

Построение формальной модели распределения системных ресурсов ОС Linux

Хрипко К.Ю.

Московский физико-технический институт
Факультет управления и прикладной математики
Кафедра теоретической и прикладной информатики Parallels

Научный руководитель к.ф.-м.н. П. В. Емельянов

Москва,
2014 г.

Ресурсы

- 1 Память
 - Физическая память
 - Раздел подкачки страниц (swap)
- 2 Центральный процессор
 - Количество ядер процессора
 - Частота процессора
 - Средняя загрузка процессора
- 3 Ввод-вывод
 - Время произвольного доступа

Задача

Нужно понять, как долго будет работать программа, если добавим/уберем ресурсы.

Определить возможность оптимального ускорения работы приложения путем выделения дополнительных ресурсов для ее работы методами, не связанными с непосредственным анализом исходного и машинного кода, а привлекая элементы статистики и прогнозирования.

Три программы:

1 Занимающая память



2 Долгие вычисления



3 Программа-архиватор



Используемые технологии



VirtualBox®



OpenVZ

Математический пакет





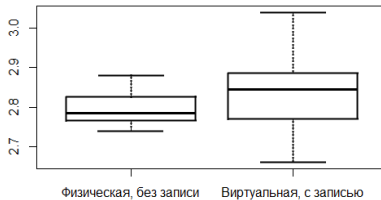
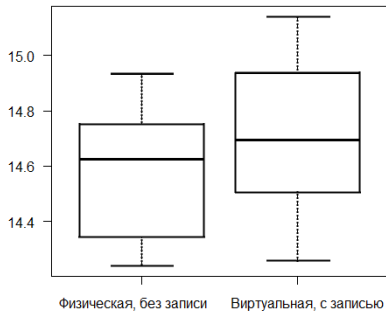
физическая
машина

?



виртуальное
окружение

Проверка эквивалентности



Проверка нормальности

Критерий Шапиро-Уилка:

$$W(X^n) = \frac{\left(\sum_{i=1}^n a_i X_{(i)} \right)^2}{\sum_{i=1}^n (X_i - \bar{X})^2},$$

Проверка равенства средних

Критерий Аспина-Уэлча:

$$T(X_1^{n_1}, X_2^{n_2}) = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}},$$

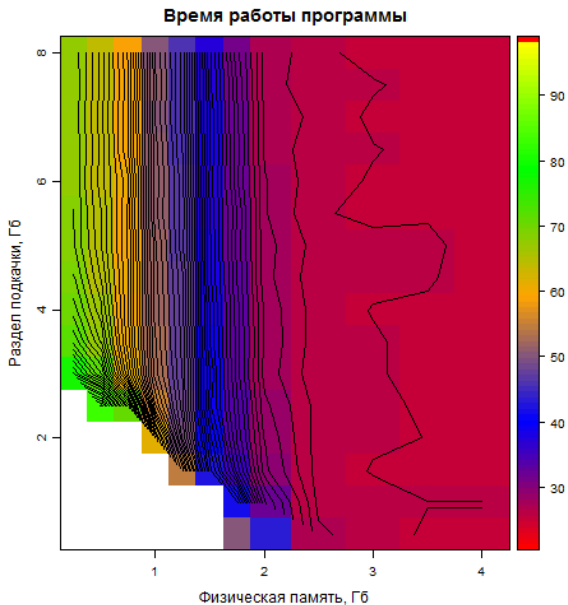
Устанавливаемые параметры:

- Размер оперативной памяти (*ram*)
- Размер раздела подкачки (*swap*)
- Число процессоров (*cpu*)
- Средняя загрузка ЦП (*load*)
- Частота процессора (*freq*)

Измерявшиеся параметры:

- Время работы программы

Пример данных



Параметры преобразованы в нелинейные функции g_1, \dots, g_n .
Например:

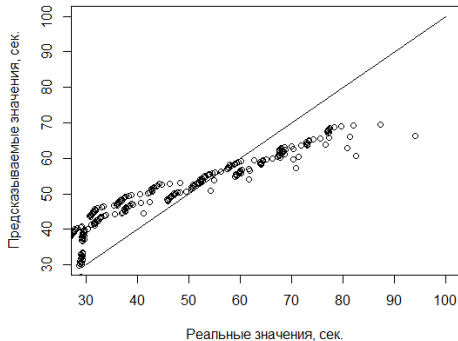
$$\begin{aligned}g_1 &\equiv x_1 = freq; & g_4 &= load \cdot ram; \\g_2 &\equiv x_2 = swap; & g_5 &= \frac{load}{cpu}; \\g_3 &= e^{ram}; & g_6 &= \frac{ram}{ram+swap}.\end{aligned}$$

Модель зависимости:

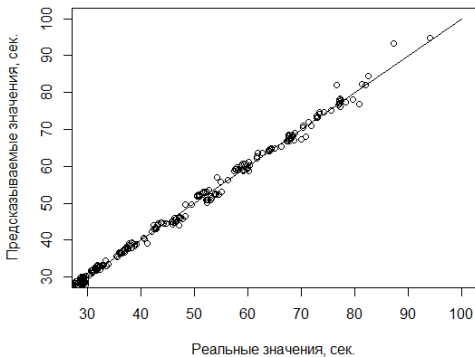
$$t = w_0 + \sum_{i=1}^m w_{1i} g_i + \sum_{i=1}^m w_{2i} g_i^2 + \nu_i$$

Пример простой модели

$$t = 84.901 - 16.157ram + 0.459swap - 9.023cpu + 1.752ram \cdot cpu.$$



$$t = 104.5 - 33.83ram - 6.801swap + 7.415ram^2 + 0.63swap^2 - 0.322e^{ram} + 67.93 \frac{1}{ram + swap} - 52.69 \frac{ram}{ram + swap} - 0.474cpu$$



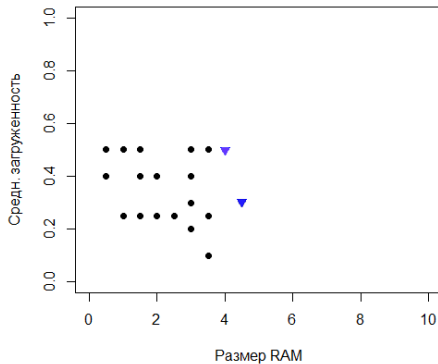
	Уравнение регрессии	R_{adj}^2
Память	$t = 104.5 - 33.83ram - 6.801swap + 7.415ram^2 + 0.63swap^2 - 0.322e^{ram} + 67.93 \frac{1}{ram+swap} - 52.69 \frac{ram}{ram+swap} - 0.474cpu$	0.9634
ЦП	$t = 21.69 - 4.22ram - 16.1freq + 10.44load - 3.08cpu - 1.415load^2 - 1.373 \frac{cpu}{load} + 5.386cpu \cdot load$	0.9319
tar	$t = 37.63 - 3.76swap - 0.33e^{ram} + 36.13 \cdot \frac{ram}{ram+swap} - 6.931load \cdot ram$	0.9421

Коэффициент детерминации

$$R_{adj}^2 = \left(1 - \frac{\sum_{i=1}^n (t_i - \hat{t}_i)^2}{\sum_{i=1}^n (t_i - \bar{t})^2} \right) \frac{(n - 1)}{(n - k)}$$

Проблемы и их решения

- Переобучение — ограничение степени полинома
- Мультиколлинеарность — метод шаговой регрессии
- Достоверность — эксперимент по экстраполяции



$$\hat{t}_1 = 8.775 \in [8.092, 8.805]$$

$$\hat{t}_2 = 9.31 \in [8.994, 9.647]$$

Выводы

- 1 Доказана нелинейность, построена модель
- 2 Показана неуниверсальность — модели различные для каждой из типов программ
- 3 Проверены расчетные значения в точках вне выборки

Дальнейшее развитие

- 1 Нахождение области допустимой экстраполяции
- 2 Автоматизация процесса:
 - построить нелинейных функций
 - настроить модель
 - избежать переобучения