Rruntime-мониторинг.

Как наблюдать за фрагментами кода **во время реального использования** C++ приложения?

Большинство инструментов для профилирования кода написаны системщиками и для системщиков. Эти утилиты обладают широкими возможностями, но требуют изучения и настройки перед использованием, анализ результатов часто тоже не тривиален. Иногда такие утилиты не самодостаточны и для своего использования требуют доп. программы (трансляция в asm, парсинг логов и т.п.).

Существуют множество замечательных инструментов подобных Google Benchmark для препарирования и изучения фрагментов кода в лабораторных условиях. Есть статические анализаторы вроде LLVM-MCA, которые позволяют получить точнейшие метрики опять же фрагмента кода вне реального окружения.

В сложных случаях

Для контроля производительности кода существуют различные способы:

Существует множество замечательных библиотек: Google и BOOST benchmarks, BenchmarkPP, .

Утилиты для профилирования кода: Valgrind, Visual Studio Profiler, Intel VTune Profiler и т.д.   
Статические анализаторы производительности: OSACA, LLVM-MCA и др.

Измерять производительность написанного кода полезно не только системным и HPC программистам. “Прикладники” тоже сталкиваются с ограничениями по времени выполнения.

Примеры из моей практики: реализация сетевого протокол поверх UDP с нормированным временем обработки пакета; реализация сотен тестов для проверки десятков тысяч объектов за время, которое способен вытерпеть User) и т.п.

Сколько времени потребуется компьютеру для выполнения конкретного фрагмента кода? Отвечать на подобные вопросы приходится всем программистам.