Санкт-Петербургский государственный университет

Кафедра системного программирования

# Модификация протокола локального голосования для оптимизации балансировки загрузки ресурсов распределенной системы

Елена Александровна Борисоглебская, 21.М04-мм группа

**Научный руководитель:** д.ф.-м. н., Граничин О.Н., профессор кафедры системного программирования

Санкт-Петербург
2022

# Постановка задачи

**Целью** работы является модификация протокола локального голосования для оптимизации балансировки загрузки ресурсов распределенной системы.

**Задачи**:

- Реализовать существующий алгоритм балансировки загрузки сети
- Разработать модифицированный алгоритм локального голосования
- Доказать сходимость алгоритма

# Почему мультиагентные технологии?

- Новые задачи поступают все время;
- Не известно сколько всего задач;
- Не известно время появления новых задач;
- Не известно сколько новые задачи потребуют ресурсов;
- Большое количество узлов распределенной системы.

# Балансировка загрузки сети. Постановка задачи

Изменение состояния агента описывается следующим уравнением:

$$q^i_{t+1} = q^i_t - r^i_t + z^i_t + u^i_t; \quad i = 1, \ldots, n, \quad t = 0, 1, 2, \ldots,$$

1. $q^i_t$ — загруженность очереди или длина очереди из атомарных элементарных заданий узла $i$ в момент времени $t$,
2. $r^i_t$ — производительность узла $i$ в момент времени $t$.
3. $z^i_t$ — новое задание, поступившее на узел $i$ в момент времени $t$,
4. $u^i_t$ — результат перераспределения задач между узлами (добавление или уменьшение).

# Топология мультиагентной системы

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

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

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

# Топология мультиагентной системы

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

Матрица степеней $D$:

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

# Балансировка загрузки сети.Постановка задачи

Пусть $\mathcal{N} = 1, \ldots, n$ — набор агентов (узлов). Связь между узлами определяется, топологией динамической сети: в момент времени $t$ графом $\mathcal{G}_t = (\mathcal{N}, \mathcal{E}_t)$ с матрицей смежности $A$. Обозначим через $W_t = \mathcal{L}_t(C) = D - C$ лапласиан графа $G_t$.

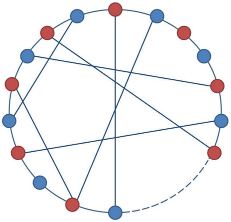

Рис.: Пример топологии сети

# Оптимальное решение[1]

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: $tm^i(q_i) = q_t^i / r_t^i$.
Функционал среднего риска (Потенциал Лапласа):

$$F_t(q_t) = \sum_{i,j \in \{1,\dots,n\}} a_t^{i,j} (tm_t^i(q_j) - tm_t^j(q_i))^2 \to \min_u .$$

[1] N. 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.

# Протокол локального голосования

Консенсусное мультиагентное управление, формируемое по так называемому «протоколу локального голосования» задаётся соотношением:

$$u_t^i = \alpha_t \sum_{j \in N_t^i} b_t^{i,j}(y_t^{i,j} - y_t^{i,i}),$$

где $\alpha_t > 0$ — размеры шагов протокола управления, $b_t^{i,j} > 0, \forall j \in N_t^i$.

Протокол локального голосования — градиентный метод для потенциала Лапласа:

$$\frac{dF_t(q_t)}{dq_t^i} = \sum_{j=1}^{m} 2(a_t^{i,j} - a_t^{j,i})\frac{1}{r_t^i}\Big(tm_t^j(q_j) - tm_t^i(q_i)\Big).$$

.

# Ускорение по Нестерову

На каждом шаге алгоритма оптимальное значение $\hat{\theta}_k$ обновляется в соответствии с направлением на предыдущем шаге $v_{k-1}$ и в направлении антиградиента.

$$v_k = \gamma v_{k-1} + \eta \nabla_\theta F(\hat{\theta}_{k-1} - \gamma v_{k-1})$$
$$\hat{\theta}_k = \quad \hat{\theta}_{k-1} - v_k$$

где $\gamma$ и $\eta$ — коэффициенты, $F$ — минимизируемая функция.

# Постановка задачи

Рассматривается модель для конкретного узла. Единственной доступной информацией служат измерения протокола локального голосования $Y_k(q)$, искаженные аддитивным шумом $\xi_k$:

$$Y_k(q) = \nabla F_k(q) + \xi_k, \quad k = 0, 1, 2, \dots . \tag{1}$$

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

$$\exists N, C < \infty : \forall k > N \quad \mathbb{E}\|\hat{q}_k - q_k\|^2 \leq C. \tag{2}$$

# Условия накладываемые на функцию

1. Функции $F_n$ имеют общую константу Липшица $L > 0$ и общую константу строгой выпуклости $\mu > 0$:

$$\forall x \in \mathbb{R}^d : \|\nabla F_n(x)\| \leq L\|x - q_n\|,$$
$$\langle \nabla F_n(x), x - q_n \rangle \geq \mu \|x - q_n\|^2.$$

2. Для $\forall n > 0 > 0$ изменение функции $F_n$ ограничено, т.е. существуют такие константы $a, b, c$, что для $\forall n > 0$ выполняется:

$$\|F_n(x) - F_{n+1}(x)\| \leq a\|\nabla F_n(x)\| + b,$$
$$\|\nabla F_{n+1}(x) - \nabla F_n(x)\| \leq c.$$

3. Шум $\xi_n$ имеет нулевое математическое ожидание и ковариация шума в среднем ограничены:

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

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

# Алгоритм

1. Выбрать $\hat{q}_0 \in \mathbb{R}^d$ и $\gamma_0 > 0,\ v_0 = \hat{q}_0$. Выбрать $h > 0, \eta \in (0, \mu), \alpha_x \in (0, 1)$ такое, что неравенство (3) всегда выполнялось. Вычисляется значение $H_1 = h - \frac{h^2 L}{2}$.
2. На $k$-ой итерации:
    1. Найти $\alpha_k \in [\alpha_x, 1)$:
    $$H_1 - \frac{\alpha_k^2}{2\gamma_{k+1}} > 0 \tag{3}$$
    2. Вычислить $\gamma_{k+1} = (1 - \alpha_k) * \gamma_k + \alpha_k * (\mu - \eta)$
    3. Вычислить
    $$z_k = \frac{\alpha_k \gamma_k v_k + \gamma_{k+1} \hat{q}_k}{\gamma_k + \alpha_k(\mu - \eta)}$$
    и посчитать значение $Y_k(z_k)$
    4. Вычислить $\hat{q}_{k+1}$:
    $$\hat{q}_k = z_k - hY_k(z_k)$$
    5. Вычислить $v_{k+1} = \frac{1}{\gamma_k}\Big[(1 - \alpha_k)\gamma_k v_k + \alpha_k(\mu - \eta)z_k - \alpha_k Y_k(z_k)\Big]$

# Сходимость алгоритма

Для доказательства сходимости метода введем следующие обозначения: Пусть $\{\alpha_k\}_{k=0}^{\infty}, \{\lambda_k\}_{k=0}^{\infty}, \{A_k\}_{k=0}^{\infty}, \{Z_k\}_{k=0}^{\infty}, \{D_k\}_{k=0}^{\infty}$ последовательности определенные следующим образом:

$$\alpha_k \in [\alpha_x, 1), \lambda_0 = 1, \lambda_{k+1} = (1 - \alpha_k)\lambda_k,$$

$$A_0 = 0,$$

$$A_{k+1} = (1-\alpha_k)((1 - \alpha_k)a + A_n),$$

$$Z_n = (1-\lambda_k)(b + ac) + A_k c,$$

$$D_0 = 0,$$

$$D_{k+1} = (1 - \alpha_k)D_k + \frac{a(1 + \alpha_k) + hc}{4\epsilon} + (1 + \alpha_k)b +$$

$$+(1 - \alpha_k)Z_k + h^2 \frac{L}{2}\sigma^2 + \frac{\alpha_k c^2}{2\eta}.$$

# Сходимость алгоритма

Тогда имеем:

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

где

$$\Gamma = \max_{n\geq 0}\gamma_k,$$

$$\epsilon \in \left(0, \frac{1}{a(1+\alpha_x)+hc}\left(H_1 - \frac{\alpha_x^2}{2\Gamma}\right)\right]$$

# Сходимость алгоритма

**Теорема**[2] Если предположения 1–3 выполнены, то алгоритм, описанный выше, решает проблему (2) со следующими параметрами:

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

Ошибка оценки после конечного числа итераций ограничена:

$$\mathbb{E}_k F_k(\hat{q}_k) - F_k(q_k) \leq \prod_{i=1}^{k} (1 - \alpha_k)(\phi_0(q_0) - F_k(q_k) + \Phi) + D_k,$$

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

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

# Результаты

- Реализовала существующий алгоритм балансировки загрузки сети на основе протокола локального голосования
- Был найден алгоритм, позволяющий ускорить протокол локального голосования
- Изучены существующие доказательства сходимости алгоритмов ускоренных по Нестерову