

Агрессивное невежество как один из главных барьеров в суперкомпьютерной области.

Н. Н. Непейвода (ИПС РАН) nepejvodann@gmail.com

Отмечались три барьера на пути развития суперкомпьютеров: скорость света. Выделение тепла (предел Ландауэра) и предел Чейтина (предел сложности конструкций, которые может понять человек). Казалось бы, последний предел самый невинный. Но, как отмечают многие, именно он сейчас стал определяющим в очень многих случаях. Почти патологическим проявлениям этого барьера посвящён данный доклад.

Примечание. Термин «агрессивное невежество» в данной связи принадлежит не мне.

Основные тезисы доклада следующие.

1. Человек, научившийся хорошо кодить на Java или другом подобном традиционном языке приобретает логику, несовместимую с той, которая требуется для решения задач на компьютере нестандартной конфигурации и в итоге пишет последовательные программы и затем кое-как их «распараллеливает». Написать программу на C и затем прибавить к ней параллелизм — чаще всего порождает мёртворождённое решение. Тем не менее такой человек полностью уверен, что он знает всё, что нужно для «дела», а всё остальное — фигня. Это порождает весьма агрессивное невежество, особенно тогда, когда такой человек, ставший начальником, объясняет свою позицию примерно так: «Я в этом ничего не понимаю, поэтому это чушь».

2. Параллельные алгоритмы с самого начала строятся по другим законам, чем последовательные. Для их создания нужно логическое мышление, а логика как наука практически выброшена из учебных программ в России. «Математическая логика», под которой понимают классическую и в которой полностью опущен как неформализуемый раздел, посвящённый взаимосвязям формального и содержательного, лишь ещё более отупляет. Философская логика не лучше. А давать науку в комплексе не хватает ни часов, ни сил, ни методических разработок.

3. Логически созданная программа с самого начала максимально совместна (незаслуженно забытый термин из Алгола-68), что означает возможность легко уложить её на любую подходящую для неё архитектуру.

4. Логически созданная в традиционной парадигме программа даёт отнюдь не оптимальное для конкретной аппаратуры решение. Но данное решение на порядок лучше того, которое получается «распараллеливанием» изначально последовательной программы. Далее уже нужно одновременно и в комплексе видеть как алгоритм, так и ресурсные требования, чему совершенно не учат.

5. Разные архитектуры суперкомпьютеров требуют для получения решения выше среднего различных неклассических логик. Это дополнительная объективная трудность: человек, который, преодолев недостатки образования, вошёл в одну из архитектур (например, в матрично-конвейерную), теряет, когда надо программировать, скажем, на ассоциативном процессоре обработки данных.

6. Поэтому я полностью согласен с предложением С. М. Абрамова: суперкомпьютерная отрасль требует обучения студентов с самого начала по совершенно другим программам, чем обычные информатики.

7. Ситуация ещё обостряется в связи с перспективой перехода к алгебраическим процессорам. Даже выпускники мехмата МГУ хватаются за голову: «Тут крутейшая

математика, причём такая, которой нас совершенно не учили». Поэтому современный программист суперкомпьютеров, не знающий алгебры, может освоить максимум одну структуру и перестроить мозги под неё. При переходе на другую его ум полностью сломается. А в бурно развивающейся отрасли такое недопустимо.

8. Резюмируя, можно сказать. Мышление на уровне конкретных моделей неизбежно приводит к агрессивному невежеству в данной области. Нужно мышление по крайней мере на уровень выше (метамодели). На самом деле адекватно области мышление на уровне преобразований метамodelей.