

Метрики эффективности распределения человеческих ресурсов

Задача планирования загрузки ресурсов в последнее время становится все более популярной и значимой. Распределение ресурсов в любой предметной области стремится быть равномерным. Все существующие труды и исследования направлены на распределение, касающееся машинных ресурсов. К настоящему времени уже существует большое количество алгоритмов планирования, многие из которых были разработаны еще в двадцатом веке. Все подобные алгоритмы основаны на автоматизации планирования загрузки вычислительных мощностей, что, как известно, относится к точным наукам. В данной предметной области зачастую отсутствует возможность разбиения задач на подзадачи, срок выполнения задачи ядром или слотом является точным и диктуется его производительностью. Я же хочу рассмотреть автоматизацию планирования загрузки именно человеческих ресурсов на примере процесса разработки программного обеспечения. Наиболее популярным и широко используемым программным обеспечением в данной сфере является решение от компании Microsoft – Project. Его алгоритмы основаны на методе критических работ. Отличительными особенностями рассматриваемой мной предметной области являются волонтаристский срок задач и возможность рефакторинга задач. Целью данной работы является постановка метрик для оценки эффективности распределения человеческих ресурсов.

Для начала необходимо выполнить постановку задачи и сформулировать ограничения по загрузке ресурсов и планированию задач. Существует пул задач для разработки.

Пусть сущность задачи будет w_i , тогда множество задач $W = \{w_i \text{ где } i \in [1; N_w]\}$, где N_w - количество задач. Каждая задача определяется и описывается определенными атрибутами. Объект задача определяется следующим кортежем: $w_i = (dt_s, dt_f, t, e, us, nm, pred, pos, p, dt_{min}, dt_{max})$, где:

- dt_s – дата начала задачи;
- dt_f – дата окончания задачи;
- t – тип задачи (аналитика, разработка, тестирование);

- e – исполнитель задачи;
- us – уровень сложности задачи;
- nm – название задачи (идентификатор);
- $pred$ – предшественник задачи;
- pos – последователь задачи;
- p – приоритет задачи;
- dt_{min} – наиболее ранняя дата начала задачи;
- dt_{max} – крайняя дата внедрения задачи.

Свойство задачи e является отдельным объектом, который составляет множество $E = \{e_j \text{ где } j \in [1; N_e]\}$, где N_e - количество исполнителей.

Сущность исполнитель описывается следующим кортежем свойств: $e_j = (nm, k, u, z, pr, o, prz, kv)$, где:

- nm – имя исполнителя (идентификатор);
- k – область специализации исполнителя;
- u – загруженность исполнителя;
- z – заработная плата исполнителя;
- pr – производительность исполнителя;
- o – отпуск исполнителя;
- prz – стабильность производительности исполнителя;
- kv – уровень квалификации исполнителя.

Исходя из постановки задачи и экспертных мнений, я считаю следующие метрики оценки наиболее эффективными для бизнеса при планировании загрузки человеческих ресурсов:

1. Средняя загруженность исполнителей U . Средняя загруженность должна стремиться к U_{nom} (задачи должны равномерно распределяться между ресурсами).

$$U = \frac{\sum_{i=1}^{N_w} U_i}{N_e} \longrightarrow U_{nom}$$

2. Индекс сбалансированности загрузки исполнителей σ . Индекс сбалансированности должен стремиться к минимуму.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N_w} (U_i - U)^2}{N_e - 1}} \longrightarrow \min$$

3. Стоимость проекта. Стоимость проекта должна стремиться к минимуму.

$$Cost = \sum_{i=1}^{N_w} (dt_f(w_i) - dt_s(w_i)) * z_{e(w_i)} \longrightarrow \min$$

4. Сроки проекта. Крайняя дата окончания задач должна стремиться к минимуму.

$$\max(dt_f(w_i)) \longrightarrow \min$$

Таким образом, в данном докладе выполнена постановка прикладной задачи, рассмотрены алгоритмы теории графов и метрики для оценки распределения ресурсов. Используя описанные выше метрики необходимо понимать, что оценка эффективности алгоритмов планирования человеческих ресурсов с помощью комбинации данных метрик сводится в первую очередь к компромиссу стоимости и сроков проекта.

Текст.

Задача планирования загрузки ресурсов в последнее время становится все более популярной и значимой. Распределение ресурсов в любой предметной области стремится быть равномерным. Все существующие труды и исследования направлены на распределение, касающееся машинных ресурсов. К настоящему времени уже существует большое количество алгоритмов планирования, многие из которых были разработаны еще в двадцатом веке. Все подобные алгоритмы основаны на автоматизации планирования загрузки вычислительных мощностей, что, как известно, относится к точным наукам. В данной предметной области зачастую отсутствует возможность разбиения задач на подзадачи, срок выполнения задачи ядром или слотом является точным и

диктуется его производительностью. Я же хочу рассмотреть автоматизацию планирования загрузки именно человеческих ресурсов на примере процесса разработки программного обеспечения. Наиболее популярным и широко используемым программным обеспечением в данной сфере является решение от компании Microsoft – Project. Его алгоритмы основаны на методе критических работ. Отличительными особенностями рассматриваемой мной предметной области являются волонтаристский срок задач и возможность рефакторинга задач. Целью данной работы является постановка метрик для оценки эффективности распределения человеческих ресурсов.

Для начала необходимо выполнить постановку задачи и сформулировать ограничения по загрузке ресурсов и планированию задач. Существует пул задач для разработки.

Пусть сущность задачи будет w_i , тогда множество задач $W = \{w_i \text{ где } i \in [1; N_w]\}$, где N_w - количество задач. Каждая задача определяется и описывается определенными атрибутами. Объект задача определяется следующим кортежем: $w_i = (dt_s, dt_f, t, e, us, nm, pred, pos, p, dt_{min}, dt_{max})$, где:

- dt_s – дата начала задачи;
- dt_f – дата окончания задачи;
- t – тип задачи (аналитика, разработка, тестирование);
- e – исполнитель задачи;
- us – уровень сложности задачи;
- nm – название задачи (идентификатор);
- $pred$ – предшественник задачи;
- pos – последователь задачи;
- p – приоритет задачи;
- dt_{min} – наиболее ранняя дата начала задачи;
- dt_{max} – крайняя дата внедрения задачи.

Свойство задачи e является отдельным объектом, который составляет множество $E = \{e_j \text{ где } j \in [1; N_e]\}$, где N_e - количество исполнителей.

Сущность исполнитель описывается следующим кортежем свойств: $e_j = (nm, k, u, z, pr, o, prz, kv)$, где:

- nm – имя исполнителя (идентификатор);
- k – область специализации исполнителя;
- u – загруженность исполнителя;
- z – заработная плата исполнителя;
- pr – производительность исполнителя;
- o – отпуск исполнителя;
- prz – стабильность производительности исполнителя;
- kv – уровень квалификации исполнителя.

Свойство исполнителя o является отдельным объектом, который составляет множество $O = \{o_j \mid j \in [1; N_o]\}$, где N_o - количество отпусков.

Сущность отпуск описывается следующим кортежем свойств: $o_j = (dt_s, dt_f)$, где:

- dt_s – дата начала отпуска;
- dt_f – дата окончания отпуска.

Далее необходимо обозначить интересующие ограничения:

1. Дата начала задачи должна быть больше или равна дате наиболее раннего начала.

$$dt_s(w_i) \geq dt_{min}(w_i)$$

2. Дата окончания задачи должна быть меньше либо равна крайней дате внедрения.

$$dt_f(w_i) \leq dt_{max}(w_i)$$

3. Дата начала следующей задачи должна быть больше либо равна дате окончания задачи.

$$dt_s(w_{i+1}) \geq dt_f(w_i)$$

4. Дата начала задачи не должна быть в период отпуска.

$$dt_s(w_i) \notin [dt_s(o(e_i)); dt_f(o(e_i))]$$

5. Дата окончания задачи не должна быть в период отпуска.

$$dt_f(w_i) \notin [dt_s(o(e_i)); dt_f(o(e_i))]$$

6. Дата окончания задачи должна быть равна дате начала задачи последователя.

$$dt_f(w_i) = dt_s(w_{pos(w_i)})$$

7. Дата начала задачи должна быть равна дате окончания задачи предшественника.

$$dt_s(w_i) = dt_f(w_{pred(w_i)})$$

8. Если приоритет задачи выше приоритета следующей, то дата окончания задачи должна быть меньше даты окончания следующей задачи.

$$\text{Если } p(w_i) > p(w_{i+1}), \text{ то } dt_f(w_i) < dt_f(w_{i+1})$$

Для решения задачи распределения ресурсов используется теория графов. Наиболее популярными алгоритмами для нахождения критического пути являются:

- 1) KB/L
- 2) Саркара
- 3) DSC (Dominant Sequence Clustering)

Общим для всех алгоритмов является завершение в момент, когда рассмотрены все дуги графа, а целевыми функциями является минимизация параллельного времени графа. Данные алгоритмы обычно рассматриваются в контексте выполнения задачи ядром, но даже если применять их к человеческим ресурсам, все они предполагают выполнение одной задачи одним исполнителем. Описанный выше факт значительно ограничивает использование подобных методов при реальном распределении человеческих ресурсов.

В трудах Кима (Kim S.J. A general approach to multiprocessor scheduling. Report TR-88-04. Department of Computer Science, University of Texas at Austin, 1988.) в качестве одной из метрик для оценки длины критического пути используется стоимостная функция:

$$Cost\ function = w_1 * \sum \tau_i + (1 - w_1) * (w_2 * \sum c_{ij} + (1 - w_2) * \sum c_{ij}^{adj}) \quad ,$$

где:

- w_1 и w_2 – нормализующие множители;
- $\sum \tau_i$ – сумма вычислительных стоимостей всех вершин критического пути;
- c_{ij}^{adj} – коммуникационная стоимость дуг между вершиной критического пути и всеми его смежными вершинами, не входящими в критический путь.

Исходя из постановки задачи и экспертных мнений, я считаю следующие метрики оценки наиболее эффективными для бизнеса при планировании загрузки человеческих ресурсов:

5. Средняя загруженность исполнителей U . Средняя загруженность должна стремиться к U_{nom} (задачи должны равномерно распределяться между ресурсами).

$$U = \frac{\sum_{i=1}^{N_w} U_i}{N_e} \longrightarrow U_{nom}$$

6. Индекс сбалансированности загрузки исполнителей σ . Индекс сбалансированности должен стремиться к минимуму.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N_w} (U_i - U)^2}{N_e - 1}} \longrightarrow min$$

7. Стоимость проекта. Стоимость проекта должна стремиться к минимуму.

$$Cost = \sum_{i=1}^{N_w} (dt_f(w_i) - dt_s(w_i)) * z_{e(w_i)} \longrightarrow min$$

8. Сроки проекта. Крайняя дата окончания задач должна стремиться к минимуму.

$$\max(dt_f(w_i)) \longrightarrow min$$

Таким образом, в данном докладе выполнена постановка прикладной задачи, рассмотрены алгоритмы теории графов и метрики для оценки распределения ресурсов. Используя описанные выше метрики необходимо понимать, что оценка эффективности алгоритмов планирования человеческих ресурсов с помощью комбинации данных метрик сводится в первую очередь к компромиссу стоимости и сроков проекта.

Коротков Александр Владимирович, 13.05.1992, ООО МКК «Домашние деньги онлайн», руководитель отдела информационных технологий, 89660411488, a.vl.korotkov@gmail.com