Планирование задач для вычислительного кластера с учетом сети и многопроцессорности узлов *

П.Н. Полежаев
Оренбургский государственный университет

* Исследования выполнены при поддержке Министерства образования и науки Российской Федерации в рамках реализации ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009-2013 гг. (государственные контракты №14.740.11.0287, №14.740.11.0689)

Недостатки современных алгоритмов планирования задач

- **Не учитывается топология** вычислительной системы
- **Не принимаются** во внимание явно **коммуникационные задержки**, связанные с обменом информацией между процессами, а также с **сетевой конкуренцией**
- **Не рассматривается иерархичность** организации вычислительных узлов:

  узел – процессор – ядро

Исследованные алгоритмы планирования

Сочетания **политик планирования** Most Processors First Served Scan (MPFS Scan) или Backfill с **методами назначения задач**:

- **толстое дерево** - Sorting Nodes by Performance (SNP), Fat Tree Sorting Commutators by Performance (FTSCP), Fat Tree Sorting Commutators by Cores (FTSC)
- **тор** – модифицированные варианты алгоритмов Minimizing message-passing Contention 1x1 (MC1x1) и Minimizing message-passing Contention 1x1 Incremental (MC1x1+Inc)
- **звезда** – Sorting Nodes by Performance (SNP) и Sorting Nodes by Speed (SNS).
- **без учета топологии** - First Fit (FF), Best Fit (BF), Fastest Node Fist (FNF) и Random First (RF)
- **для различных топологий – Summed Distance Minimization (SDM), Maximum Distance Minimization (MDM)**

Симулятор – инструмент экспериментального исследования алгоритмов планирования

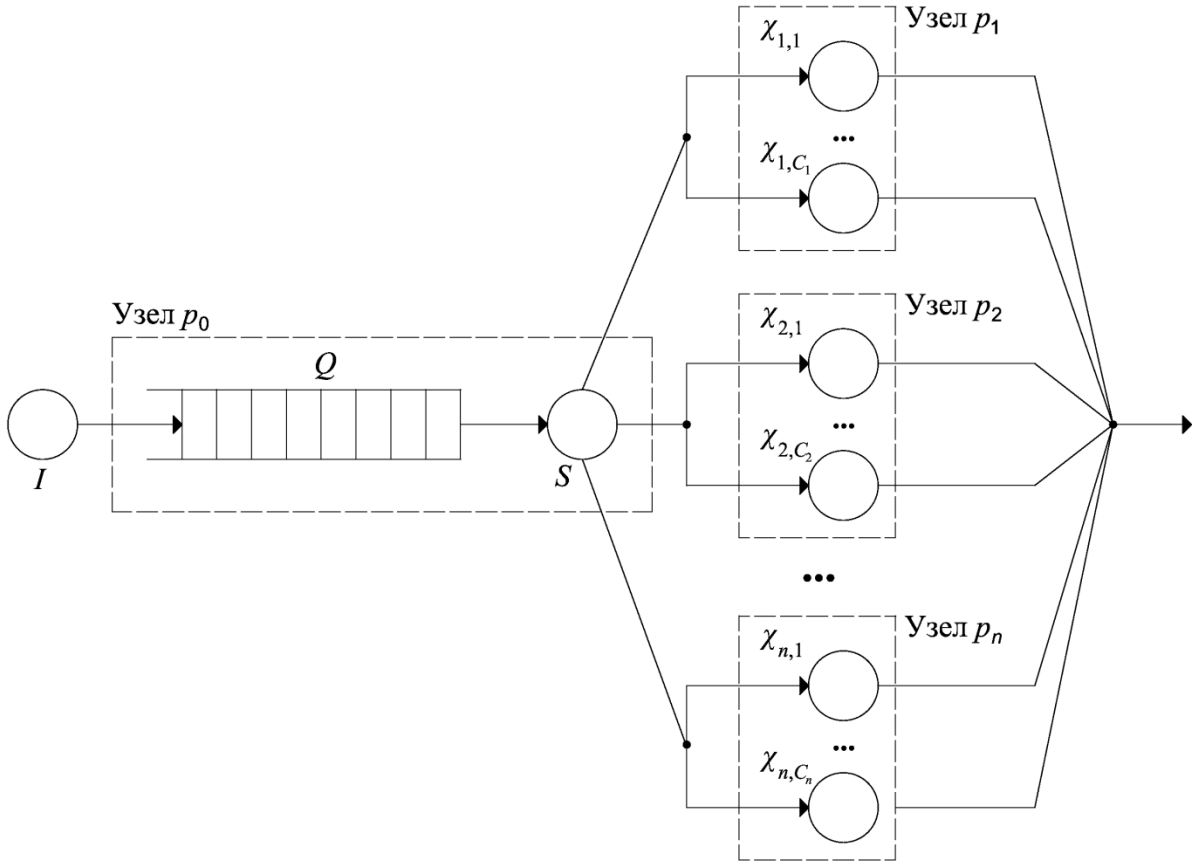
Почему симулятор, а не реальный кластер?

- дороговизна кластерного времени;
- обеспечение контролируемой вычислительной среды;
- гибкость модификации аппаратной платформы, режимов работы;
- простота генерации потока параллельных задач

Характеристики созданного симулятора TopSimity

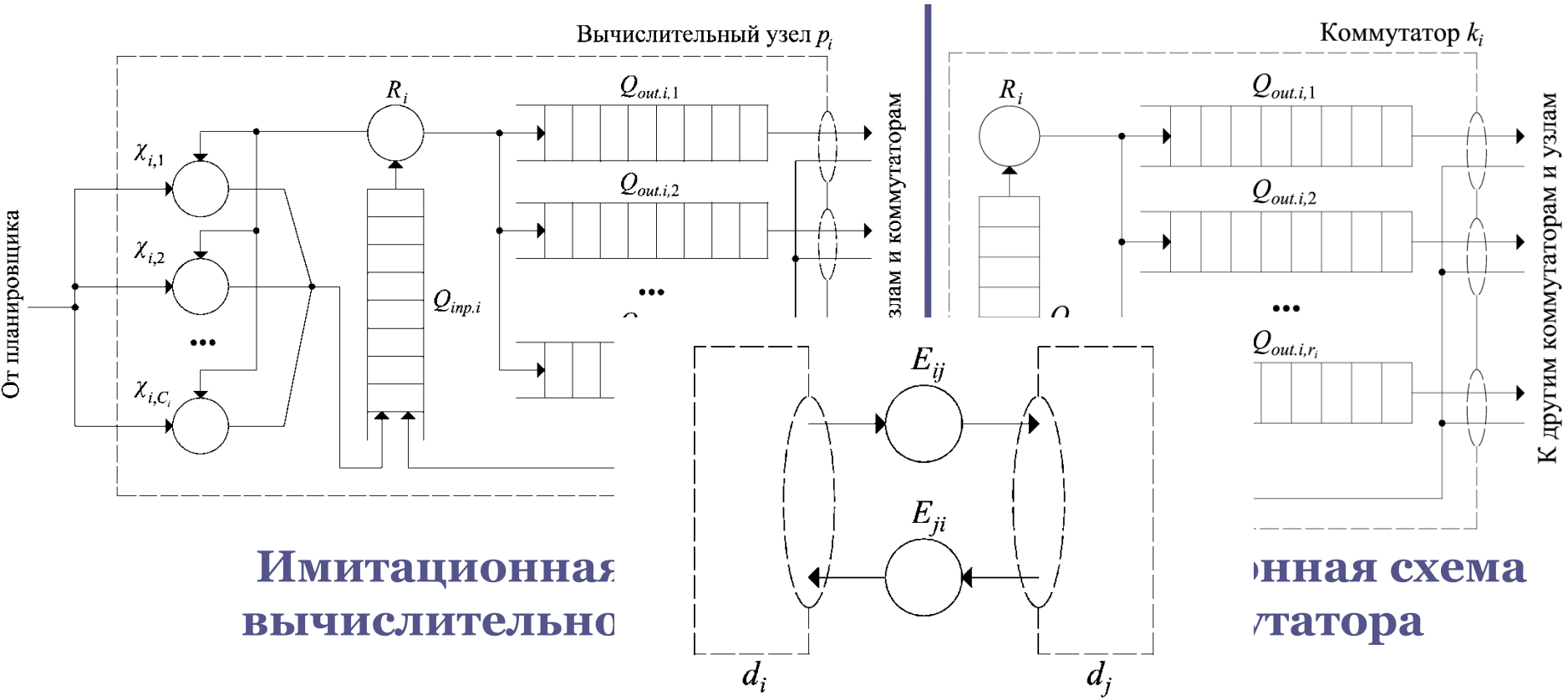
- Возможность задания **произвольной топологии** в виде взвешенного ориентированного графа
- **Поддержка коммутаторов**
- **Поддержка неоднородных узлов** (по оперативной и дисковой памяти, количеству вычислительных ядер, их производительности) и **неоднородной сети**
- Использование **реалистичной модели для генерации потока вычислительных задач**
- Генерация типовых **коммуникационных паттернов** взаимодействия процессов задач

Имитационная схема вычислительного кластера и его управляющей системы



Имитационная схема управляющеи системы вычислительного кластера

Имитационная схема вычислительного кластера и его управляющей системы (продолжение)



Имитационная … вычислительно

…нная схема …татора

Схема дуплексной связи

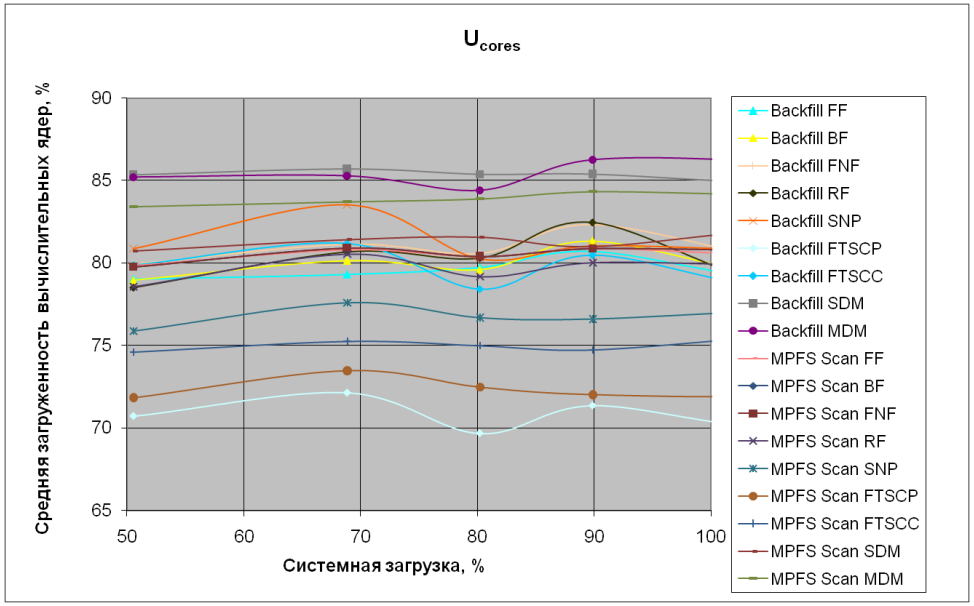
Система критериев и метрик сравнения алгоритмов планирования

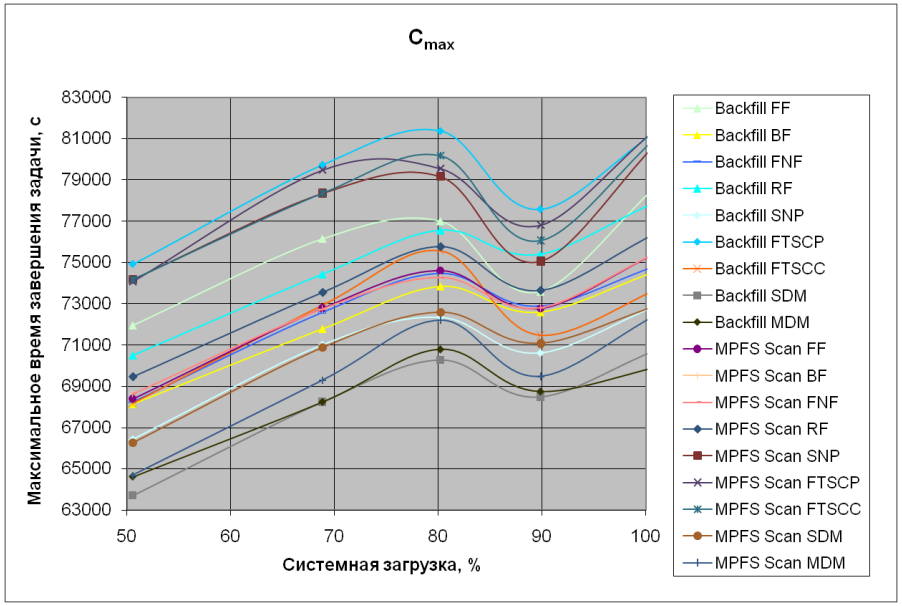
- Производительность расписаний
- Используемости оперативной и дисковой памяти узлов
- Сбалансированность загрузки узлов
- Гарантированность обслуживания задач
- Коммуникационный критерий

Характеристики проводимого экспериментального исследования

- **Две группы сценариев**:
  - вычислительный кластер с однородными узлами и однородной сетью
  - вычислительный кластер с неоднородными узлами и однородной сетью
- **Исследуемые топологии:**
  - толстое дерево
  - двумерный тор
  - звезда
- **Усреднение значения метрик** путем повторения симуляции 100 раз по 500 сгенерированных задач

Графики зависимости метрик производительности





Сценарий толстого дерева с неоднородными вычислительными узлами

$L \gtrsim$

Результаты исследования

Группа сценариев	Топология	Наилучшие алгоритмы
Неоднородные вычислительные узлы и однородная сеть	Толстое дерево	Backfill SDM (при L < 85-93%), Backfill MDM (при L >= 85-93%)
	Двумерный Тор	Backfill SDM (при L < 67-71%), Backfill MDM (при L >= 67-71%)
	Звезда	Backfill SDM, Backfill MDM
Однородные вычислительные узлы и однородная сеть	Толстое дерево	Backfill SDM (при L < 75-79%), Backfill MDM (при L >= 75-79%)
	Двумерный Тор	Backfill SDM (при L < 71-73%), Backfill MDM (при L >= 71-73%)
	Звезда	Backfill SDM, Backfill MDM, Backfill SNP

Планы на будущее

- Реализация алгоритмов в рамках Torque и их **апробация на реальных потоках задач**
- Исследование работы алгоритмов **для случая кластеров рабочих станций и вычислительных гридов**

Вопросы?

Summed Distance Minimization (SDM)

- Для каждого допустимого окна планирования $\boldsymbol{W}$ перебирает все сетевые устройства $\boldsymbol{d}$ кластера (коммутаторы и узлы) и определяет для каждого из них $\boldsymbol{n_j}$ ближайших вычислительных ядер окна $\boldsymbol{W}$, подходящих по конфигурации для задачи $\boldsymbol{J_j}$. Эти вычислительные ядра формируют возможное окно запуска $\boldsymbol{R_{W,d}}$ задачи $\boldsymbol{J_j}$
- Для назначения в качестве результата выбирается окно $\boldsymbol{R_{W,d}}$ с минимальным суммарным попарным расстоянием, если таких несколько, то выбирается окно с большей суммарной производительностью вычислительных ядер

Maximum Distance Minimization (MDM)

- Для каждого допустимого окна планирования W перебирает все сетевые устройства d кластера и для каждого из них запускает алгоритм поиска в ширину. В процессе его работы формируется множество достигнутых вычислительных ядер узлов окна W, которые подходят по конфигурации для задачи J_j, до тех пор, пока не будет получено n_j ядер. При этом среди ядер, находящихся на одинаковом расстоянии от начального сетевого устройства в первую очередь выбираются те, которые имеют большую скорость. Результатом работы поиска в ширину является сформированное возможное окно запуска $R_{W,d}$
- Для назначения задаче J_j в качестве результата выбирается окно $R_{W,d}$ с минимальнальным максимальным расстоянием от первоночального сетевого устройства d