30.11.2013

Математическое обеспечение – состоит из готовых модулей, умеющих считать математику

МО – стоит над всеми, над железом, ПО, ОС, над всем, что есть в ЭВМ.

Подпрограммы -> пакет программ -> пакет прикладных программ –> библиотеки программ

Пакет программ – комплекс программ для решения заданного круга задач в конкретной прикладной области.

LINPAC

ASPAC

PETS

BLAS – в нём есть 3 уровня для матричных, векторных и матрично-векторных операций.

ATLAS - создание инструментальных систем для создания программ оптимизированных под конкретную машину.

CATRA – решает проблему того, что диапазон параметров может быть меньше, и есть ещё много параметров, что помогает подстроиться под конкретную задачу, после чего оптимизировать решение этой задачи.

Библиотеки программ – набор пакетов программ (т.е. это пакет пакетов) – библиотека решает более широкий круг задач. (их бывает много по нескольку тысяч, но тогда появляется проблема навигации)

Библиотека линейной алгебры – это LINPAC + EISPIC. Для MPI – есть соответствующая версия этой библиотеки.

ANSYS – тут есть особенность, пакеты сертифицированы по разным стандартам, включая ядерщиков Америки и России, может интегрироваться с другими пакетами. Т.е. появился новый фактор – лицензия.

Библиотека программ широкого назначения, и с дружественным интерфейсам. Раньше в других сборщик был в зависимости от языка, а тут сборщик стал свой – пример mathlab.

Расширение функциональных возможностей можно, но тут есть проблема, в том что всё становится сложно и дорого.

**Программирование для векторных ЭВМ**

SIMD – раньше считался eliac4, теперь считается CRAY

Пример:

DO 24 I=2,M

 DO 24 j=1,N

 aij = bij + ci - это не годится для simd , потому что здесь мы тогда обновим всё, а потом нам понядобятся старые через строчку

 ci=bi-1j

 bij=ai+1j \* 2

чинят проблему, переставив индексы, переставив местами команды и создав дополнительную переменную (метод координат)

Т.е. мы тут запользовали следующие операции преобразования – реверс, заведение дополнительного параметра.

DO 24 j=1,N

 DO 24 i=2,M

 tempi = ai+1j

 aij = bij+ci

 bij = tempi \* 2;

Метод называется методом координат, потому что нам нужно работать построчно (либо по столбцам, если мы храним массив, в транспонированном виде)

Тут есть проблема с ограничениями:

Индексы должны быть обязательно константами (в случае, если оптимизация проводится в ручную, то по крайней мере, быть достаточно предсказуемыми (это можно вытащить из логики программы)) а в случае автоматического преобразования – почти нужны константы

Для таких преобразований строят своего рода таблицы умножения и сложения, (как в линале) по которым уже заданы, в каком случае, какие нужны преобразования.

Характеристический вектор:

Когда пишут, какие есть зависимости uij, ui-1j вектор = |<=|

Дальше составив для выражения, или блока выражений характеристические вектора, можно смотреть и анализировать зависимости и придумывать оптимизации.

Вектора при правильных соответствующих преобразованиях можно располагать не обязательно горизонтально, а ещё можно и по диагонали и вообще разными различными способами. Такое называется “Фронтом волны”.

Сейчас пытаются снять ограничения на индексы.

Пример полной засады: когда у нас например есть минимум и максимум (на время этого предложения забудь про то, что максимум и минимум можно вычислять без if)

Ещё сильно всё становится хуже, если мы индексы преобразуем не только на какую-то константу, но появляется индексная арифметика – например сложение индексов.

Ещё есть попытки автоматического распараллеливания по MPP – ну тут всё совсем как-то жалко.