

Оптимизация подхода к online-планированию в мультиагентной среде

Кудинкин Алексей

Кафедра Математических и Информационных Технологий
Санкт-Петербургский Академический Университет
Санкт-Петербург

3 июня, 2013

Научный руководитель: Кошкин Е. В., СПбГУ

Цель:

Оптимизировать процедуру распределения задач исполнителям в мультиагентной среде в условиях априорной неопределенности на примере системы непрерывной интеграции.

Задачи:

- 1 Исследовать подходы применяемые в области планирования (англ. *scheduling theory*).
- 2 Сформулировать критерии оценки качества распределения.
- 3 Выявить подходы улучшающие качество распределения.
- 4 Реализовать предложенные решения.
- 5 Сравнить предложенные решения с имеющимся.

Continuous Integration

Что же такое *Continuous Integration*?

Continuous Integration

Что же такое *Continuous Integration*?

Непрерывная интеграция (англ. *Continuous Integration*^a, CI) – это практика разработки ПО, которая заключается в выполнении частых автоматизированных сборок проекта для скорейшего выявления и разрешения *интеграционных* проблем.

^ahttp://en.wikipedia.org/wiki/Continuous_integration

Continuous Integration



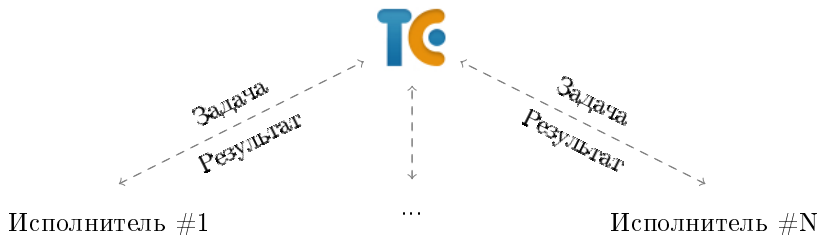
TeamCity – одно из решений в области *continuous-integration* от компании JetBrains.

Continuous Integration



TeamCity – одно из решений в области *continuous-integration* от компании JetBrains.

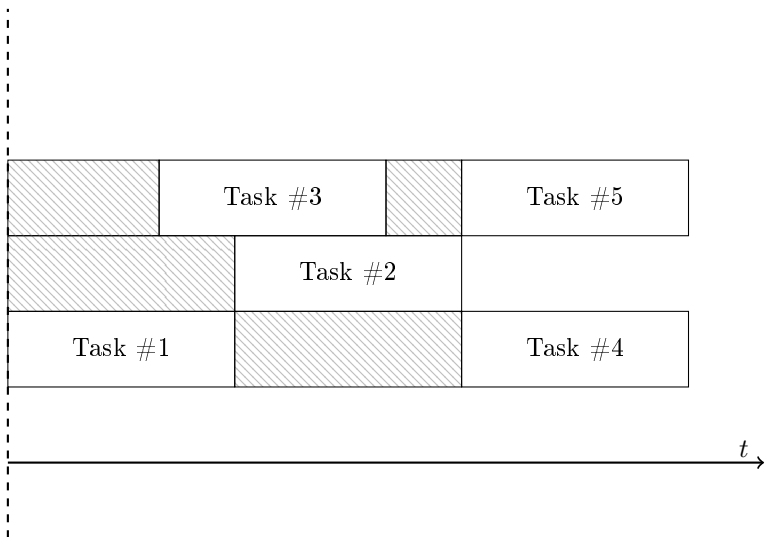
TeamCity реализует концепцию *computational-grid*:



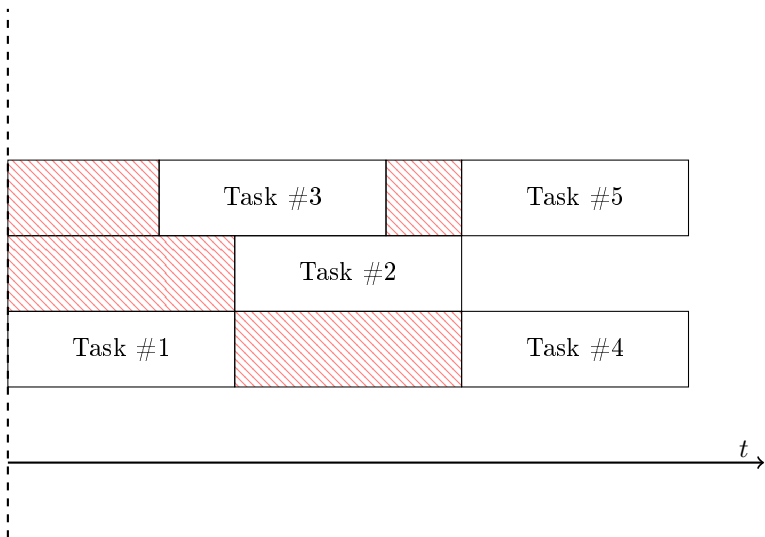
Мотивация

- Быстро идентифицировать *регрессии*.
- Быстро идентифицировать *изменения*, вводящие регрессии.
- Поддерживать *консистентность* исходного кода.

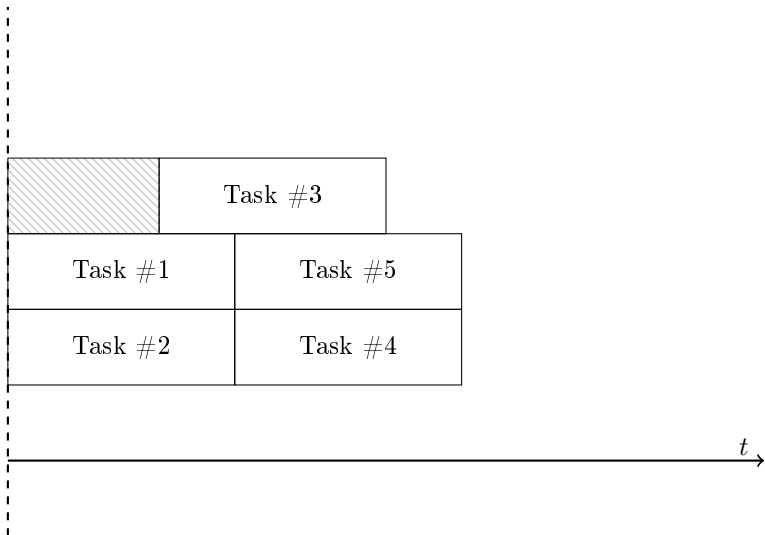
Планирование



Планирование



Планирование



Определения

J (**J**obs)

Множество задач.

Определения

J (**J**obs)

Множество задач.

E (**E**xecutors)

Множество исполнителей.

Определения

J (**J**obs)

Множество задач.

E (**E**xecutors)

Множество исполнителей.

$r_j, j \in J$ (*release-date*)

Момент времени, когда задача j становится доступной для исполнения.

Определения

J (**J**obs)

Множество задач.

E (**E**xecutors)

Множество исполнителей.

$r_j, j \in J$ (*release-date*)

Момент времени, когда задача j становится доступной для исполнения.

$C_j, j \in J$ (*completion-time*)

Момент времени, когда задача j завершает свое исполнение.

Критерии

Необходимо составить расписание $\hat{S}(J, E)$, удовлетворяющее следующему условию:

$$\hat{S} = \arg \min_S R(S)$$

Критерии

Необходимо составить расписание $\hat{S}(J, E)$, удовлетворяющее следующему условию:

$$\hat{S} = \arg \min_S R(S)$$

$$\mathbb{C}_{avg} : R(S) = \sum_{j \in J(S)} C_j$$

$$\mathbb{C}_{max} : R(S) = \max_{j \in J(S)} C_j$$

Ограничения

- На множестве задач J определено отношение *предшествования* R_{PREC} :

$$(j, k) \in R_{PREC} \rightarrow r_k \geq C_j \mid j, k \in J$$

Ограничения

- На множестве задач J определено отношение *предшествования* R_{PREC} :

$$(j, k) \in R_{PREC} \rightarrow r_k \geq C_j \mid j, k \in J$$



Ограничения

- На множестве задач J определено отношение *предшествования* R_{PREC} :

$$(j, k) \in R_{PREC} \rightarrow r_k \geq C_j \mid j, k \in J$$



- Отсутствует *вытеснение* (англ. *preemption*) задач.

Ограничения

- На множестве задач J определено отношение *предшествования* R_{PREC} :

$$(j, k) \in R_{PREC} \rightarrow r_k \geq C_j \mid j, k \in J$$



- Отсутствует *вытеснение* (англ. *preemption*) задач.
- Совместимость задачи j и исполнителя e определяется отношением *совместимости* R_{COMPAT} :

$$(j, e) \in R_{COMPAT} \leftrightarrow t_j^e < \infty \mid j \in J, e \in E$$

Ограничения

- На множестве задач J определено отношение *предшествования* R_{PREC} :

$$(j, k) \in R_{PREC} \rightarrow r_k \geq C_j \mid j, k \in J$$



- Отсутствует *вытеснение* (англ. *preemption*) задач.
- Совместимость задачи j и исполнителя e определяется отношением *совместимости* R_{COMPAT} :

$$(j, e) \in R_{COMPAT} \leftrightarrow t_j^e < \infty \mid j \in J, e \in E$$

- Исполнители *существенно* не однородны.

Используемый подход

Текущий подход к решению задачи планирования:

- Основан на стратегии FIFO (First-In – First-Out).

Используемый подход

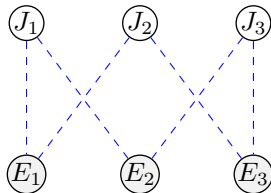
Текущий подход к решению задачи планирования:

- Основан на стратегии FIFO (First-In – First-Out).
- Назначает задачу *незанятому* исполнителю, требующему наименьшее время для ее исполнения.

Исследованные подходы

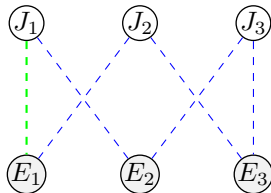
- Адаптированный *list-algorithm* (Graham, 1966).
- Адаптированный *virtual circuit routing algorithm* (Azar et. al., 1992).
- Сведение к ‘задаче о мастерской’.
- Сведение к ‘задаче о назначениях’.

Проблема #1



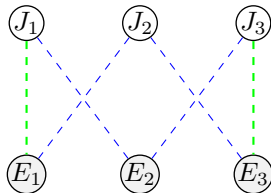
J_1, J_2, J_3 представляют собой задачи;
 E_1, E_2, E_3 представляют собой исполнителей;

Проблема #1



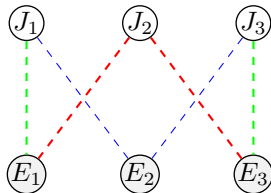
J_1, J_2, J_3 представляют собой задачи;
 E_1, E_2, E_3 представляют собой исполнителей;

Проблема #1



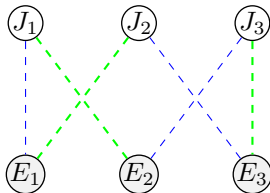
J_1, J_2, J_3 представляют собой задачи;
 E_1, E_2, E_3 представляют собой исполнителей;

Проблема #1



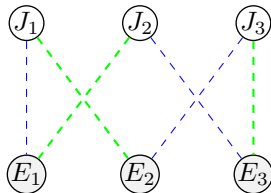
J_1, J_2, J_3 представляют собой задачи;
 E_1, E_2, E_3 представляют собой исполнителей;

Проблема #1



J_1, J_2, J_3 представляют собой задачи;
 E_1, E_2, E_3 представляют собой исполнителей;

Проблема #1



J_1, J_2, J_3 представляют собой задачи;

E_1, E_2, E_3 представляют собой исполнителей;

При каждом назначении планировщик *должен* стремиться распределить *максимальное* (по включению) подмножество *множества* задач J_t .

Сведение к задаче о назначениях

Дополним множество исполнителей E псевдо-исполнителями $e \in E_{CF}$, совместимыми с каждой из задач $j \in J_t$, причем:

$$t_j^e = \infty \mid j \in J_t, e \in E_{CF}$$

Сведение к задаче о назначениях

Дополним множество исполнителей E псевдо-исполнителями $e \in E_{CF}$, совместимыми с каждой из задач $j \in J_t$, причем:

$$t_j^e = \infty \mid j \in J_t, e \in E_{CF}$$

Assignment-Problem-Reduction

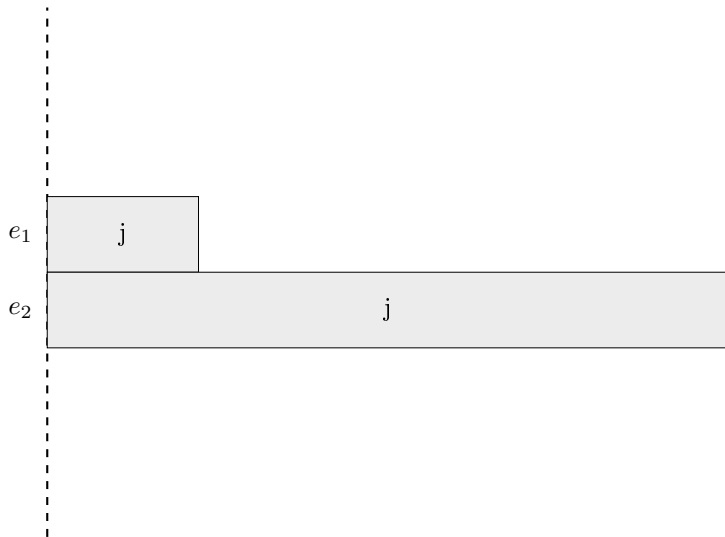
Сконструируем *двудольный граф*

$$G = (J_t, E \cup E_{CF}, R_{COMPAT})$$

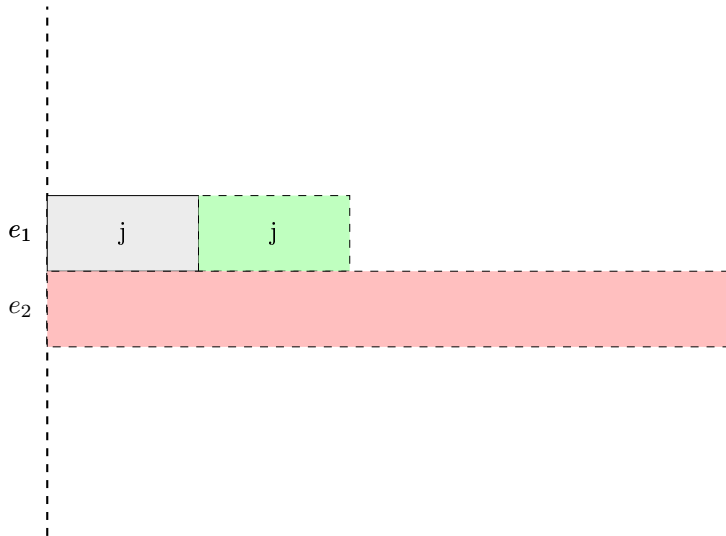
в котором пара вершин (j, e) соединена ребром *iff* задача j может быть выполнена исполнителем e . Вес ребра:

$$\omega_{(j,e)} = t_j^e$$

Проблема #2



Проблема #2



Сведение к задаче о назначениях 2.0

Заменяем множество исполнителей E множеством *пар*:

$$P = \{(e, k) \mid e \in E\}$$

где k – *порядковый* номер задачи в расписании для исполнителя e .

Сведение к задаче о назначениях 2.0

Заменим множество исполнителей E множеством *пар*:

$$P = \{(e, k) \mid e \in E\}$$

где k – *порядковый* номер задачи в расписании для исполнителя e .

$$k \leq |J_t^e| : J_t^e = \{j \in J_t \mid (j, e) \in R_{\text{COMPAT}}\}$$

Сведение к задаче о назначениях 2.0

Заменим множество исполнителей E множеством par :

$$P = \{(e, k) \mid e \in E\}$$

где k – *порядковый* номер задачи в расписании для исполнителя e .

$$k \leq |J_t^e| : J_t^e = \{j \in J_t \mid (j, e) \in R_{COMPAT}\}$$

Assignment-Problem-Reduction-Revisited

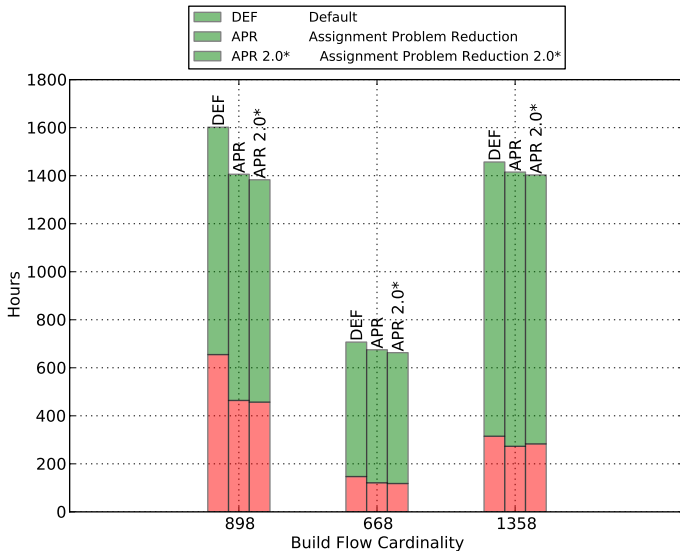
Сконструируем *двудольный граф*

$$G = (J_t, P, R_{COMPAT})$$

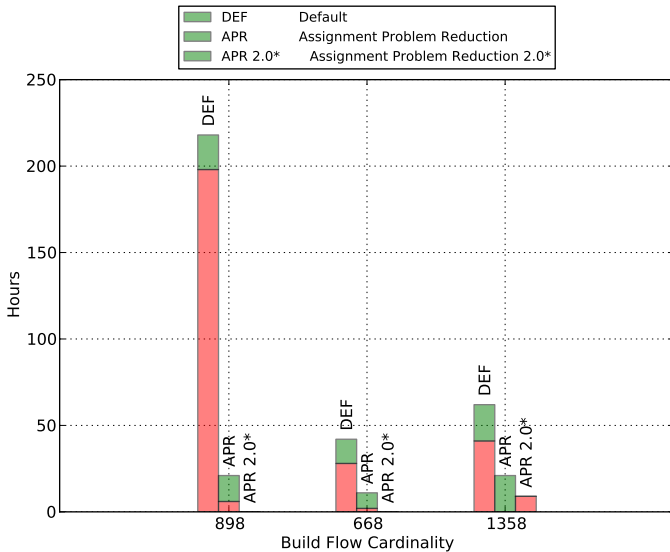
в котором пара вершин (j, p) соединена ребром *iff* задача j может быть выполнена исполнителем $p.e$.

$$\omega(j, p) = t_j^{p.e} \times p.k$$

Моделирование



Моделирование



Результаты

- Проведено исследование подходов применяемых в онлайн-планировании.
- Выявлены подходы, повышающие качество планирования в рассматриваемой системе.
- Некоторые решения реализованы.
- Проведено моделирование с целью сравнения эффективности различных подходов.

Спасибо за внимание!