности для повышения производительности программ в SMP-системах». Объектом исследования диссертационной работы являются современные SMP системы. Предметом исследования – схемы реализации потоков выполнения в современных ОС с поддержкой SMP. К ним относятся все популярные многозадачные ОС: Windows 7, Debian Linux, Mac OS, FreeBSD, Windows Server 2008 и прочие.

В ходе проведения исследовательской работы была сделана оценка производительности различных реализаций потоков, разработана серия тестов, приведено формальное представление работы многопоточного процесса, разработана и реализована на языке программирования C++ гибридная модель потоков выполнения, учитывающая проблемы существующих моделей, а также с экономической точки зрения обоснована ее значимость для использования в существующих высоконагруженных программных решениях.

## **Актуальность исследования**

С каждым годом параллельно с развитием качества Интернет-соединения индустрия программного обеспечения делает все больший уклон в сторону развития высоконагруженных систем, способных обслуживать тысячи пользователей в секунду, обрабатывая терабайты данных. К таким системам можно смело отнести продукты Google, Youtube, Yandex, Twitter, NYSE и большое множество серверов электронной торговли.

Методология разработки высоконагруженного ПО опирается на принципы масштабируемости, как вертикальной, так и горизонтальной.

Очевидно, чем больше аппаратных ресурсов задействует ПО, тем острее встает проблема вертикального масштабирования. В большинстве случаев она решается использованием современных приемов программирования, таких как active object, balking pattern, thread pool pattern, reactor patter.

Однако на сегодняшний день существуют программные комплексы, разрабатываемые более десятка лет (NYSE, ROX, Plaza), написанные с использованием традиционного

подхода, когда одного клиента обслуживает один поток выполнения. Практика показывает, что с точки зрения производительности эффективность такого подхода сильно снижается при большом количестве клиентов, а техническая сложность этих проектов не позволяет провести их модернизацию в виду ее дороговизны.

Также существует спектр задач, для которых разработку ПО с экономической точки зрения более приемлемо вести в рамках традиционного подхода параллельного программирования.

В эпоху однопроцессорных ЭВМ эти проблемы прекрасно решались бы заменой вытесняющих потоков выполнения на кооперативные потоки. Но на тот момент не существовало такого количества высоконагруженных систем, как в наши дни. А с появлением многоядерных процессоров неэффективность применения кооперативных потоков стала пропорциональна количеству процессоров SMP системы.

### ***Выдержка из исследовательской работы***

По результатам исследования свойств существующих базовых моделей многопоточности: кооперативной и вытесняющей – по-отдельности были сделаны выводы:

- кооперативные потоки гораздо стабильнее, чем вытесняющие, при возрастании их количества. (Я говорю, об инвариантности общей производительности);
- вытесняющие (системные) потоки могут задействовать более одного логического ядра ЦП.

Имитационная модель послужила базисом для изучения свойств разработанной гибридной модели еще до ее реализации и помогла с приемлемой точностью ответить на вопрос о целесообразности ее разработки. Результаты экспериментов с целью подтверждения выдвигаемых гипотез анализировались с использованием специализированного ПО для профилирования программ.

Одна из главных проблем экспериментальной модели заключалась в том, что модель кооперативных потоков не позволяет процессу задействовать больше одного логического вычислительного ядра процессора. В связи с этим понятно, что сравнивать производительность моделей кооперативных и вытесняющих потоков (потоков ядра ОС) в SMP системах некорректно. Проблема эта была решена путем отображения в равных долях потоков пространства пользователя на процессы ядра таким образом, чтобы количество занимаемых логических ядер для каждой модели потоков выполнения совпадало в каждой тестовой ситуации. Для того, чтобы не возникало издержек планирования процессов, максимальное количество выделяемых процессов не должно превышать количества доступных логических ядер ЦП. Понятно, что разработка гибридной модели сводится к

нетривиальной замене процессов, на которые отображаются кооперативные потоки, на системные потоки. Введенное ограничение на количество процессов сводит на нет «тяжелые» кросс-ядерные переключения их контекстов (на 20-25% медленнее), что максимально приближает поведение имитационной модели для кооперативных потоков к поведению гибридной модели.

Получение графиков теоретической производительности для гибридной модели свелось к разделению графиков на две части по границе, где количество потоков достигает количества логических ядер ЦП, и дальнейшему сведению левой части графика для системных с правой частью для кооперативных потоков путем масштабирования последней.

В качестве кооперативной модели были выбраны реализации State Threads и GNU Portable Threads. На сегодняшний день это, возможно, единственные работоспособные хорошо задокументированные библиотеки.

Ниже приведены блок-схема взаимодействия программ имитационной модели и график с результатами применения разработанной библиотеки гибридных потоков на экспериментальном web-сервере.

