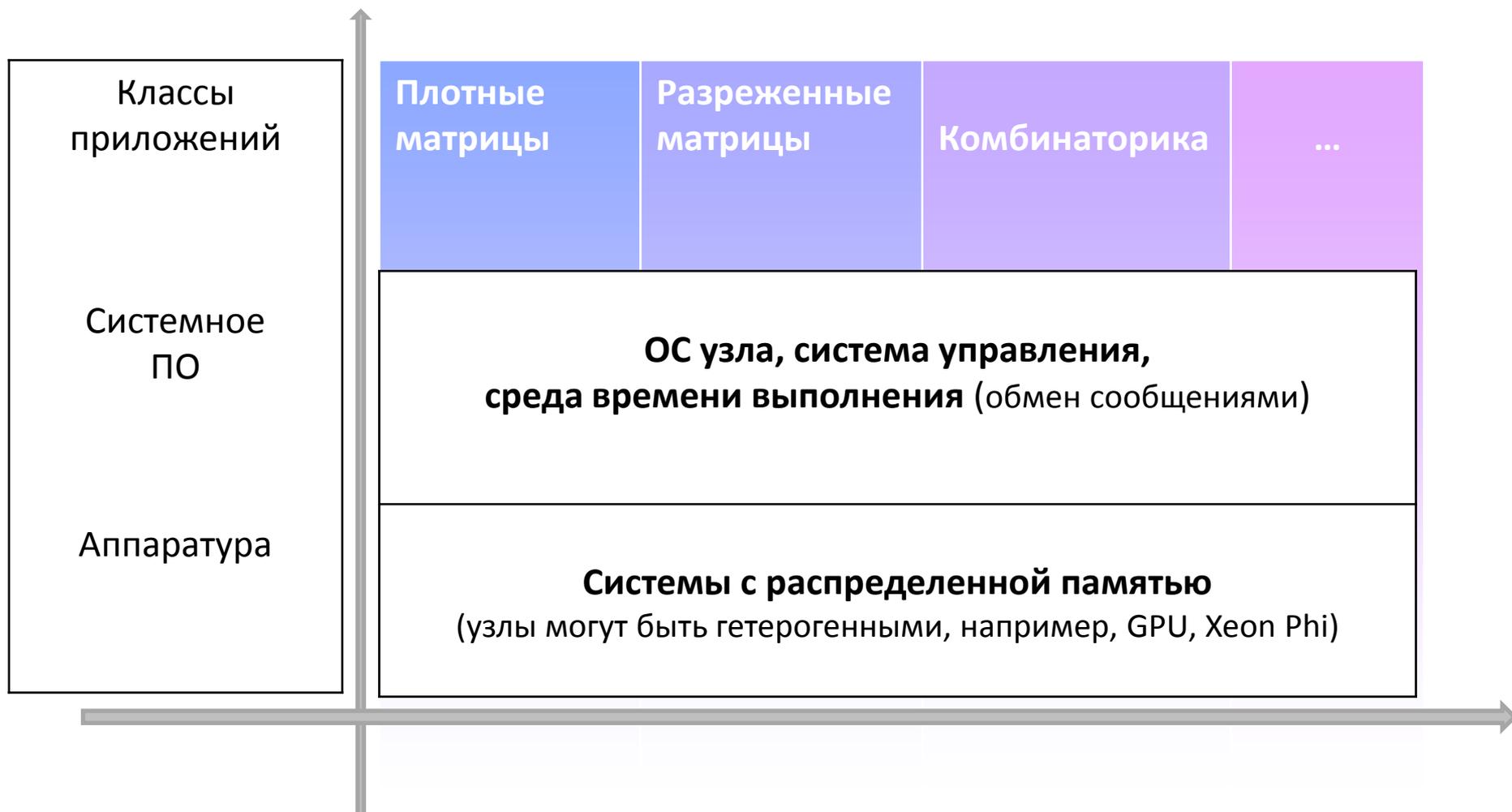


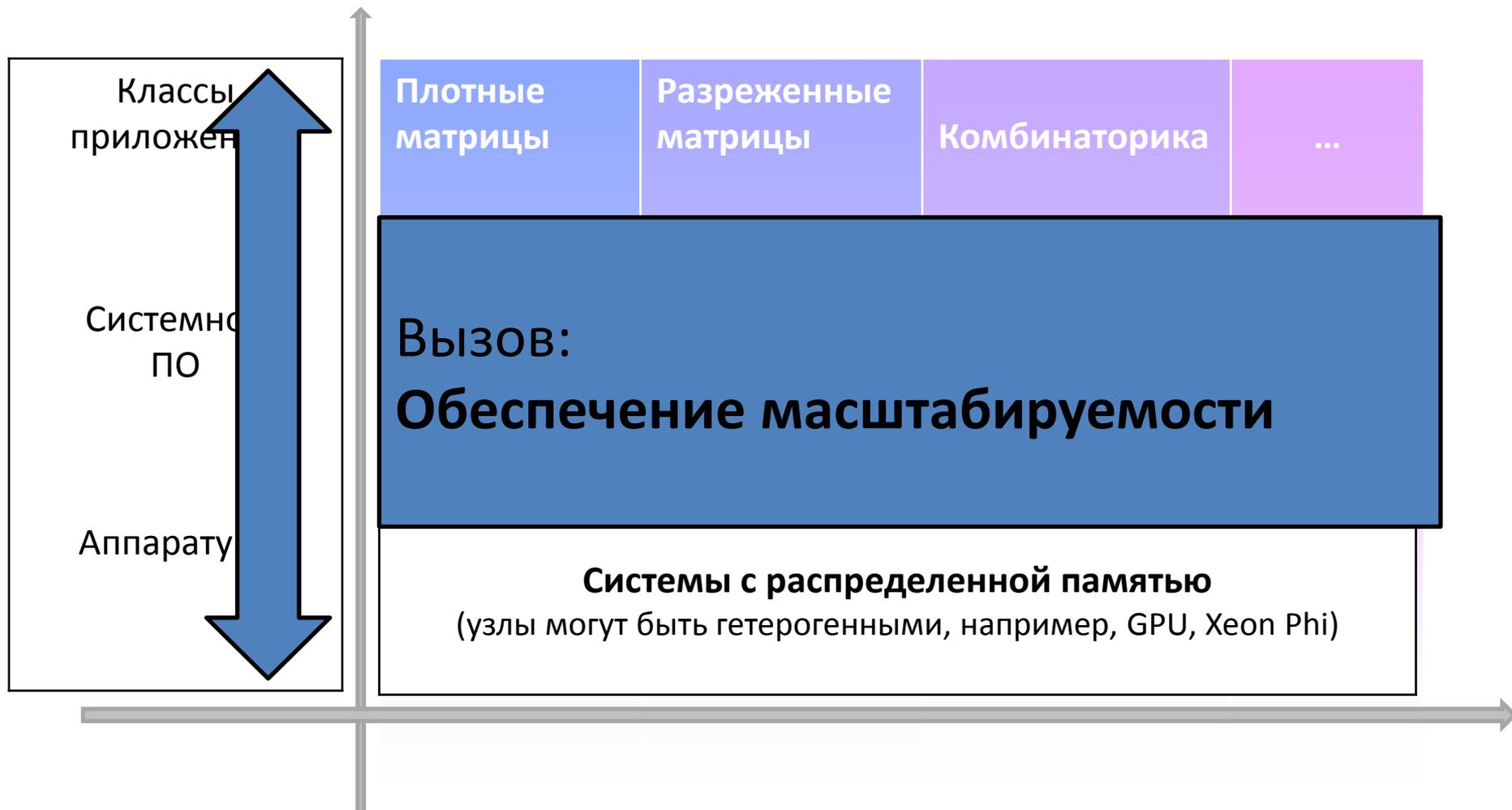
Системное ПО суперкомпьютеров – проблемы и перспективы

Четвертый Московский Суперкомпьютерный Форум,
23 октября 2013

Аветисян Арутюн Ишханович
arut@ispras.ru







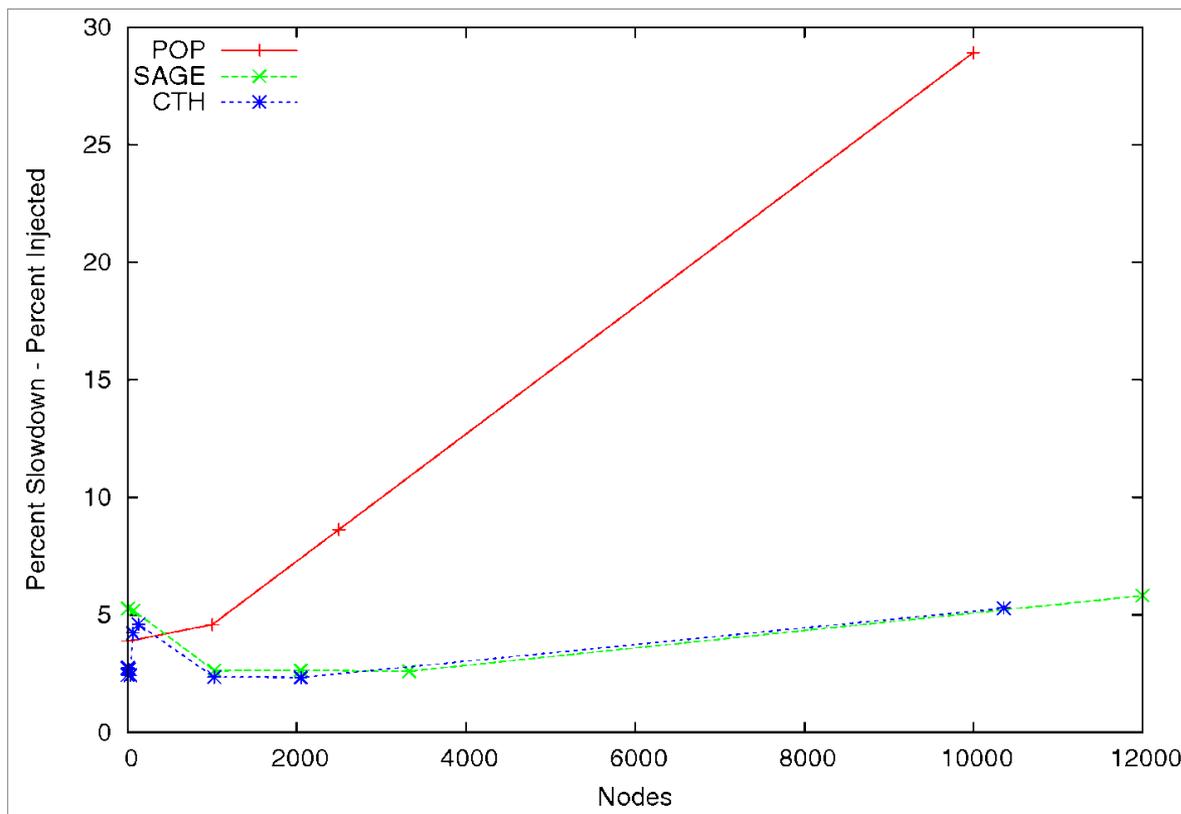
Аппаратура - масштабируемость

Top500	Количество узлов	R_{\max}/R_{peak} (Linpack)	R_{\max} (TFlop/s)
1. Tianhe-2 (MilkyWay-2) - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P	16,000 2 ×12-и ядерных Xeon + 3 ×57-и ядерных Xeon Phi	~62%	33,862.7
2. Titan - Cray XK, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x	18,688 1 ×16-и ядерный Opteron + 1 K20x	~65%	17,590.0
3. Sequoia – BlueGene/Q, Power BQC 16C 1.60 GHz, Custom	98,304 1 ×16-и ядерный Power	~85%	17,173.2
4. K computer , SPARC64 VIIIfx 2.0GHz, Tofu interconnect	88,128 1 ×8-и ядерный SPARC	~93%	10,510.0

Масштабируемость кластерных систем ограничена (~10000 узлов) на уровне аппаратуры (commodity компоненты)

ОС узла - масштабируемость

Шум ОС (высокочастотный, низкочастотный) существенно ограничивает масштабируемость



Необходимы специальные ОС с минимальным уровнем шума
(например, CNL, Catamount, CNK – закрытые ОС)

Среды программирования - масштабируемость

Проблема: обеспечить эффективность использования возможностей программно-аппаратных систем при минимизации стоимости разработки и сопровождения (**продуктивность**)

Мечта! Создание языка высокого уровня, позволяющего эффективно использовать существующие возможности аппаратуры, как в случае последовательных программ

Попытки создания параллельных языков высокого уровня не прошли: *HPF, Cilk (MIT), Unified Parallel C (Berkeley), Fortress (Sun), Chapel (Cray), X10 (IBM)* и др.

Одна из причин: отсутствие компиляторных технологий, позволяющих генерировать эффективный параллельный код (**отладка** в любом случае **на уровне MPI**)

Параллельное программирование является искусством!

Пути повышения продуктивности

- ❑ Создание языков высокого уровня, поддержанных компиляторными технологиями
- ❑ Создание методов анализа программ и соответствующих инструментов, поддерживающих разработку приложений на, так называемом, ассемблерном уровне (*MPI, Cuda, OpenCl*)
- ❑ Создание технологий для классов приложений (например, MapReduce)
- ❑ Разработка библиотек и пакетов прикладных программ
- ❑ Использование лучших практик, шаблоны программирования
- ❑ Создание междисциплинарных команд