



*Разработка параллельного  
алгоритма СПУИП в составе Allted  
для систем с общей памятью*

**студент гр. ДА-62 Горбик А.В.**

# *Содержание*

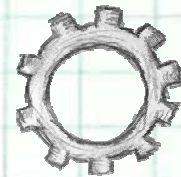


- Поставленные задачи
- СПУИП
- Параллельный алгоритм с PVM
- Параллельный алгоритм с Multithreading
- Основные показатели параллельных алгоритмов
- Планы на будущее

# *Поставленные задачи*



- показать возможность объединения PVM и Multithreading ;
- адаптировать алгоритм СПУИП для МВС с общей памятью;
- получить соотношения для оценки параллельных алгоритмов.



# *Актуальность решаемой задачи*



- уменьшение времени модуляции;
- адаптация работы для МВС с общей памятью;
- оптимальное количество процессоров для задач – 8;

# *Актуальность решаемой задачи*

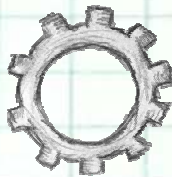


- многоядерность как способ увеличения производительности;
- анонсированный уровень технологии многоядерности.

# Преимущества метода СПУИП



- поиск глобального оптимума;
- эффективность не зависит от количества параметров и начальной точки;



# Преимущества алгоритма СПУИП

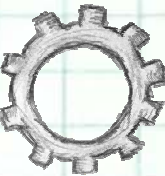


- используется для нахождения «хорошей» начальной точки;
- обладает явно выраженным параллелизмом.

# *Работа с библиотекой PVM*



- работа на виртуальной машине;
- передача данных по сети через буферы обмена;
- запуск отдельных процессов.





# *Параллельный алгоритм СПУИП с использованием PVM*



*Master*- и *Slave*-подпрограммы.

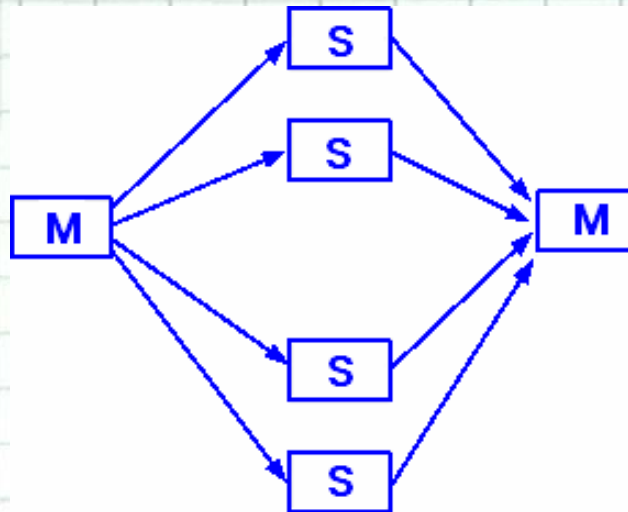
*Master*: организация вычислений,  
управление работой.

*Slave*: расчет ЦФ.

# Параллельный алгоритм СПУИП с использованием PVM



Схема обмена данными:



Проверка критериев остановки в  
конце шага

# *Работа с Multithreading*



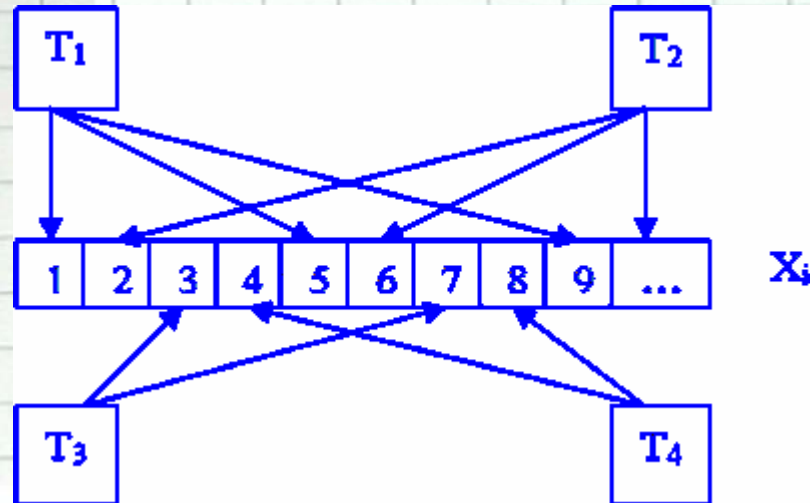
- использование ядер процессора;
- обращение к глобальной памяти;
- запуск “легких” потоков;
- синхронизация доступа к общим данным.

# Параллельный алгоритм СПУИП с использованием Multithreading



Ведомый и ведущие потоки.

Измененный порядок расчета ЦФ:



Проверка критериев остановки после  
каждого расчета ЦФ.

# Параллельный алгоритм СПУИП с использованием Multithreading



Выигрыш в количестве расчетов:



# *Преимущества алгоритма СПУИП с использованием Multithreading*



- запуск потоков быстрее, чем запуск новых задач;
- обмен данными через глобальную память, без сети;

# *Преимущества алгоритма СПУИП с использованием Multithreading*



- не надо алгоритм балансирования загрузки;
- асинхронное обращение к памяти;
- выигрыш в количестве подсчитанных ЦФ.

## *Основные показатели*



- $T_1$ - время решения последовательного алгоритма;
- $T_p$ - время решения параллельного алгоритма;
- $K_y$ - коэффициент ускорения.





# Основные показатели



Для последовательного алгоритма:

$$T_1 = T_{\text{формирования}} + T_{\text{оптимизации}}^{\text{послед}} + T_{\text{результата}}$$

Число подсчитанных ЦФ:

$$Z = (N + 2n)(N_{\text{step}} - 1) + N_K + 2n$$

$$T_{\text{оптимизации}}^{\text{послед}} = t_{1\text{ЦФ}} Z = t_{1\text{ЦФ}} \left( (N + 2n)(N_{\text{step}} - 1) + N_K + 2n \right),$$

$$T_1 = t_{1\text{ЦФ}} \left( (N + 2n)(N_{\text{step}} - 1) + N_K + 2n \right) + T_{\text{формирования}} + T_{\text{результата}}$$

# Основные показатели



Для параллельного алгоритма с PVM:

$$T_{P1} = T_{\text{оптимизации}}^{\text{парал}} + T_{\text{затраты}}^{\text{PVM}} + T_{\text{результата}} + T_{\text{формирования}} =$$
$$= t_{\text{ЦП}} \left[ \frac{N}{p} + 2n \right] N_{\text{step}} + T_{\text{затраты}}^{\text{PVM}} + T_{\text{результата}} + T_{\text{формирования}}$$

Для параллельного алгоритма с Multithreading:

$$T_{P2} = T_{\text{оптимизации}}^{\text{парал}} + T_{\text{затраты}}^{\text{thread}} + T_{\text{результата}} + T_{\text{формирования}} =$$
$$= t_{\text{ЦП}} \left[ \left( \frac{N}{p} + 2n \right) (N_{\text{step}} - 1) + \frac{N_K}{p} + 2n \right] + T_{\text{затраты}}^{\text{thread}} + T_{\text{результата}} + T_{\text{формирования}}$$

# Коэффициенты ускорения



Для параллельного алгоритма с PVM:

$$K_{v1} = \frac{T_1}{T_{P1}} = \frac{t_{1Ц\neq} ((N+2n)(N_{step}-1) + N_K + 2n) + T_{результата} + T_{формирования}}{t_{1Ц\neq} \left[ \frac{N}{p} + 2n \right] N_{step} + T_{затраты}^{PVM} + T_{результата} + T_{формирования}}$$

Для параллельного алгоритма с Multithreading:

$$K_{v2} = \frac{T_1}{T_{P2}} = \frac{t_{1Ц\neq} ((N+2n)(N_{step}-1) + N_K + 2n) + T_{результата} + T_{формирования}}{t_{1Ц\neq} \left[ \left( \frac{N}{p} + 2n \right) (N_{step}-1) + \frac{N_K}{p} + 2n \right] + T_{затраты}^{thread} + T_{результата} + T_{формирования}}$$

# Полученная оценка



Из условия  $K_{y2} \geq 1$  получим:

$$t_{1\text{ц\#}} \left(1 - \frac{1}{p}\right) \left( (N_{\text{step}} - 1)N + N_K \right) \geq T_{\text{загрузка}}^{\text{thread}}$$

пренебрегая значением  $2N(N_{\text{step}} - 1)$ :

$$t_{1\text{ц\#}} Z \left(1 - \frac{1}{p}\right) \geq T_{\text{загрузка}}^{\text{thread}}$$

# *Полученный выигрыш для тестовой схемы*



Из данного файла задания, результатов:

- о размер выборки  $N = 200$ ;
- о количество рассчитанных ЦФ  $Z=710$ ;
- о количество задач/ потоков  $p=4$ ;
- о количество шагов  $k=4$ .

# Полученный выигрыш для тестовой схемы



• время расчета одной ЦФ 2,2с;  
• время выполнения базового алгоритма 484 с.

$$\Delta T_{ЦФ} = t_{1ЦФ} \left( \left[ \frac{N_{step} * N - Z}{p} \right] - 1 \right).$$

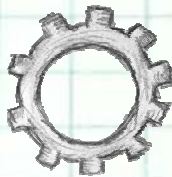
$$\Delta T_{ЦФ} = 2.2 \left( \left[ \frac{4 * 200 - 710}{4} \right] - 1 \right) = 48.4(c).$$

$$\frac{48.4}{484} * 100\% = 10\%.$$

# *Особенности работы для МВС с общей памятью*



- оптимизация параллельного алгоритма под архитектуру кластера;
- зависимость работы от архитектуры, неэффективность на МВС с распределенной памятью.



# *Решение проблемы*

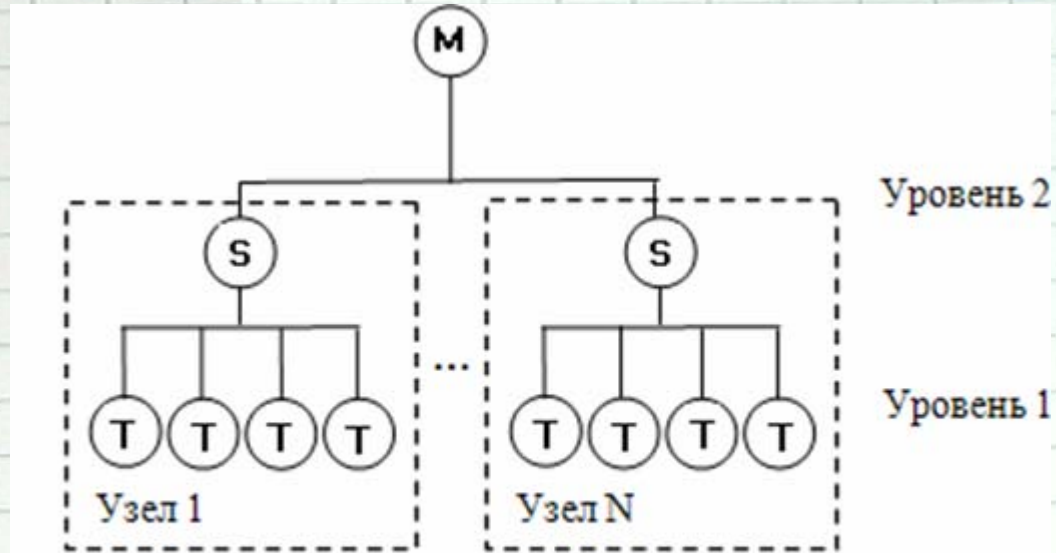


о использование дополнительных  
вычислительных узлов

о реализация двухуровневой  
структуры Allted



# Двухуровневая структура



уровень 1 – между ведущим и ведомыми потоками;  
уровень 2 – между *master*- и *slave*-программами.

# Двухуровневая структура



2-ой уровень отвечает за настройку выполнения, выбирая одну из трех реализаций алгоритма:

- PVM;
- Multithreading;
- PVM + Multithreading.

# *Анализатор сложности вычислений*



Анализатор должен учитывать:

- о объем вычислений (подсчет элементов в электрической схеме и деление их на классы);

# *Анализатор сложности вычислений*



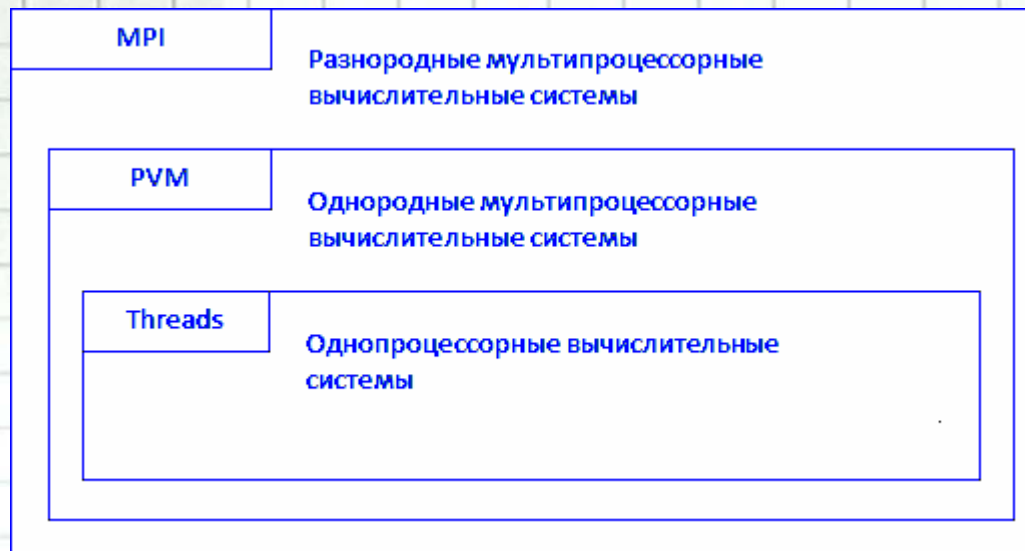
- о архитектуру вычислительной среды (конфигурационный файл);

- о степень загруженности сети (динамическая структура данных; баланс загрузки).

# Трехуровневая организация



Надстройка, объединяющая разные типы суперкомпьютеров (кластер «КПИ» и кластер кафедры СП).





*Спасибо за внимание*

*Вопросы?*

