

## Assessing the Impact of Operating System Architecture on Supercomputer Performance.

### Оценка влияния архитектуры операционной системы на производительность суперкомпьютера.

Семанин А.В.

Севастопольский государственный университет (СевГУ)

Высокопроизводительные вычисления (HPC) являются локомотивом развития науки и техники. Мы видим новые рекорды производительности и объемам обрабатываемой информации. Крупнейшие корпорации вкладывают гигантские деньги в развитие отрасли. Но движется ли индустрия HPC в правильном направлении? Оборудование в современных ЦОДах все меньше отличается от устанавливаемого в суперкомпьютеры. Уникальные архитектуры в HPC встречаются все реже и реже, и чаще всего представляют собой кластеры из серверов для предприятий. Наверное, для этого есть объективные причины, связанные с производством.

Но нужно ли нам поступать также и с программным обеспечением в HPC? Зачем устанавливать операционную систему (ОС), предназначенную для управления предприятием или корпорацией в суперкомпьютер? Такие системы, как `unix (linux)`, разрабатываются для противостояния всем угрозам, обрушивающимся на сервера предприятий как изнутри, так и снаружи, от всех хакеров Интернета. Большая часть кода ОС избыточна для HPC. Разнообразие операционных систем, библиотек и их версий привели к несовместимости программного кода и необходимости его контейнеризации. Но зачем нам эти новые сущности как `Singularity` и `Apptainer`? Это вынужденная мера, потому что мы продолжаем использовать `unix` и наследуем все её достоинства и недостатки. Мы забыли, для чего вообще создавались системы HPC – для решения важных, но очень сложных задач, а не решением множества мелких простых задач. Когда суперкомпьютеры были редкостью и их ресурсов на всех не хватало, а желающих их потреблять было много, то весь этот “колхоз”, которым управляли системы типа `slurm` был оправдан. Но сейчас, нередко, выделяют весь суперкомпьютер для решения одной задачи для одной научной группы. Накладные расходы и сложность организации вычислений растут, а удобство использования суперкомпьютера падает. Это отталкивает новых пользователей и увеличивает время их первоначального обучения и адаптации их программ.

Но есть и альтернативный путь, на который указали отцы основатели `unix` – и это `Plan9`. Они говорили – **“The early catch phrase was to build a UNIX out of a lot of little systems, not a system out of a lot of little UNIXes.”** **Rob Pike**. Этот подход дает массу преимуществ и делает `Plan9` идеальной системой для HPC. Странно, что ее редко используют. Возможно, это связано с ее неудачным лицензированием, которого не было у `unix (linux)`. Но сейчас это проблема снята. В марте 2021 года права на операционную систему переданы фонду `Plan 9` с лицензией MIT.

Сама система имеет приятный графический интерфейс, с которым удобно работать прямо с домашнего компьютера. Система имеет возможности гибкого расширения и масштабирования по ресурсам. В системе используется простой, но крайне эффективный протокол `9P` соединяющий все ресурсы в единое целое. Совместимость библиотек на должном уровне.

Система построена на трёх основных принципах:

- все ресурсы представлены как файлы и доступны в иерархической файловой системе.
- локальные и удалённые ресурсы не различаются, для доступа к ним реализован стандартный протокол `9P`.
- каждая группа процессов имеет собственное пространство имён, собранное из файловых иерархий, предоставленных различными ресурсами. В отличие от современных операционных систем, в которых пользователь получает доступ к персональному компьютеру или рабочей станции, в `Plan 9` пользователь получает доступ к распределённой вычислительной среде и имеет возможность конфигурировать своё рабочее пространство. Так, `/dev/mouse` для процесса — это мышь на компьютере, с которого этот процесс запущен, причём это может быть не тот компьютер, на котором выполняется процесс.

Несмотря на простоту, `Plan9` поддерживает современные протоколы шифрования, что дает возможность использовать распределенные вычисления в разных частях света.

Основным языком программирования является диалект языка `ANSI C`, отличающийся встроенной поддержкой `Unicode` и рядом других полезных расширений, таких как формирование структур и инициализация массивов. Реализованы кроссплатформенная компиляция и отладка, успешно портированы `Perl`, `Python`, `Scheme`, `noweb`, `Haskell`, `Newsqueak`, `Go` и `ML`.

Мы помним те времена, когда мини ЭВМ «убили» мэйнфреймы. Та же судьба возможно ожидает `Plan9`.