

Содержание

1	План	2
1.0.1	1 часть - ранжирование	2
1.0.2	2 часть - определение оптимальной стратегии исполнения	2
1.0.3	Дополнительно	3
2	Статьи и ссылки	4

1 План

1.0.1 1 часть - ранжирование

- Определена область исследования: направленные ациклические графы зависимости задач (DAG). В каждом графе присутствует финальный узел "finish" означающий завершение процесса выполнения задач.
- Получены 3 алгоритма ранжирования узлов графов зависимостей задач по влиянию на остальные задачи:
 1. simpleRank - алгоритм начинается с последней вершины (всегда с весом 1) и для каждого узла i рассчитывает вес как сумму весов всех исходящих узлов $+ 1$
 2. successorsRank - алгоритм рассчитывает вес пропорционально количеству всех зависимых задач
 3. pageRank - алгоритм основан на поиске собственных векторов P -матрицы для собственного значения 1. Для невырожденности матрицы граф зацикливается - финальный узел соединяется с начальными. Эта ситуация соответствует последовательному выполнению подобных процессов.
- Все алгоритмы, предположительно, дают топологическую сортировку - TODO исследовать
- Алгоритмы дают разную сортировку
- Вставить данные сравнения алгоритмов на графах 10, 100, 1000 узлов TODO

1.0.2 2 часть - определение оптимальной стратегии исполнения

- Реализована симуляция выполнения процесса по графу зависимостей в однопоточной среде со следующими параметрами:
 - Время выполнения задачи
 - Вероятность возникновения ошибки
 - Время исправления ошибки (опционально, считается » времени выполнения процесса)
- Введено понятие покрытия процесса: среднее отношение количества выполненных задач к количеству задач в процессе в результате симуляции
- При условии время исправления ошибки » времени выполнения процесса среднее покрытие не зависит от выбора топологической сортировки и может быть рассчитано аналитически через вероятности возникновения ошибок.

1.0.3 Дополнительно

- Добавление дополнительных факторов - вероятность падения ошибки, разные типы зависимостей (условная зависимость, "Finish to start "Start to start etc)
- Обобщение задачи на многопоточную систему - минимизирование времени проста (без перераспределения задач между потоками)
- Обобщение задачи на многопоточную систему - алгоритмы перераспределения задач при возникновении ошибки

2 Статьи и ссылки

DAG - [https : //ipyparallel.readthedocs.io/en/latest/dag_dependencies.html](https://ipyparallel.readthedocs.io/en/latest/dag_dependencies.html)

Топологическое упорядочивание <https://ru.wikipedia.org/wiki/>