Тема: Разработка, анализ и реализация алгоритма динамической балансировки нагрузки на мультикомпьютер для случая моделирования взрыва/коллапса

Студент: Сумбатянц Илья

Научный руководитель: Малышкин В. Э.

PIC-метод

Описание:

- Моделирование физических явлений
- Физическое пространство→пространство моделирования
- Дискретизация ПМ→разбиение сеткой на ячейки
- С каждой ячейкой могут быть связаны частицы

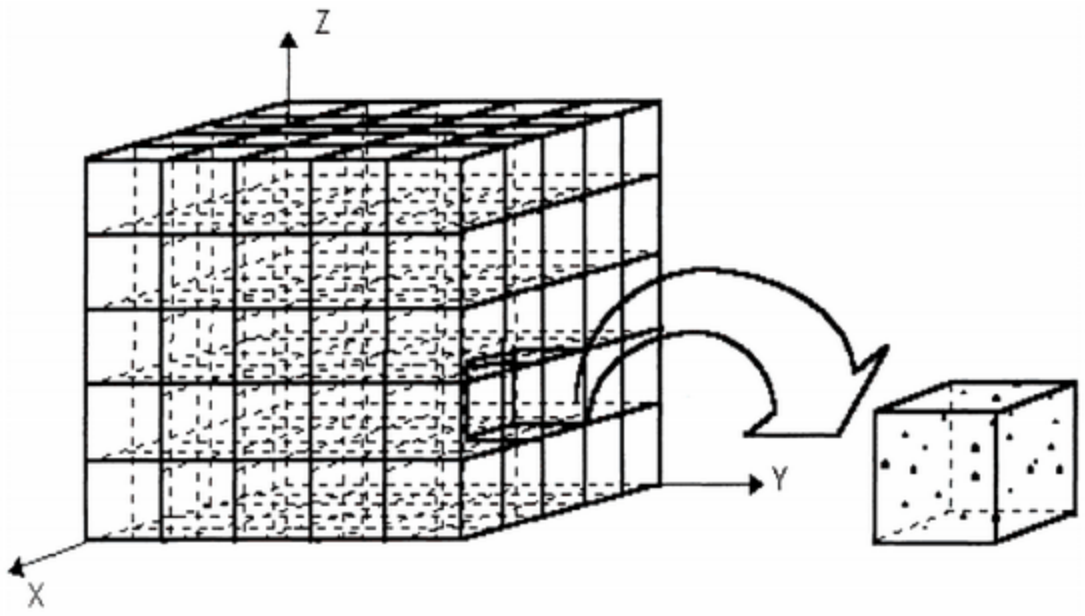
Моделирование:

- Серия временных шагов
- Один шаг→перемещение частиц не дальше одной ячейки

Шаг моделирования:

1. Вычисление значения полей ячеек
2. Перемещение частиц под действием полей в ячейке

PIC-метод



Параллельный PIC

Декомпозиция пространства моделирования:

- Атомарный фрагмент – ячейка пространства моделирования
- Пространство делится на фрагменты и отображается на ресурсы мультикомпьютера

Виртуальные фрагменты

Проблема: недостаток ресурсов узла для обработки фрагмента ПМ

Примеры: моделирование взрыва, коллапса, движение солитона

Решение: отображение фрагмента на несколько узлов без дублирования частиц

Источники дисбаланса

1. Движение частиц
2. Асинхронное исполнение
3. Переменный временной шаг
4. Адаптивные сетки

Динамическая балансировка

Цели:

- Расширение класса моделируемых задач
- Увеличение производительности

Требования:

1. Связанность атомарных фрагментов ПМ
2. Равномерная нагрузка на узлы
3. Масштабируемость

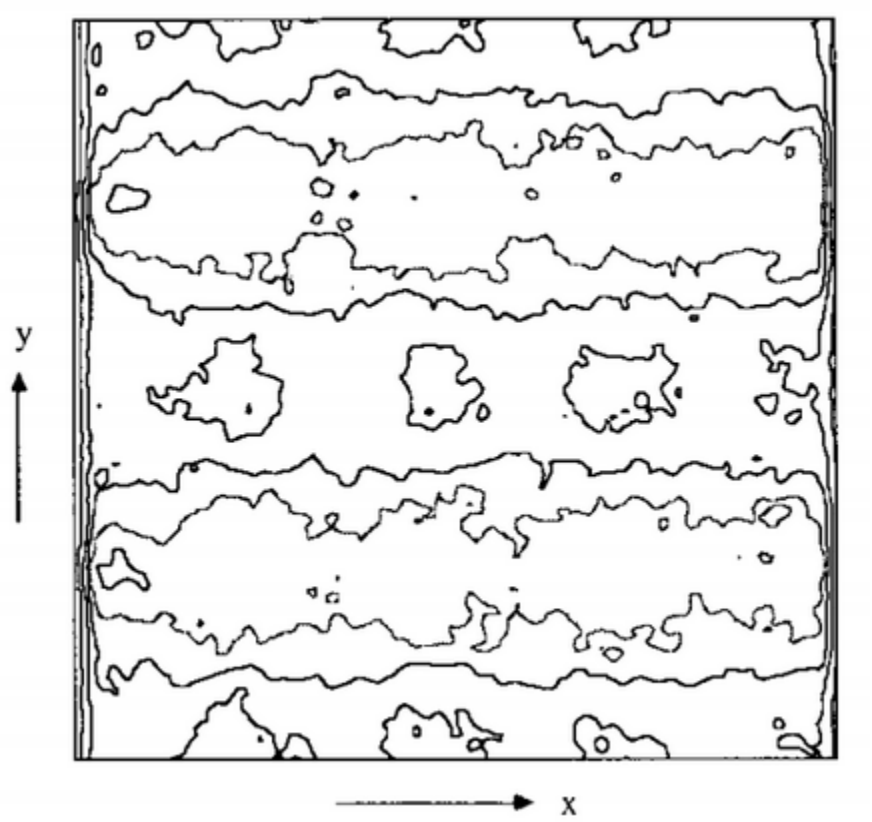
Dynamic Load Balancing for a 2D Concurrent Plasma PIC Code *(Robert D. Ferraro, Paulet C. Liewer, Viktor K. Decyk)*

Идея алгоритма:

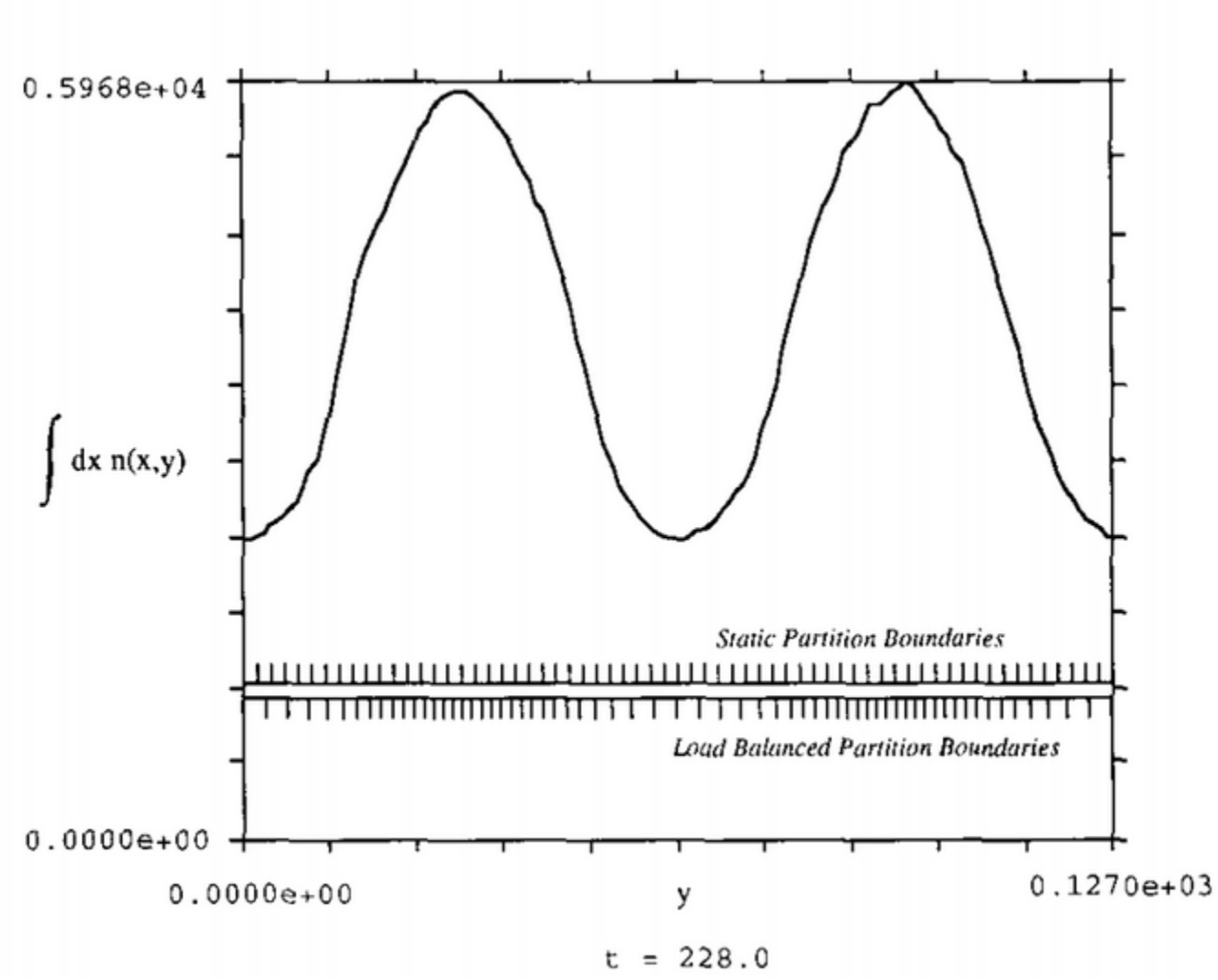
1. 2D $\longrightarrow$ 1D
2. Вычислить плотности по *y* в текущем разбиении
3. При дисбалансе рефрагментировать ПМ так, чтобы плотности по *y* были равны

Dynamic Load Balancing for a 2D Concurrent Plasma PIC Code *(Robert D. Ferraro, Paulet C. Liewer, Viktor K. Decyk)*

2D PIC-метод (контуры плотности):



Dynamic Load Balancing for a 2D Concurrent Plasma PIC Code *(Robert D. Ferraro, Paulet C. Liewer, Viktor K. Decyk)*



Dynamic Load Balancing for a 2D Concurrent Plasma PIC Code *(Robert D. Ferraro, Paulet C. Liewer, Viktor K. Decyk)*

Вычисление левой границы:

$$p\frac{N_p}{N_{proc}} = \int_0^{y_l} dy\, n(y)$$

Алгоритм балансировки:

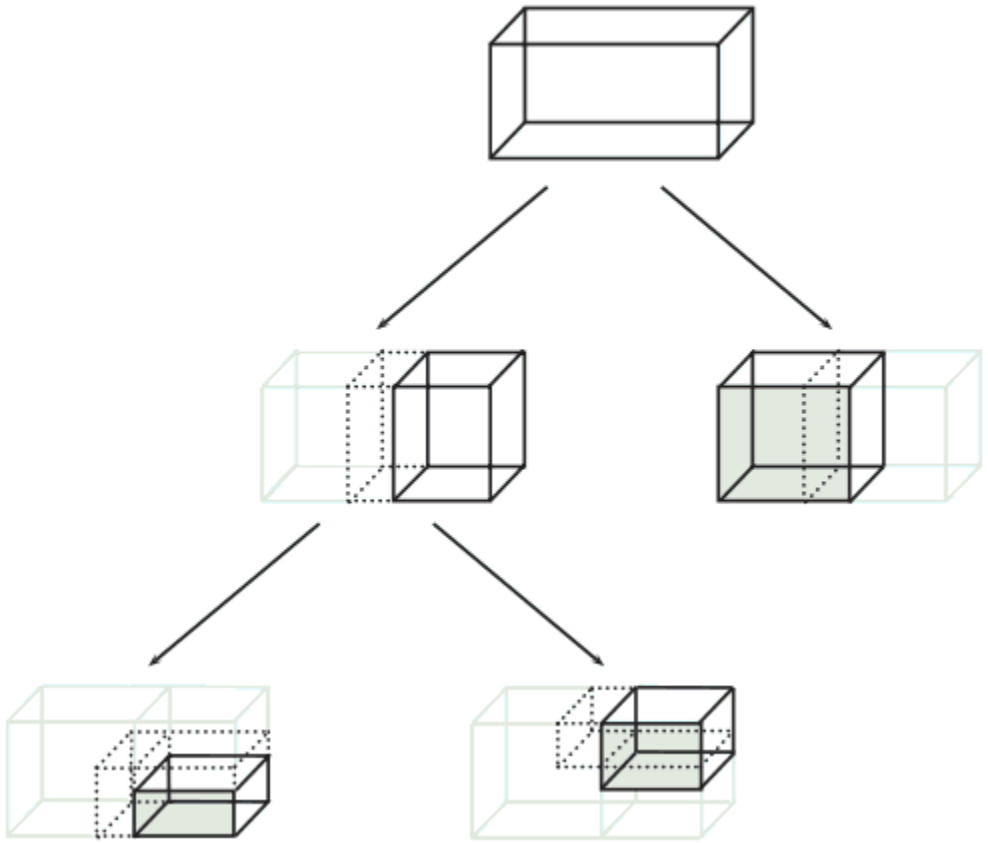
1. Частичное значение плотности вычисляется локально в текущем разбиении и распространяется по узлам
2. Вычисляется функция плотности
3. Каждый узел вычисляет значение своей левой границы и отправляет его левому соседу

Ортогональная рекурсивная бисекция

“Characterizing the parallel performance of a large-scale, particle-in-cell plasma simulation code” *(David W. Walker)*

“A parallel 3D particle-in-cell code with dynamic load balancing” *(Felix Wolfheimer, Erion Gjonaj, Thomas)*

Ортогональная рекурсивная бисекция



Assembly technology for parallel realization of numerical models on MIMD-multicomputers (*M. A. Kraeva , V. E. Malyshkin)*

Централизованный алгоритм

Выделенный узел собирает информацию о плотностях всех минимальных фрагментов и передает её всем узлам. Узлы опираясь на эту информацию строят "карту" нагрузки. После чего соседние узлы обмениваются минимальными фрагментами в соответствии с "картой".

Децентрализованный алгоритм

Каждый узел следит за миграцией частиц в окрестности. При любом изменении состояния узел принимает или отдает часть частиц узлу, в котором произошло изменение. Следует отметить, что этот алгоритм оперирует только виртульными фрагментами.

Специализированный децентрализованный алгоритм

Данный алгоритм опирается не только на информацию об изменении состояния окрестности, но и на информацию о векторе скорости движения частиц. В процессе балансировки частицы мигрируют в направлении противоположном направлению перемещающихся частиц. Таким образом частицы могут мигрировать с упреждением, чтобы сократить количество балансировок.

Algorithms of Parallel Realisation of the PIC Method with Assembly Technology *(M. A. Kraeva, V. E. Malyshkin)*

Диффузионный алгоритм

1. Линейка ПЭ
2. Размер окрестности – 2

Алгоритм

1. На нечетном шаге каждая пара ПЭ (2i, 2i + 1; 0 <= i <= N/2 - 1) обменивается информацией о количестве частиц и принимает решение о передаче части частиц.
2. На четном шаге каждая пара ПЭ (2i-1, 2i; 0 <= i < (N - 1)/2) обменивается информацией о количестве частиц и принимает решение о передаче части частиц.
3. И так D шагов.

Тестирование алгоритмов балансировки

Проблема: анализ алгоритмов балансировки на различных задачах

Цель: реализация тестового стенда

Тестовый стенд (требования)

Глобальные требования:

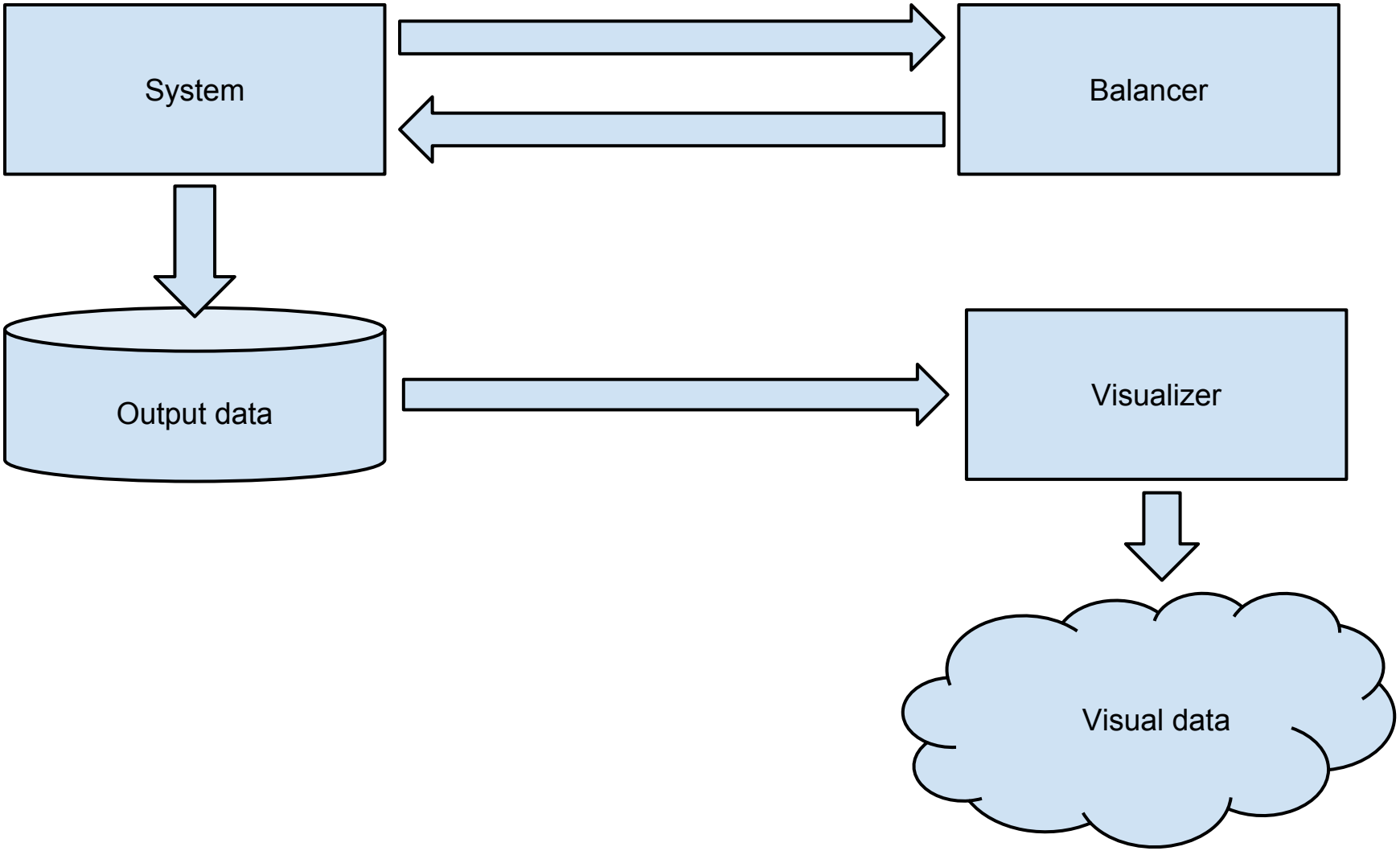
Вход: [задача, начальные данные], балансировщик

Выход: данные, полученные в процессе моделирования

Требования, необходимые для дипломной работы:

Выход: результаты тестирования в графическом представлении и методики интерпретации результатов

Тестовый стенд (архитектура)



Тестовый стенд (API)

API доступный балансировщику:

1. [команда] Передать часть задач определенному узлу
2. [команда] Отправить сообщение
3. [запрос] Текущая нагрузка
4. [запрос] Соседние узлы
5. [оповещение] Начало балансировки
6. [оповещение] Завершение балансировки

API доступный стенду (системе):

1. [команда] положить сообщение

Данные на выходе:

1. Данные каждого узла в отдельной директории
2. Разные данные в разных файлах
3. Формат вывода:
    a. (значение)
    b. (шаг, значение)
    c. (значение, значение)

Тестовый стенд (методики оценки)

Сохраняемые данные:

1. Нагрузка узлов
2. Загрузка узлов
3. Время балансировок на каждом узле
4. Время моделирования на каждом узле
5. Время коммуникаций

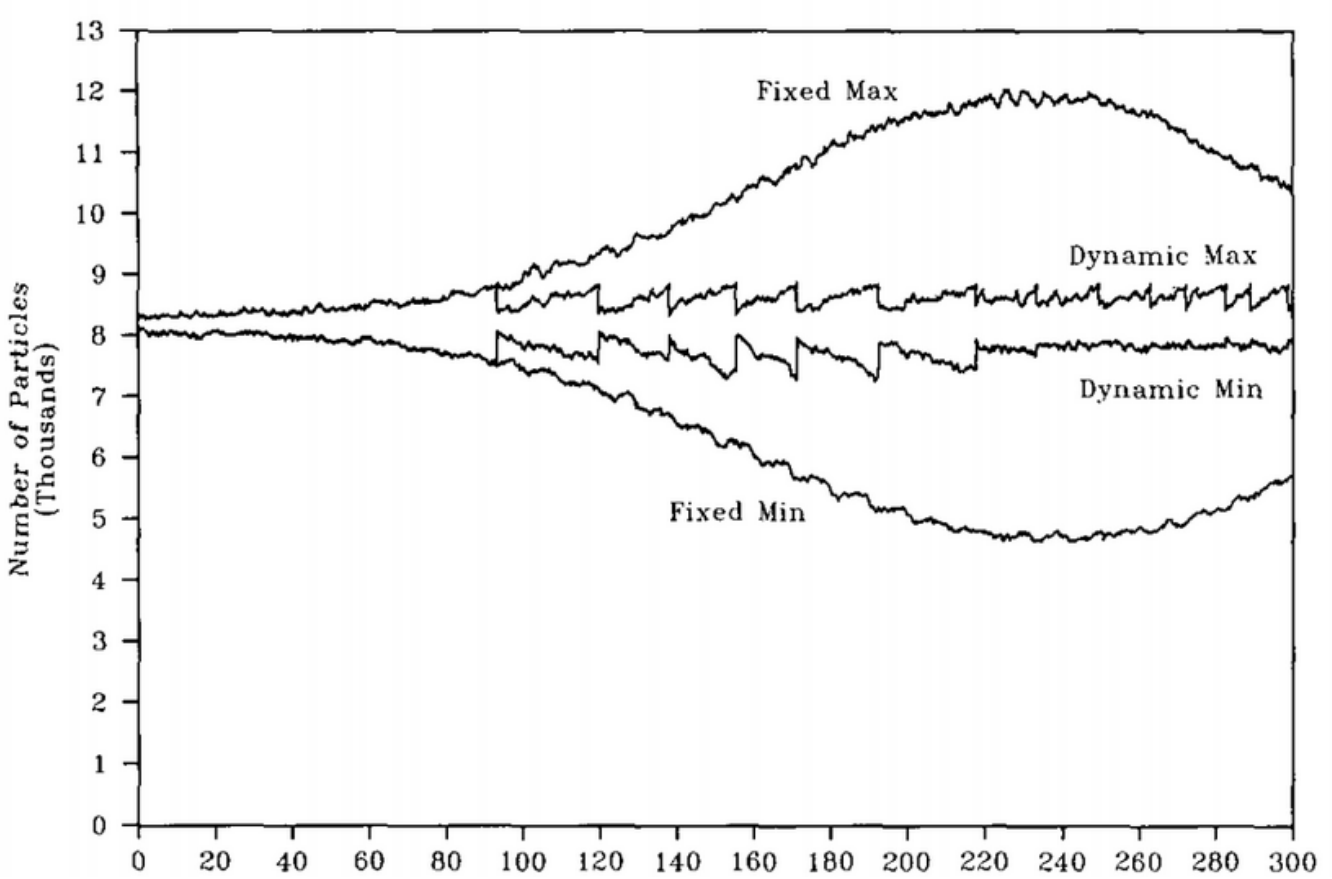
Тестовый стенд (методики оценки)

Информация, которую можно получить из выходных данных:

1. Развертка нагрузки на узлы
2. Зависимость максимальной нагрузки от времени
3. Зависимость минимальной нагрузки от времени
4. Развертка количества балансировок на каждом узле
5. Развертка количества передач данных
6. Время коммуникаций
7. Развертка загрузки узлов
8. Средняя загрузка мультикомпьютера
9. Время процесса моделирования

Тестовый стенд (методики оценки)

Визуализация максимума и минимума нагрузки (пример):



Локальная цель

Сделать стенд