

# Масштабирование систем визуализации

Александр Анциферов (ARBYTE)

В современном быстро меняющемся мире постоянным является только одно: год от года задачи, которые приходится решать той или иной компании, становятся всё более и более сложными. Эта тенденция очень ярко проявляется в увеличении объемов генерируемых данных, росте массивов данных и сложности математических расчетных моделей.

Другой важной тенденцией является увеличение объемов информации, перерабатываемой во время жизни человека. Согласно статистическим данным компании *Gartner Inc.*, объем информации, производимой человечеством, растет экспоненциально (рис. 1). Так же экспоненциально растет и производительность вычислительных компонентов, основанных на полупроводниковых технологиях. Согласно закону Мура (*Gordon E. Moore*) число транзисторов в одном микропроцессоре удваивается каждые 18 месяцев (двукратный рост количества ключей в процессорах компании *Intel* происходит в среднем за 24 месяца. – *Прим. ред.*). Хотя скептики и сторонники альтернативных подходов и недорогих решений в сфере обработки информации считают, что рост вычислительных мощностей диктуется монополией нескольких компаний на выпуск аппаратных средств и программного обеспечения, очевидно, что причиной такого роста являются потребности рынка в инструментах, способных обрабатывать растущие объемы информации.

Руководство любой компании так или иначе сталкивается с вопросами расширения ИТ-ресурсов для решения новых, всё более сложных задач. Одновременно каждая компания заинтересована в финансировании основного профиля бизнеса и в сокращении издержек на ИТ. Однако в ситуации постоянного роста запросов к ИТ со стороны подразделений, осуществляющих основной бизнес компании (в частности, для визуализации данных или оперативной обработки результатов проведенных экспериментов), возникает необходимость думать о расширении ИТ-инфраструктуры и добавлении новых вычислительных мощностей.

Именно здесь возникает вопрос – насколько масштабируемой системой обладает предприятие? Ведь если аппаратная платформа не позволяет легко дополнять систему новыми вычислительными блоками, то компания неизбежно начинает вести себя как мать в бедной семье, которая сначала покупает ребенку штанишки “на вырост”, а потом заставляет носить их еще “пару лет”, пока эти штанишки не начнут трещать по швам. Это характерно и для ИТ. Нередко компании тратят лишние

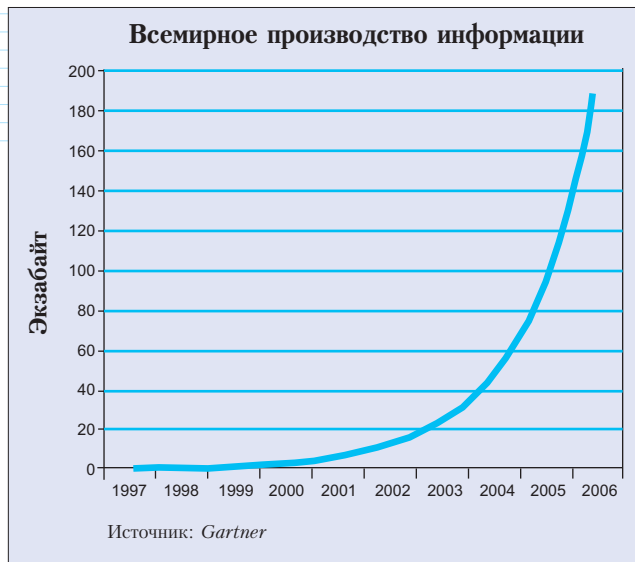


Рис. 1

деньги, покупая мощные серверы, которые не могут полностью задействовать, а потом оттягивают момент модернизации, лишая тем самым многие отделы возможности провести оперативный анализ данных. В идеале, конечно, хотелось бы, чтобы систему можно было нарастить в любой момент и на любое количество вычислительных единиц.

## Чем больше, тем лучше?

На первый взгляд, при подключении к системе всё большего количества ресурсов, время обработки данных должно сокращаться пропорционально добавленным мощностям. Однако в реальном мире по мере того, как система расширяется, из-за несбалансированности отдельных компонентов начинают проявляться внутренние “узкие места”, что негативно сказывается на скорости обработки

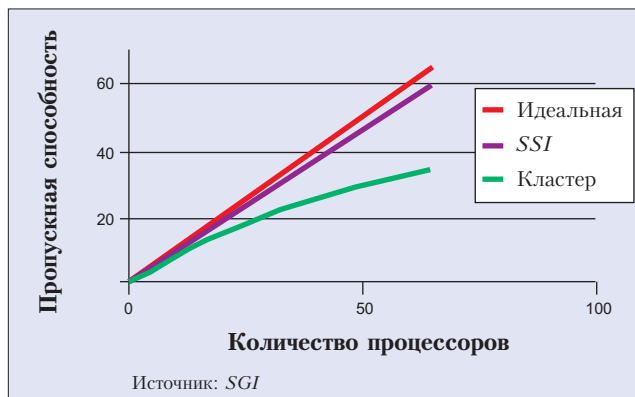


Рис. 2

## Масштабируемость: прошлое, настоящее и будущее

Компания *SGI* производит масштабируемые вычислительные и графические системы более 20 лет, большую часть этого времени – на основе процессоров *MIPS* и операционной системы *IRIX*. С самого начала *SGI* разрабатывала масштабируемые системы для работы в средах с общим системным образом (*Single System Image – SSI*). Технология *SSI* обеспечивает управление системой как единым ресурсом, а не совокупностью отдельных подсистем. При этом вся память и все устройства доступны всем процессам. Сначала системное масштабирование позволило нарастить систему с 1 до 4 процессоров, а затем сделало возможным масштабировать графику в *SSI* до двух графических каналов (*graphics pipes*).

При традиционной архитектуре с симметричной многопроцессорностью (*Symmetric Multi-Processing – SMP*) масштабируемость была лимитирована 24 процессорами и тремя графическими каналами, что быстро стало серьезным ограничением. В конце 1995 года *SGI* представила семейства *Origin* и *Onyx2*, которые стали первыми коммерческими компьютерными системами, построенными на основе архитектуры не-

стандартного доступа к памяти с когерентной кэш-памятью (*Cache Coherent Non-Uniform Memory Access – ccNUMA*).

Архитектура *ccNUMA* открыла возможность эффективно масштабировать систему в более широких пределах, чем это позволяли традиционные шинные решения, наращивая пропускную способность по мере того, как добавлялись процессоры и память. Это гарантировало сбалансированность ресурсов даже для очень больших систем. Чуть позже суперкомпьютеры *Origin* стали масштабироваться уже до 512 процессоров и *1Tb* памяти.

Сейчас суперкомпьютеры семейства *SGI Altix* на базе процессоров *Intel Itanium 2* и стандартной операционной системы *Linux* масштабируются в пределах до 512 процессоров при работе в среде с общим системным образом. Система *Silicon Graphics Onyx4 UltimateVision* допускает использование до 32 стандартных графических карт, доступных в среде *SSI*, а система *Silicon Graphics Prism* – до 512 процессоров *Intel Itanium 2*, до *6.1 Tb* памяти и до 16 независимых графических потоков в *Linux SSI*.

данных. Поэтому компания *SGI* попыталась создать систему, близкую к идеалу, в которой влияние внутренних “затворов” было бы сведено до минимума, а реальная производительность расширенной вычислительной системы была сопоставима с теоретической суммарной производительностью её компонент (рис. 2).

Современная реализация *ccNUMA* называется *NUMAflex* и является фундаментальной системной архитектурой, которая минимизирует задержки, возникающие при обмене данными между различными компонентами системы. В результате центры обработки данных (ЦОД), построенные с использованием *NUMAflex*, достигают на многих приложениях таких величин стабильной производительности, которые соответствуют теоретическим возможным в конфигурациях до 1024 процессоров. Надо сказать, что сегодня для этих вычислительных машин доступно также и программное обеспечение, раскладывающее вычислительную задачу на сегменты, которые будут выполняться сразу на всех доступных процессорах. Но если проблема не может быть достаточно быстро решена на 64 или 128 процессорах, то для ускорения расчетов количество процессоров, отведенных под эту задачу, может быть увеличено. Для достижения предельной производительности число процессоров должно определяться из соображений размещения задачи в полном объеме в доступную кэш-память процессоров.

Тот же метод оценки распределения задачи на несколько вычислительных узлов *SGI*

применяет и к графическим компонентам системы. Ведь проблемы визуализации, некогда казавшиеся неразрешимыми, становятся достаточно простыми, когда задействуется мощность десятков и даже сотен графических процессоров *GPU* (*graphics processing units*). Когда технологии *SGI* применяются для масштабирования *GPU*, эффект достигается так же, как и в случае обыкновенных вычислительных процессоров. Комбинация архитектуры *SGI NUMAflex* с общей глобальной памятью и масштабируемой графики, использующей множественные потоки *OpenGL*, обеспечивает эффективное масштабирование *GPU* без потерь во времени на передачу данных между процессорами. Кроме этого, любое программное обеспечение, которое прежде работало с одним графическим конвейером, может быть модифицировано для использования всех процессорных и графических ресурсов, доступных в системе.

Нельзя забывать, что некоторый рост производительности может быть достигнут за счет масштабирования систем с применением традиционных архитектур – например, при объединении множества *SSI*-машин в кластер. Однако, трудности применения кластера начинаются с физического конфигурирования узлов, дисков, операционных систем, сетевой инфраструктуры, стоечного пространства, энергораспределения и пр. В продуктах *SGI* всё это настроено для сдачи “под ключ”. Но дело не только в этом, ведь кластер – это объединение нескольких независимых вычислительных

машин, которые обмениваются друг с другом данными по сети. Это означает, что кластерная система состоит из нескольких (причем, иногда из сотен или даже тысяч) независимых компьютеров, каждый из которых требует отдельного администрирования. Впрочем, и это не было бы такой уж большой преградой, если бы не ограниченные возможности кластеров в решении сложных задач, которые не могут быть разбиты на отдельные фрагменты. Любой, даже самый быстрый *интерконнект* несравним по скорости с тем, что достигается при использовании архитектуры *NUMAflex*. Таким образом, когда задача требует постоянного обмена большими массивами данных между вычислительными узлами, то основное время работы кластера будет тратиться на передачу данных, а процессоры будут простаивать... Кластеры, безусловно, являются хорошим решением, когда каждый фрагмент задачи помещается в память отдельно взятого узла. В других случаях кластеризация, увы, оказывается неоправданной.

Что касается машин *SSI* с архитектурой *NUMAflex*, то они отличаются использованием глобальной памяти, через которую процессоры обмениваются данными. Таким образом, работая с одним и тем же адресным пространством, процессоры могут легко решать

“недробимую” задачу параллельно. Это подобно тому, как строится самолет в большом ангаре – один инженер работает в хвосте, другой – под крылом. В случае же с кластером мы будем строить самолет очень долго, так как придется постоянно перетаскивать детали между различными рабочими площадками (узлами кластера).

Дополнительное удобство применения *SSI* заключается также в том, что разработка приложений в среде, основанной на *SSI*, очень похожа на разработку для системы с одним процессором и одним графическим конвейером. Единый исполнительный модуль может иметь доступ ко всем данным в памяти одновременно и одновременно же задействовать все *GPU* со множеством графических потоков, работая при этом в едином адресном пространстве. Чтобы отследить эффективность работы приложения в системе *SSI* используется единый отладчик, который позволяет полностью видеть состояние приложения. Это принципиально отличается от процесса разработки кода для кластера, где каждый исполнительный модуль является отдельным процессом на отдельном узле, нуждающемся в отдельном отладчике для определения программных сбоев. ☐

(Продолжение следует)

**Экономия рабочего времени до 49%\***

**Исключительно низкий уровень ШУМА (менее 35дБА)**

**ГАРАНТИЯ 5 ЛЕТ**

**Исключительная производительность для высокопроизводительных вычислительных CAD систем**

Графические станции ARBYTE® CADStation оптимизированы под приложения САПР ведущих производителей ПО: UGS, Autodesk, Dassault Systemes, PTC, AСKON.

Графические рабочие станции ARBYTE® CADStation на базе процессоров Intel® Xeon™ – выдающееся соотношение цена/производительность для решений в области высокопроизводительных вычислительных систем.

\*В сравнении с неспециализированными ПК аналогичной конфигурации. По методика, опубликованной в журнале "САПР и графика" №11 2004, №3 2005.

Обозначения: Celeron, Celeron Inside, Celeron Inside logo, Core Inside, Intel, Intel Core, Intel logo, Intel Inside, Intel Inside logo, Intel SpeedStep, Intel Xeon, Xeon, Xeon Inside, Pentium и Pentium Inside являются товарными знаками, либо зарегистрированными товарными знаками, права на которые принадлежат корпорации Intel или ее лицензиарам на территории США и других стран.

**ARBYTE**  
 Москва ARBYTE  
 (495)-725-8008  
 www.arbyte.ru

