**BC/NW 2017 № 1 (30):3.5**

**АНАЛИЗ МЕТОДОВ ОПТИМИЗАЦИИ УРОВНЯ ЗАГРУЗКИ РЕСУРСОВ СЕРВЕРНОГО ПАРКА**

Осипов А.В., Оцоков Ш.А.

Одним из современных подходов в оптимизации ресурсов серверного парка является применение механизмов виртуализации. Поиск оптимального распределения является задачей дискретной оптимизации.

Входными данными будут являться: множество типов ресурсов серверного парка {1,.,NC}; множество серверов {Hk};множество логических серверов {Sj}; а также матрица их требований {Qi,j}, j=1..NS, i=1..NC; дополнительные ограничения {Ed,j}, d=1..NX, i=1..NC; вектор приоритетов ресурсов {Q}, i=1..NC.

Целью оптимизации является минимизация числа физических серверов путем максимизации коэффициента загрузки ресурсов тех серверов, которые уже задействованы.

Для того, чтобы решить задачу многокритериальной оптимизации, к которой мы пришли, необходимо отказаться от попыток охватить все комбинации физических и логических серверов одновременно. Нужно разбить задачу на множество более простых подзадач.

Простейшим способом решить поставленную задачу является полный перебор. Но большая трудоемкость делает его нецелесообразным. Рациональней будет разбить задачу оптимизации на подзадачи локальной оптимизации для каждого из серверов без учета глобальных целей оптимизации. Для каждого физического сервера при этом ставится локальная задача оптимизации. Объем перебора становится меньше. Однако на каждом шаге теряются все серверы, кроме рассматриваемого.

Наиболее популярным методом решения поставленной задачи является метод динамического программирования Беллмана, [1] который обеспечивает глобальную оптимизацию при разбиении одной сложной задачи на множество простых. Однако и в этом методе число шагов может сильно колебаться. Глобальная целевая функция не может быть сформулирована из-за неизвестного числа задействованных серверов.

Таким образом, в чистом виде появляются трудности для его применения.

Для решения задачи также применимы методы случайного поиска, гриди-методы, метод локального поиска [2]. Однако их анализ приводит к выводу, что наилучшим методом оптимизации загрузки ресурсов станет комбинация из нескольких методов и и возможная их модификация.

**Литература**

1. Беллман Р., Дрейфус С. Прикладные задачи динамического программирования. М.: Наука, 1965.

2. Беллман Р., Калаб Р. Динамическое программирование и современная теория управления. М.: Наука 1969.

**ТЕМА:АНАЛИЗ МЕТОДОВ ОПТИМИЗАЦИИ УРОВНЯ ЗАГРУЗКИ РЕСУРСОВ СЕРВЕРНОГО ПАРКА**

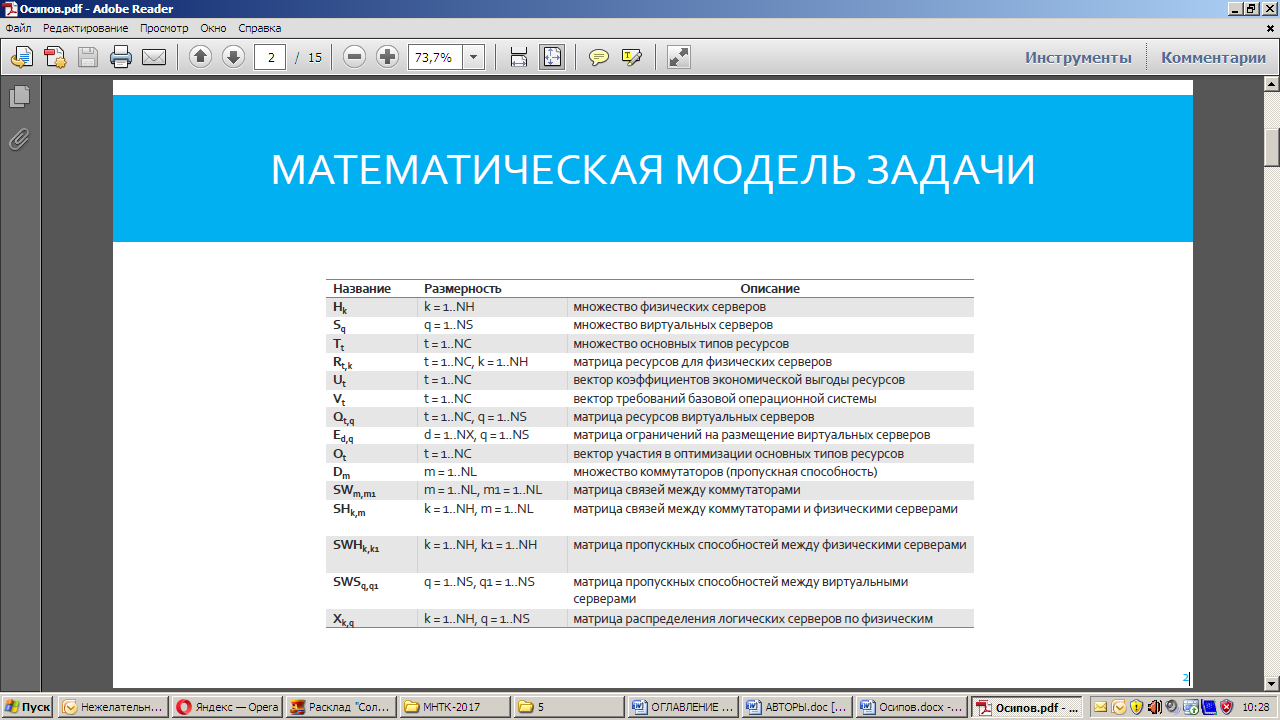
Аспирант: Осипов А.В.

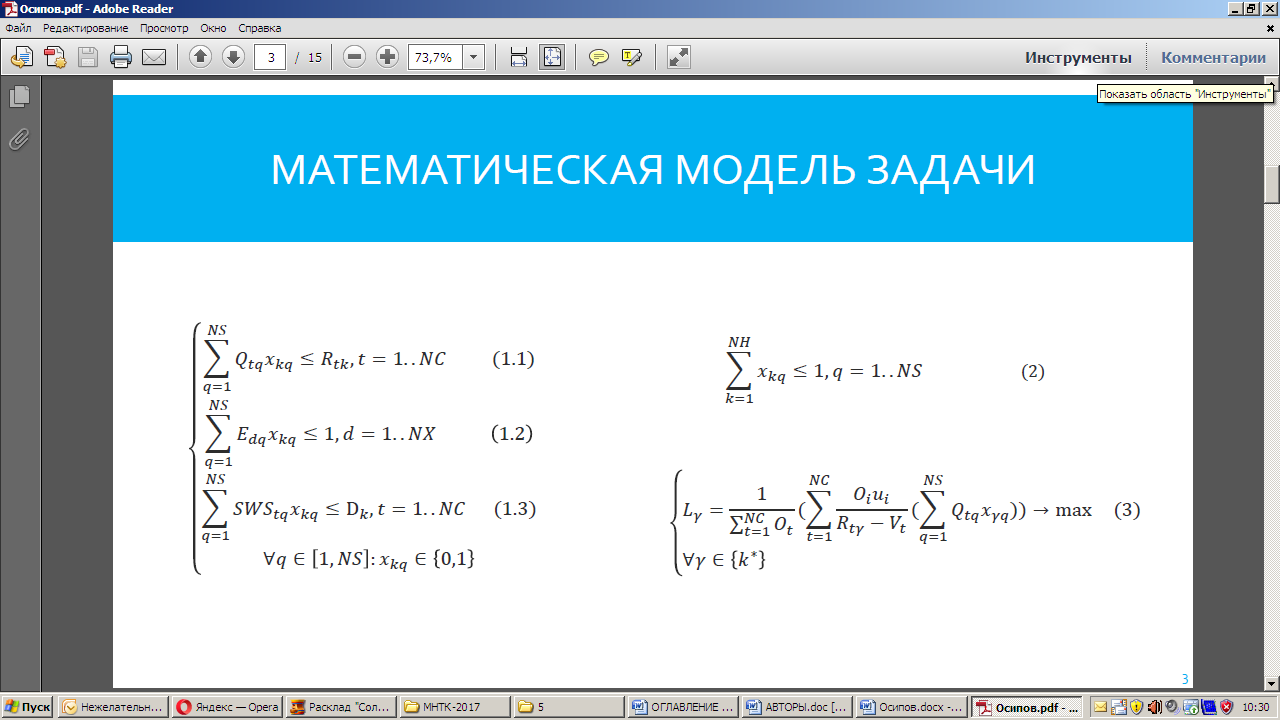
Научный руководитель: ЛадыгинИ.И.

ОцоковШ.А.

Москва, 2017

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ЗАДАЧИ





МЕТОДЫ РЕШЕНИЯ

1)Полный перебор

2)Разбиение на подзадачи

3)Динамическое программирование

4)Метод Беллмана

5)Приближённые методы решения

6)Жадные алгоритмы

7)Метод случайного поиска

ПОЛНЫЙ ПЕРЕБОР

Объём перебора: (NH+1)𝑁𝑆, где NH–кол-во физических серверов

NS–кол-во виртуальных серверов

•Применим при небольшом количестве физических и логических серверов

•Даёт точный глобальный оптимум

•Большая трудоёмкость при большом количестве физических и логических серверов

РАЗБИЕНИЕ НА ПОДЗАДАЧИ

Объём перебора для полного перебора: NH∗2𝑁𝑆

•Снижает трудоёмкость решения за счёт исключения из рассмотрения физических серверов

•Получаемые решения не глобально оптимальны из-за неполного рассмотрения вариантов на каждом шаге

ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

Показатель эффективности U = u1+ … + un

•Обеспечение глобального оптимума

•Применим только если показатель эффективности задачи обладает свойством аддитивности

МЕТОД БЕЛЛМАНА

Управление на каждом шаге надо выбирать так, чтобы оптимальной была сумма выигрышей на всех оставшихся до конца процесса шагах, включая выигрыш на данном шаге

Необходимо учитывать:

•Возможные исходы предыдущего шага

•Влияние текущего управления на все оставшиеся до конца процесса шаги

•Обеспечение глобального оптимума

•Неизвестно количество шагов для нашей задачи

МОДИФИКАЦИЯ МЕТОДОВ

Воспользуемся разбиением на подзадачи, но на каждом шаге будем выбирать не просто следующий физический сервер, а рассматривать все оставшиеся

•Во время работы алгоритма, не теряются из виду оставшиеся физические серверы, в отличие от обычного разбиения на подзадачи

•Не всегда находится глобальный оптимум

ПРИБЛИЖЁННЫЕ МЕТОДЫ РЕШЕНИЯ

X = (x1, … ,xn), ∀𝑗∈1,𝑁:𝑥𝑗∈{0,1} -Точка

D(XP, XQ) = σ𝑗=1𝑁|𝑥𝑗𝑃−𝑥𝑗𝑄| -Расстояние

Wr(X0)-Сфера поиска с центром X0 и радиусом r

Zr(X0)-Зона поиска с центром X0 и радиусом r.

Включает в себя все сферы с центром вX0с радиусами от 0до r

X\*∈Zr(X0) еслиD(X\*, X0) ≤ r10

ЖАДНЫЕ АЛГОРИТМЫ

Объём перебора: ~NS2

•Снижает трудоёмкость решения за счёт исключения из рассмотрения последующих шагов

•Получаемые решения не глобально оптимальны

•Рассчитаны на одно ограничение

МЕТОД СЛУЧАЙНОГО ПОИСКА

•Снижает трудоёмкость решения

•Не попадает в локальный экстремум безвозвратно, может найти новый локальный экстремум

•Может пропустить удачные решения, потому как шаги выбираются случайным образом

•Получаемые решения могут быть не глобально оптимальны

МЕТОД ЛОКАЛЬНОГО ПОИСКА

Объём перебора: ≤(NS+1)2

•Быстрое нахождение экстремума

•Возможность соблюсти ограничения введением функции штрафов

•Повышение вероятности нахождения глобального экстремума, путём многократного запуска алгоритма

•Снижение трудоёмкости решения

•Велика вероятность попасть в локальный экстремум, не найдя глобальный

•Получаемые решения могут быть не глобально оптимальны