



Санкт-Петербургский государственный университет
Кафедра системного программирования

Модификация протокола локального голосования для оптимизации балансировки загрузки ресурсов распределенной системы

Елена Александровна Борисоглебская, 21.M04-мм группа

Научный руководитель: д.ф.-м.н., Граничин О.Н., проф. каф. сис. прогр.

Санкт-Петербург
2023

Целью работы является разработка прототипа системы балансировки загрузки ресурсов распределенной системы.

Задачи:

- Формирование требований
- Разработка архитектуры
- Разработка ускоренного алгоритма локального голосования
- Разработка прототипа системы балансировки сети на основе модифицированного алгоритма локального голосования
- Проведение апробации системы

Почему мультиагентные технологии?

- Новые задачи поступают все время
- Неизвестно, сколько всего задач
- Неизвестно время появления новых задач
- Неизвестно, сколько новые задачи потребуют ресурсов
- Большое количество узлов распределенной системы

Изменение состояния агента описывается следующим уравнением:

$$q_{t+1}^i = q_t^i - p_t^i + z_t^i + u_t^i; \quad i = 1, \dots, n, \quad t = 0, 1, 2, \dots$$

- ❶ q_t^i — загруженность очереди или длина очереди из атомарных элементарных заданий узла i в момент времени t
- ❷ p_t^i — производительность узла i в момент времени t
- ❸ z_t^i — новое задание, поступившее на узел i в момент времени t
- ❹ u_t^i — результат перераспределения задач между узлами (добавление или уменьшение)

Взаимодействие внутри системы, состоящей из n элементов, можно описать с помощью ориентированного графа $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, где $\mathcal{N} = 1, \dots, n$ — множество вершин, а $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ — множество дуг

Для узла $i \in N$, множество соседей обозначим $N^i = \{j \in \mathcal{N} : (j, i) \in \mathcal{E}\}$. Полустепень входа узла $i \in N$ равна $|N^i|$.

Сопоставим дуге $(j, i) \in \mathcal{E}$ вес (стоимость передачи данных по дуге) $b^{i,j} < 0$ и $b^{i,j} = 0$ для $(j, i) \notin \mathcal{E}$. $B = [b^{j,i}]$ — взвешенная матрица смежности или матрица связности.

Взвешенная полустепень входа узла $i \in N$ равна $deg_i^+(B) = \sum_{j=1}^n b^{j,i}$. $deg_{max}^+(B)$ - наибольшая полустепень входа среди всех узлов в графе \mathcal{G} .

Матрица степеней D :

$$d_{i,j} = \begin{cases} deg_i^+(B), & i = j; \\ 0, & \text{иначе.} \end{cases}$$

Лапласиан графа связей

Пусть \mathcal{N} — набор агентов (узлов). Связь между узлами определяется графом $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ с матрицей смежности B . Обозначим через $W = \mathcal{L}(B) = D - B$ лапласиан графа G .

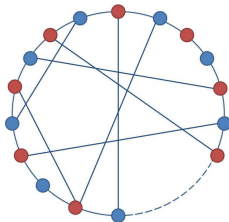


Рис.: Пример топологии сети

ПО

- Optimization Tool Matlab
- ADMB (AD Model Builder)
- CUTer (Constrained and Unconstrained Testing Environment, revisited)
- TOMLAB
- GAMS (General Algebraic Modeling System)

Пакеты

- GEKKO
- ALGLIB
- IMSL Numerical Libraries
- MIDACO
- IPOPT (Interior Point OPTimizer)

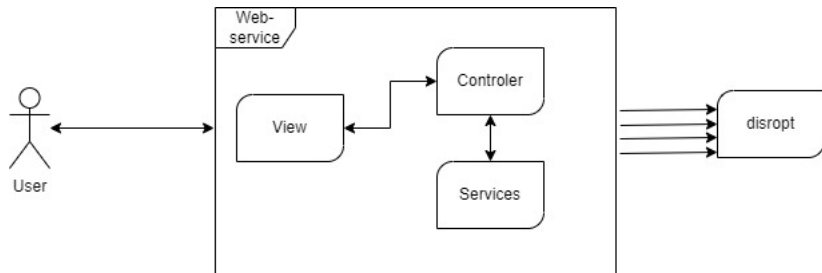
Требования к системе

- 1 Возможность задания количества узлов системы
- 2 Возможность задания модели измерения
- 3 Возможность задания количества шагов
- 4 Возможность задания матрицы связности
- 5 Возможность сравнения работы выбранных алгоритмов
- 6 Возможность задания шумов

Используемые инструменты

- 1 Python
- 2 Flask
- 3 Bootstrap
- 4 Disropt

Архитектура системы



```
1      # create local agent
2      agent = AgentLB(queue=queue,
3                      produc=5,
4                      in_neighbors=[2, 3],
5                      out_neighbors=[2, 3],
6                      in_weights=[0, 0, 0.5, 0.5, 0, 0])
7
8      # instantiate the consensus algorithm
9      algorithm = AcceleratedLocalVoting(
10         parameters=parameters,
11         agent=agent,
12         initial_condition=np.array([0]))
13
14     # run the algorithm
15     sequence = algorithm.run(iterations=100, verbose=True)
16
```

Оптимальное решение

Пусть необходимо выполнить z задач, q_i — задачи, которые выполняет i -ый узел. Тогда:

$$\sum_{i=1}^n q_i = z.$$

Время обработки задач i -ым узлом равняется: $x^i(q_i) = q_t^i / p_t^i$. При отсутствии новых заданий оптимальная загруженность системы будет достигнута при одновременном окончании работы всех узлов системы¹:

$$x^1 = x^2 = \dots = x^n. \quad (1)$$

¹N. Amelina, A. Fradkov, Y. Jiang, and D. J. Vergados, "Approximate consensus in stochastic networks with application to load balancing," IEEE Trans. Inf. Theory, vol. 61, no. 4, pp. 1739–1752, Apr. 2015.

Управляющее воздействие u_t может быть найдено следующим образом:

$$u_t = \operatorname{Argmin}_{u \in \mathbb{R}^n} F_t(u), \quad \forall t = 0, 1, \dots,$$

$$F_t(u) = \sum_{i,j \in \mathcal{N}} \frac{b^{ij}}{2} (x_t^i(u^i))^2 \rightarrow \min_u.$$

Управляющее воздействие u_t может быть найдено следующим образом:

$$u_t = \operatorname{Argmin}_{u \in \mathbb{R}^n} F_t(u), \quad \forall t = 0, 1, \dots,$$
$$F_t(u) = \sum_{i,j \in \mathcal{N}} \frac{b^{ij}}{2} \left(x_t^i(u^i) - x_t^j(u^j) \right)^2 + \sum_{i,j \in \mathcal{N}} \frac{b^{ij}}{p^2} u^i u^j -$$
$$-\frac{2}{p} \sum_{i,j \in \mathcal{N}} b^{ij} u_t^i \left(x_{t-1}^j + \frac{z_{t-1}^j}{p} - 1 \right) \rightarrow \min_u.$$

Градиент функции риска

Пусть для каждого агента i в момент времени t известны зашумленные измерения производной $\frac{d}{du^i} F_t(u)$:

$$\begin{aligned} g_t^i &= f_t^i(u^i) = \frac{d}{du^i} F_t(u) + \xi_t^i = \\ &= - \sum_{j \in \mathcal{N}} \frac{b_{ij}}{p} \left(x_t^i(u^i) - x_t^j(u^j) - \frac{w^j}{p} \right) - \frac{2}{p} \sum_{i,j \in \mathcal{N}} b^{ij} \left(x_{t-1}^j + \frac{z_{t-1}^j}{p} - 1 \right) + \xi_t^i, \\ &\quad t = 0, 1, \dots \end{aligned} \tag{2}$$

где $\xi_t^i \in \mathbb{R}$ — шум.

Условия накладываемые на функцию

- ❶ Для любого $k \geq 0$ должны существовать константы A и B такие, что:

$$\forall i \in \mathcal{N} \quad | -z_k^i + u_k^i + p | < A, \quad \left| x_k^j + \frac{z_k^j}{p} - 1 \right| < B.$$

- ❷ Шум ξ_k имеет нулевое математическое ожидание и ковариация шума в среднем ограничены:

$$\mathbb{E}\xi_k = 0, \mathbb{E}Q \leq \sigma_{\max}^2 I,$$

где Q - матрица ковариации случайного вектора ξ_k , I - единичная матрица, σ_{\max}^2 - максимальное собственное число матрицы Q .

- ❸ Граф \mathcal{G} должен быть связным.

Протокол локального голосования

Рассматриваемый протокол локального голосования:

$$u_t^i = h \sum_{j \in \mathcal{N}^i} b_{i,j} (y_t^{i,i} - y_t^{i,j}),$$

где $h > 0$ — размер шага протокола, p — продуктивность агентов.
 $b^{ij} = b < 0$, $\forall j \in \mathcal{N}^i$ и равняется нулю $b^{ij} = 0$ для остальных пар $i, j : j \notin \mathcal{N}^i$.

Модель наблюдений

Каждый агент i вычисляет свое значение x_t^i с помехой $\xi_t^{i,i}$, а так же получает зашумленные измерения соседей $j, j \in \mathcal{N}^i$, с некоторой целочисленной задержкой $d_t^{i,j}$ ($0 \leq d_t^{i,j} \leq \bar{d}$, где $\bar{d} \geq 0$ максимальная задержка):

$$\begin{aligned} y_t^{i,i} &= x_t^i + \xi_t^{i,i}, \\ y_t^{i,j} &= x_{t-d_t^{i,j}}^j + \xi_t^{i,j}, j \in \mathcal{N}^i. \end{aligned}$$

Задача найти последовательность оценок $\{\hat{u}_k\}_{k=0}^\infty$ таких, что

$$\exists N, C < \infty : \forall k > N \quad \mathbb{E} \|\hat{u}_k - u_k\|^2 \leq C. \quad (3)$$

Некоторые константы

Введем константы $L > 0$ и $\mu > 0$:

$$L = \frac{n+1}{p^4} \max_i \|b^{ij}\|^2, \quad \mu = \max_i \frac{2}{p^2} \sum_{j \in \mathcal{N}} b^{ij},$$

а также константы a, b, c :

$$b = \sum_{i,j \in \mathcal{N}} \frac{|b^{ij}|}{p} (4AB + 2A^2), \quad a = 2A\sqrt{n+1},$$
$$c = \frac{9A^2}{p^2} \sum_{i,j \in \mathcal{N}} (b^{ij})^2 + 1.$$

Алгоритм

Для каждого узла системы i рассмотрим следующий алгоритм:

- ❶ Выбрать $\hat{u}_0^i \in \mathbb{R}^d$ и $\gamma_0 > 0$, $v_0 = \hat{u}_0^i$.
- ❷ Выбрать $h > 0, \eta \in (0, \mu), \alpha_x \in (0, 1)$ такое, что неравенство (4) всегда выполнялось.
- ❸ Вычисляется значение $H_1 = h - \frac{h^2 L}{2}$.
- ❹ На k -ой итерации:
 - ❶ Найти $\alpha_k \in [\alpha_x, 1)$:

$$H_1 - \frac{\alpha_k^2}{2\gamma_{k+1}} > 0. \quad (4)$$

- ❷ Вычислить $\gamma_{k+1} = (1 - \alpha_k) * \gamma_k + \alpha_k * (\mu - \eta)$.
- ❸ Вычислить

$$z_k = \frac{\alpha_k \gamma_k v_k + \gamma_{k+1} \hat{u}_k^i}{\gamma_k + \alpha_k (\mu - \eta)}$$

и посчитать значение $Y_k(z_k)$.

- ❹ Вычислить \hat{u}_{k+1}^i :

$$\hat{u}_{k+1}^i = z_k - h Y_k(z_k).$$

- ❺ Вычислить $v_{k+1} = \frac{1}{\gamma_k} \left[(1 - \alpha_k) \gamma_k v_k + \alpha_k (\mu - \eta) z_k - \alpha_k Y_k(z_k) \right]$.

Сходимость функции потерь

Теорема² Если предположения 1–3 выполнены, то алгоритм, описанный выше, решает проблему (3) со следующими параметрами:

$$C = \frac{2}{\mu} D_{\infty},$$

$$D_{\infty} = \alpha_x^{-1} \left[\frac{2a + hc}{4\epsilon} + 2b + (1 - \alpha_x)(b + A_{\infty}c) + h^2 \frac{L}{2} \sigma^2 + \frac{c^2}{2\eta} \right],$$

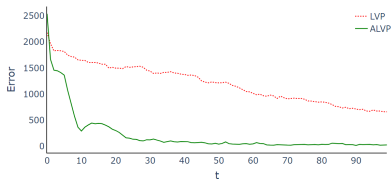
Ошибка оценки после конечного числа итераций ограничена:

$$\mathbb{E}_k F_k(\hat{u}_k) - F_k(u_k) \leq \prod_{i=1}^k (1 - \alpha_k)(\phi_0(u_0) - F_k(u_k) + \Phi) + D_k, \quad (5)$$

где $\phi_0(x) = F_0(\hat{u}_0) + \frac{\gamma_0}{2} \|x - v_0\|^2$, $\Phi = \frac{\gamma_0 c^2}{2\eta^2}$.

²D. Kosaty, A. Vakhitov, O. Granichin, and M. Yuchi, "Stochastic fast gradient for tracking," in American Control Conference (ACC). IEEE, pp. 1476–1481, 2019.

Сходимость алгоритма



[b]0.5

Рис.: Шаг алгоритма $h = 0.1$

[b]0.5 `picture/image_aver_0.2 (2).pr`

Рис.: Шаг алгоритма $h = 0.2$

Рис.: Средняя сходимость к консенсусу. Красная пунктирная линия соответствует алгоритму LVP. Зеленая сплошная линия показывает предлагаемую ускоренную версию.

Графический интерфейс системы

Load Balancing System

Input Form

Number of agents

Steps

Noise Generation

- ☒ None
☐ St. normal distr.
☐ Custom

Custom noise function

Matrix Generation

Default

Algorithms

- ☒ LVP
☒ ALVP

Matrix

LVP_Dynamic

ALVP_Dynamic

Error_comparison

0.0	0.0	0.5	0.5	0.0
0.0	0.0	0.0	0.5	0.5
0.5	0.0	0.0	0.0	0.5
0.5	0.5	0.0	0.0	0.0
0.0	0.5	0.5	0.0	0.0

Load Balancing System

Input Form

Number of agents

5

Steps

200

Noise Generation

- ☒ None
☐ St. normal distr.
☐ Custom

Custom noise function

Matrix Generation

Default

Algorithms

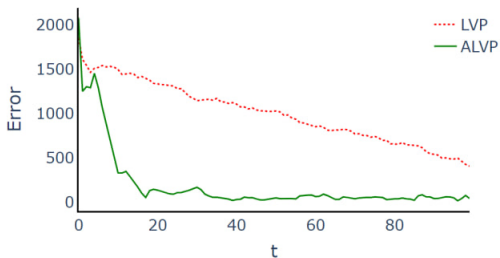
- ☒ LVP
☒ ALVP

Matrix

LVP_Dynamic

ALVP_Dynamic

Error_comparison



- Сформированы требования к разрабатываемой системе
- Разработана архитектура разрабатываемой системы
- Разработан модифицированный протокол локального голосования, ускоренный при помощи ускорения по Нестерову для задач трекинга
- Разработан прототип системы балансировки загрузки сети на основе модифицированного протокола локального голосования, позволяющая имитировать работу балансировщика сети
- Проведена апробация системы