

2. Никитенко Д.А. Рейтинг Top500 – что нового? // Суперкомпьютеры. – 2010. – № 3.

3. TOP 50 суперкомпьютеров (<http://supercomputers.ru>).

А.В. Никологорская, Ф.Н. Ясинский

Ивановский государственный энергетический университет

ОБ ИСПОЛЬЗОВАНИИ СУПЕРКОМПЬЮТЕРА ПРИ ПОСТРОЕНИИ ГИБРИДНОГО МЕТОДА ПРОГНОЗА ПОТРЕБЛЕНИЯ ЭЛЕКТРОЭНЕРГИИ НА ОСНОВЕ АНАЛИЗА ВРЕМЕННЫХ РЯДОВ

Построение оптимального прогноза величины потребляемой электроэнергии является актуальной задачей. В первую очередь это связано с расчётом затрат, планируемых на покупку и реализацию электроэнергии. От точности полученного прогноза будет зависеть экономия денежных средств компаний, работающих на рынке предоставления энергетических услуг.

В зависимости от характера представленных данных, предметной области и величины периода упреждения возникает проблема выбора одного из многих способов построения прогноза, который наилучшим образом будет удовлетворять заявленным требованиям.

В том случае, если наблюдаемое явление описывается с помощью временного ряда, т.е. последовательности числовой величины или набора величин, непрерывно изменяющихся во времени, особенно эффективными являются математические методы прогнозирования.

Временной ряд – это последовательность числовой величины или набора величин, непрерывно изменяющихся во времени. Анализ временных рядов с помощью математических методов позволяет получить будущее значение рассматриваемой величины на основе известных составляющих её временного ряда.

Была поставлена следующая задача: на основании предоставленной информации об энергопотреблении в Костромской области за прошлый период предсказать значения объемов потреб-

ляемой электроэнергии на сутки вперед. В данном случае временной ряд образуют значения объемов потребляемой электроэнергии, изменяющиеся с течением времени. При этом различный характер динамики потребления электроэнергии в рабочие и нерабочие дни привел к разбиению исходного ряда на две составляющих: для будних дней и для выходных соответственно.

В данной работе исследована принципиальная возможность построения гибридного метода прогнозирования и возможность использования суперкомпьютера для ускорения вычислений.

При построении гибридного метода применён следующий подход: в качестве основы были выбраны несколько методов прогнозирования с различными характеристиками и принципами действия. Далее результаты их работы объединяются в единый синтетический алгоритм с использованием определенного набора коэффициентов $\alpha_i \geq 0$, $\sum_i \alpha_i = 1$ $i = 1, 2, 3$ (рис. 1).

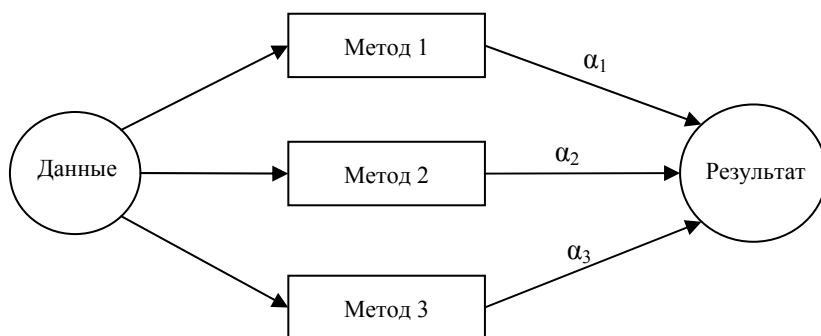


Рис. 1. Схема гибридного алгоритма

В качестве базовых методов прогнозирования были использованы три алгоритма. Выбор первого метода – фильтра Винера обусловлен квазипериодическим характером зависимости энергопотребления от времени [4, 5]. Второй – нейросетевой подход позволяет учесть нелинейность наблюдаемых процессов и сложный характер внутренних закономерностей между элементами, составляющими временной ряд [6]. Третий подход –

эволюционное моделирование призван решить задачу оптимизации [2, 3]. Параметры и характеристики этих методов подробно рассмотрены в [1].

Одной из принципиальных и сложных задач является подбор оптимального набора коэффициентов α_i для получения наименьшей ошибки прогнозирования.

Чтобы избежать полного перебора параметров ставится линейная задача оптимизации для нахождения указанных коэффициентов:

$$r(t) - (\alpha_1 \cdot f(t) + \alpha_2 \cdot n(t) + \alpha_3 \cdot m(t)) \rightarrow \min ,$$

где $r(t)$ – реальное значение величины потребляемой электроэнергии; $f(t)$ – величина потребляемой электроэнергии, полученная с применением фильтра Винера; $n(t)$ – величина потребляемой электроэнергии, полученная с применением нейронной сети; $m(t)$ – величина потребляемой электроэнергии, полученная с применением эволюционного моделирования; $\alpha_i \geq 0$, $\sum_i \alpha_i = 1 \quad i = 1, 2, 3$.

Хотя задача оптимизации и позволяет снизить количество итераций, всё же каждая итерация представляет собой очень ресурсоёмкую операцию, поскольку в ней необходимо решить одну и ту же задачу различными методами. Поэтому использование суперкомпьютера является необходимым для ускорения процесса нахождения коэффициентов. Кроме того, это позволяет применять поисковый алгоритм оптимизации с несколькими начальными точками для большего охвата области решения.

Одним из вариантов является использование классических многопроцессорных суперкомпьютеров, поскольку это дает возможность равномерно распределить нагрузку на все процессоры в системе. Они объединяются в тройки (для трёх коэффициентов) и действуют сообща, находя оптимальный набор коэффициентов.

Другим перспективным вариантом можно считать суперкомпьютер на базе графических ускорителей по технологии CUDA. В данной системе вычисления производятся множеством блоков, состоящих из некоторого числа обрабатывающих пото-

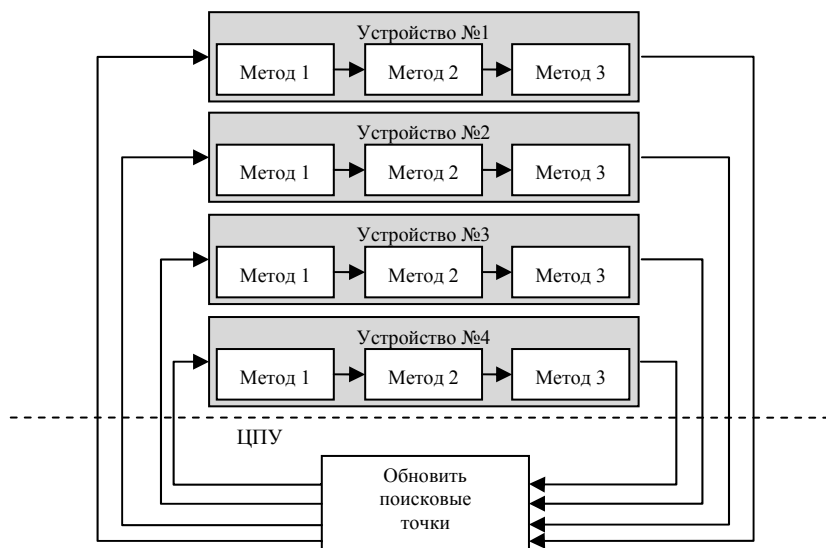
ков. В отличие от MPI Nvidia CUDA представляет собой систему с общей памятью. Технически чип графического ускорителя представляет собой несколько унифицированных мультипроцессоров, каждый из которых содержит по 8 CUDA-ядер [7].

Используемый суперкомпьютер Ивановского института ГПС МЧС России содержит в себе два графических ускорителя GTX 295, каждый из которых имеет по два чипа на плате. Общее количество ядер $240 \text{ CUDA-ядер в чипе} \times 2 \text{ чипа} \times 2 \text{ карты} = 960 \text{ CUDA-ядер}$ при суммарном объёме видеопамати 3,5 Гбайт.

Одним из вариантов разделения данных для решения задачи является назначение каждому потоку своего метода и исходных данных и дальнейшее совмещение полученных результатов. Однако, хотя количество ядер и велико, концепция CUDA не предполагает нагружать каждый поток большим количеством вычислений. Вместо этого используется подход, при котором тысячи потоков исполняют небольшой набор одинаковых команд над собственными данными. Это обусловлено ограничениями архитектуры CUDA [7]. В ней в каждом мультипроцессоре за один раз выполняются только 32 потока, называемые «warp». Причём желательно, чтобы потоки в этом наборе выполняли одну и ту же операцию, иначе warp разбивается на отдельные части.

Следовательно, лучше загрузить каждое графическое устройство какой-либо одной задачей. К тому же, если для указанных базовых методов прогнозирования использовать параллельные схемы для системы CUDA (например, использование библиотеки CUFFT [7]), то можно достичь существенного ускорения расчётов.

В этом случае эффективной схемой будет использование каждого из 4 доступных устройств для решения одной поисковой задачи, т.е. использование 4 поисковых точек. Каждое устройство последовательно выполняет параллельные версии методов прогнозирования и изменяет свои коэффициенты в зависимости от выбранного алгоритма оптимизации (рис. 2).



Рису. 2. Схема гибридного алгоритма для системы CUDA с несколькими графическими ускорителями

Таким образом, использование подобной схемы позволяет эффективно использовать вычислительные мощности суперкомпьютера на графических ускорителях для поиска оптимального соотношения коэффициентов гибридного метода.

Список литературы

1. Никологорская А.В., Сидоров С.Г. Опыт прогнозирования энергопотребления в энергосетях Костромской области // Высокие технологии, исследования, промышленность. Исследование, разработка и применение высоких технологий в промышленности: сб. тр. IX Междунар. науч.-практ. конф. – СПб.: Изд-во Политехн. ун-та, 2010. – С. 266–270.
2. Букатова И.Л. Эволюционное моделирование и его приложения. – М.: Наука, 1979.
3. Емельянов В.В., Курейчик В.В., Курейчик В.М. Теория и практика эволюционного моделирования. – М.: ФИЗМАТЛИТ, 2003.

4. Сергиенко А.Б. Цифровая обработка сигналов. – СПб.: Питер, 2002.

5. Зверев В.А., Стромков А.А. Выделение сигналов из помех численными методами. – Н. Новгород: ИПФ РАН, 2001.

6. Головкин В.А. Нейронные сети: обучение, организация и применение: учеб. пособие для вузов / под общ. ред. А.И. Галушкина. – М.: ИПРЖР, 2000.

7. <http://developer.nvidia.com/object/gpucomputing.html> – NVIDIA GPU Computing Developer.

А.С. Никонов¹, А.В. Русаков²

^{1,2}Нижегородский государственный университет им. Н.И. Лобачевского

ВОСПРОИЗВОДИМОЕ ИСПОЛНЕНИЕ ПАРАЛЛЕЛЬНЫХ SCOOP-ПРОГРАММ

SCOOP (Simple concurrent object-oriented programming) [2, 3, 4] – технология для разработки параллельных программ. Она появилась как дополнение к языку Eiffel [7] и была интегрирована в EVE [8] – исследовательскую версию EiffelStudio.

Множество примеров доказали простоту использования SCOOP [3], однако разработка параллельных программ все еще сопровождается некоторыми проблемами. Одна из основных сложностей – отладка, так как поведение параллельной программы во время исполнения зависит от планировщика операционной системы. Под поведением параллельной программы понимается очередность доступа различных потоков к общей памяти. Возникает задача записи и повторения поведения параллельной программы для последующей отладки. Чтобы решить эту проблему в общем случае, необходимо модифицировать планировщик операционной системы, что невозможно, например, для коммерческих систем. В технологии SCOOP используется собственный планировщик и простая модель синхронизации, что позволяет решить задачу другим способом.

В данной работе реализована техника, помогающая пользователям распознать ошибки синхронизации в SCOOP-программах.