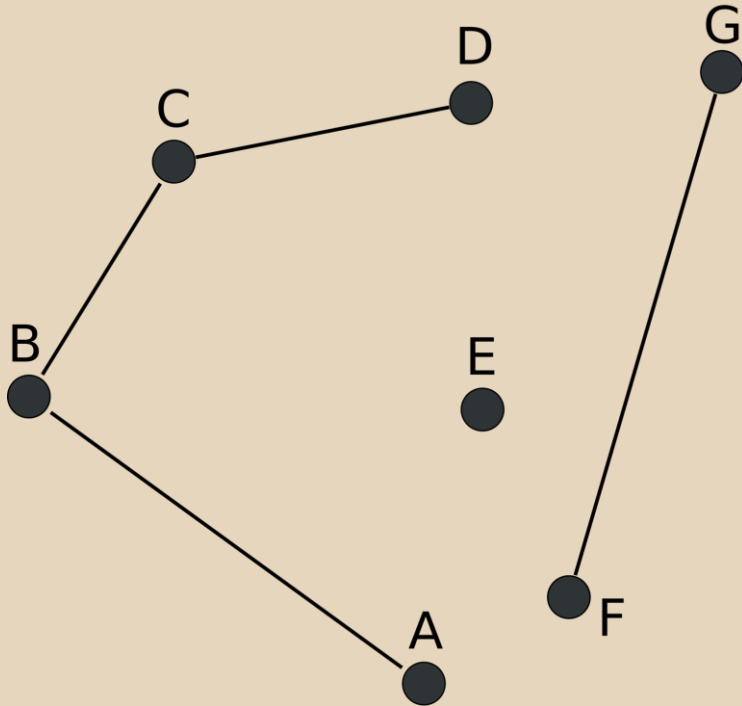


Поиск компонент связности графа на мультипроцессоре

Комаров С.О., НГУ
Калгин К.В., к.ф.-м.н., ИВМиМГ СО РАН

Компонента связности



- Вершина w графа $G(V, E)$ достижима из вершины v , если либо $v = w$, либо существует путь из v в w .
- Граф называется связным, если любая его вершина достижима из любой дугой.
- Компонентой связности графа называется его связный подграф, не являющийся собственным подграфом никакого другого связного подграфа.

Графы

Тестовые графы:

- rmat-n (2^n вершин)
- ssc2-n (2^n вершин)
- grids + single vertexs
- linear

Графы задаются в сжатом **CSR** формате.

Система

Тесты проводились на:

- *Intel Xeon E5-2690 2.90GHz 2x8 ядер*
- *Intel Xeon Phi 7110X*

Ключи компилятора icrcs:

- *O3 -ipo -march=native*

Привязка ядер:

- *Export KMP_AFFINITY=compact*

Алгоритм Shiloach-Vishkin(SV)

Массив $D[1..n]$, где $D[i]$ - номер компоненты, к которой принадлежит "i".
Изначально $D[i] = i$.

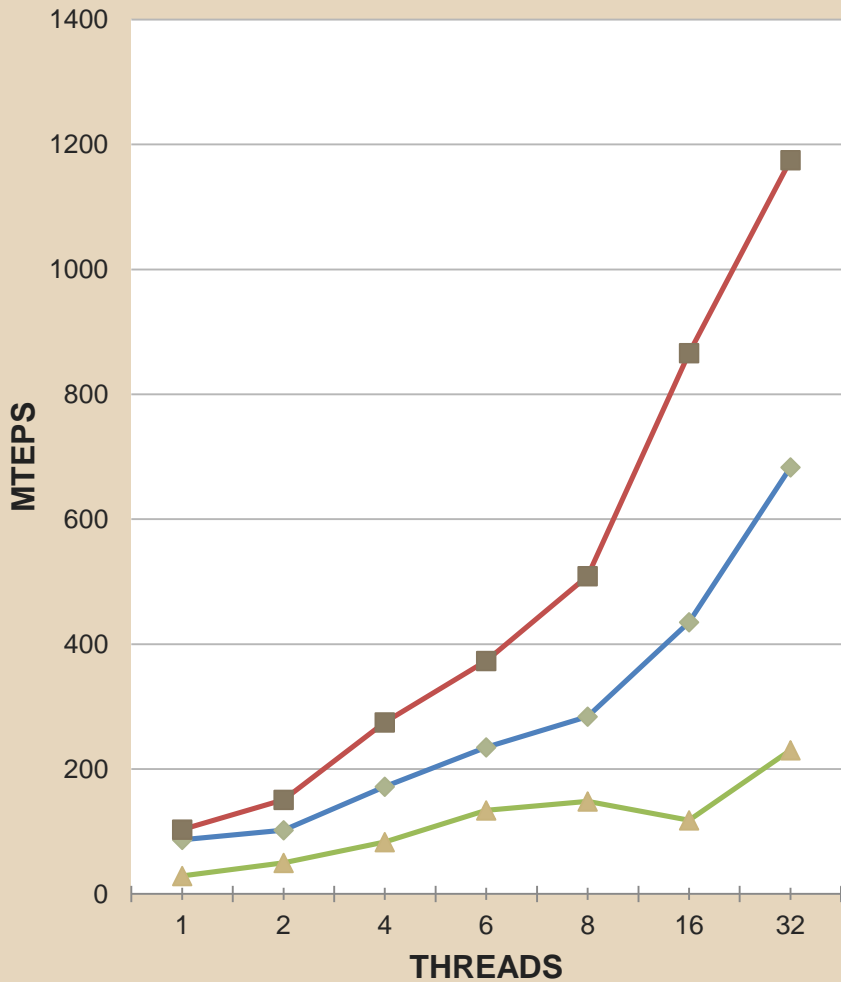
- 1) for (i, j) ∈ E do
 if((D[i] < D[j]) && (D[j] == D[D[j]])) then D[D[j]] = D[i];
- 2) Pointer Jumper
 for (i, j) ∈ E do
 while(D[i] != D[D[i]]) do D[i] = D[D[i]];
- 3) if(D не обновлялось) then exit;
 Иначе пункт 1.

Алгоритм Shiloach-Vishkin(SV)

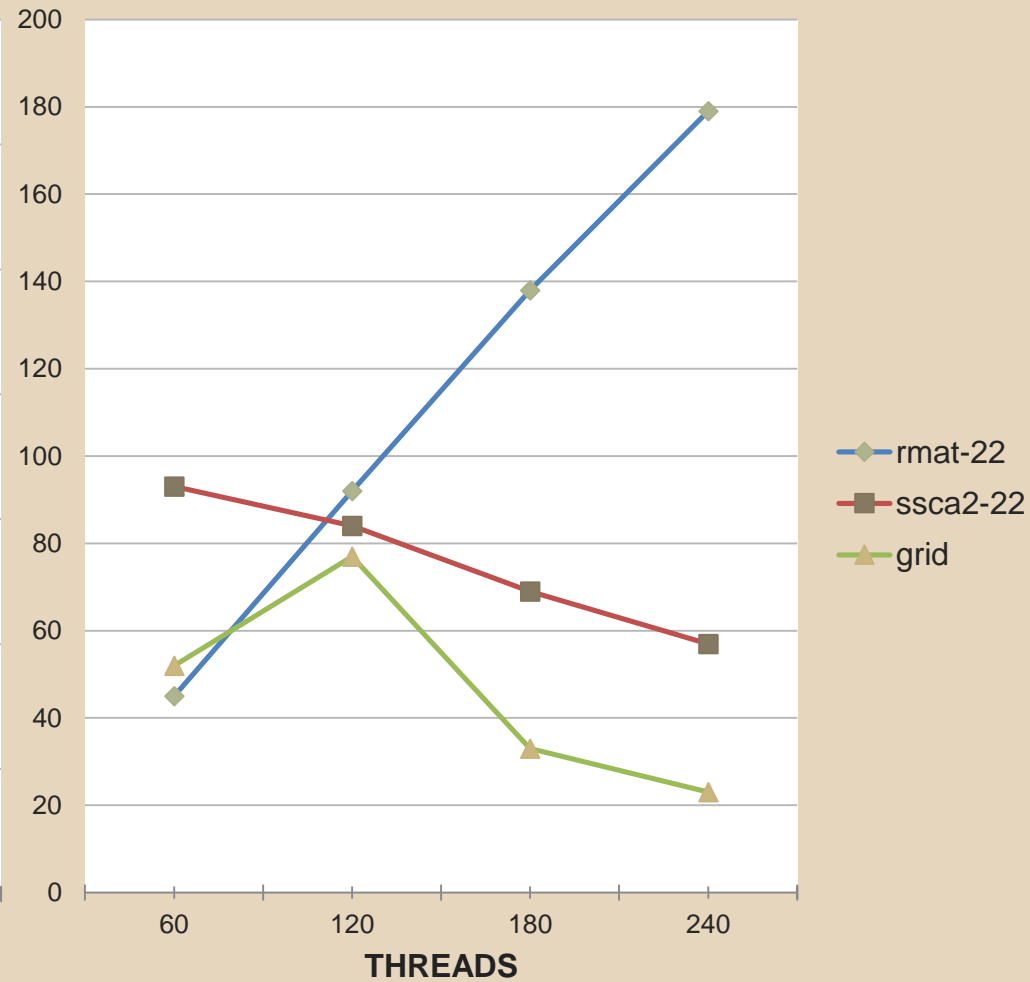
- + хорошо масштабируется
- + просто распараллеливается
 - а. вставки openMP
 - б. отсутствие зависимостей
- часто повторяет действия (переделяет работу)

Алгоритм Shiloach-Vishkin(SV)

Хеон E5-2690



Phi 7110X



Асинхронный алгоритм (ABFS)

Динамический пул цветов $P[1..n]$.

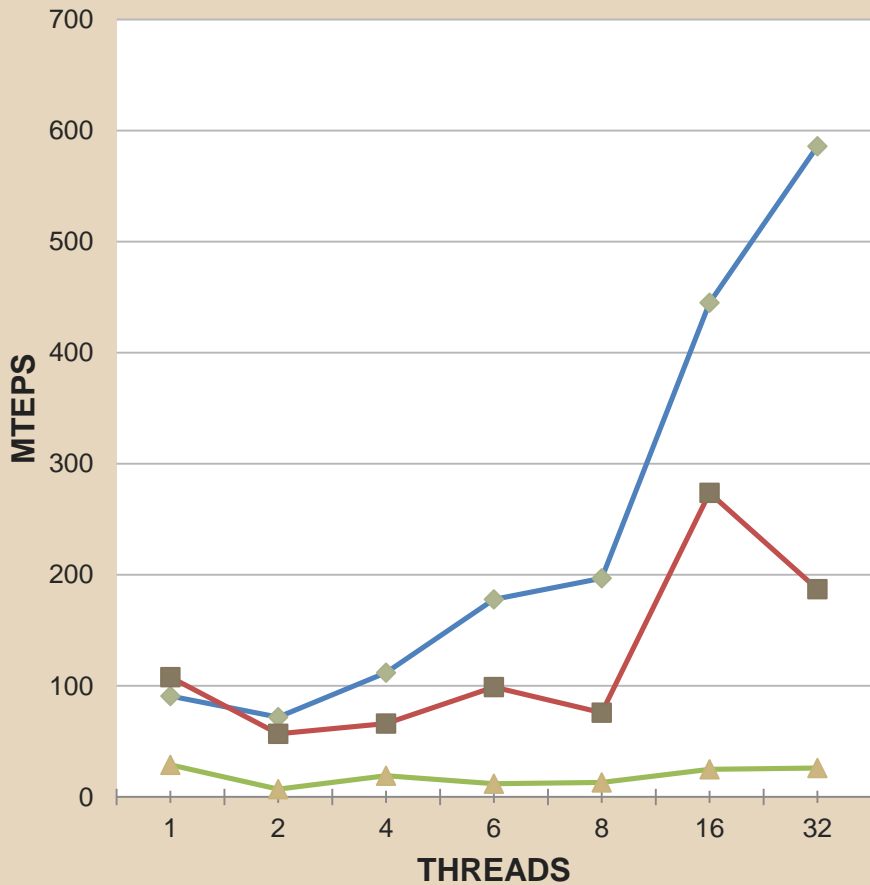
- 1) Каждый поток выполняет BFS
каждый красит подграф в определенный цвет
- 2) Pointer Jumper
- 3) Раскраска (простой обход)

ABFS(naive)+SV

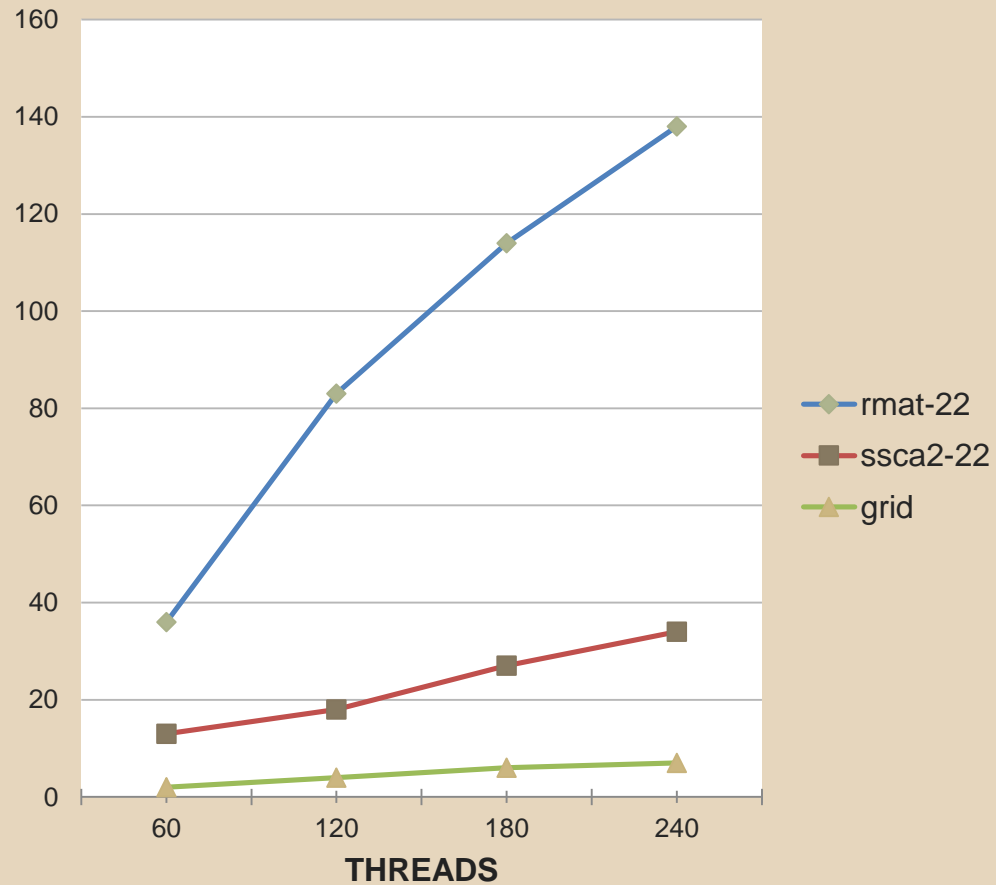
- ABFS раскрасит большие области
 - Если поток, при раскраске, наталкивается на покрашенную вершину, то он ее пропускает
- SV производит коррекцию (работа на границах)

ABFS(naive)+SV

Xeon E5-2690



Phi 7110X

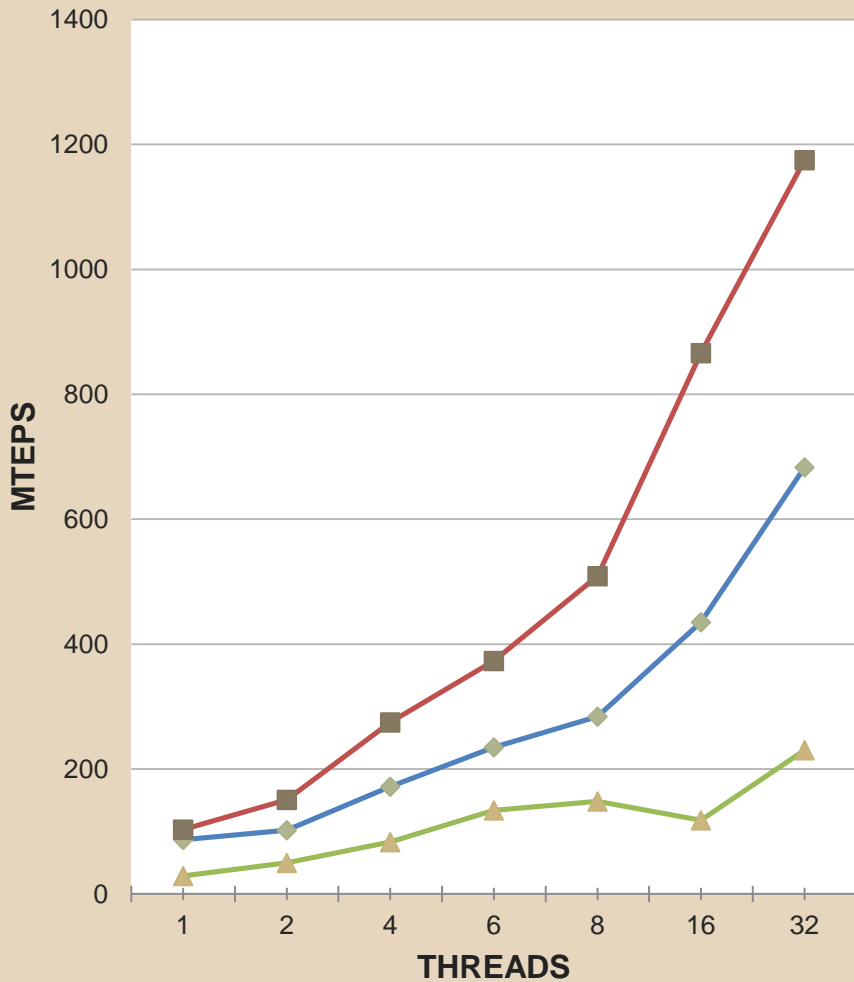


ABFS+SV

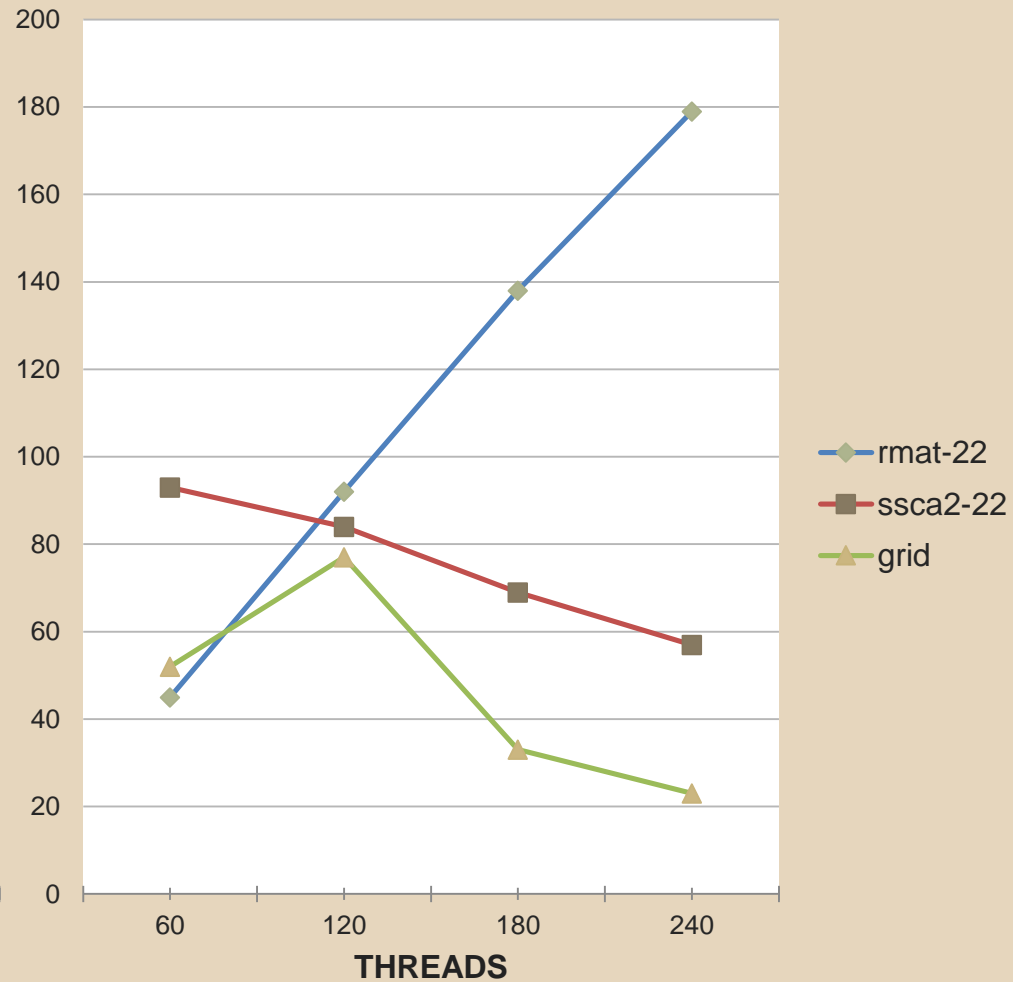
- ABFS раскрасит большие области
 - Если поток встречает покрашенную вершину, и цвет вершины меньше текущего, то поток меняет цвет
- SV производит коррекцию (на границах)

ABFS+SV

Xeon E5-2690



Phi 7110X



PBFS+critical

- BFS выполняется каждым потоком отдельно
 - Если поток, при раскраске, наталкивается на покрашенную вершину, то поток меняет цвет на меньший
 - Моменты столкновения компонент обрабатываются в критической секции
- Pointer Jumper
- Раскраска

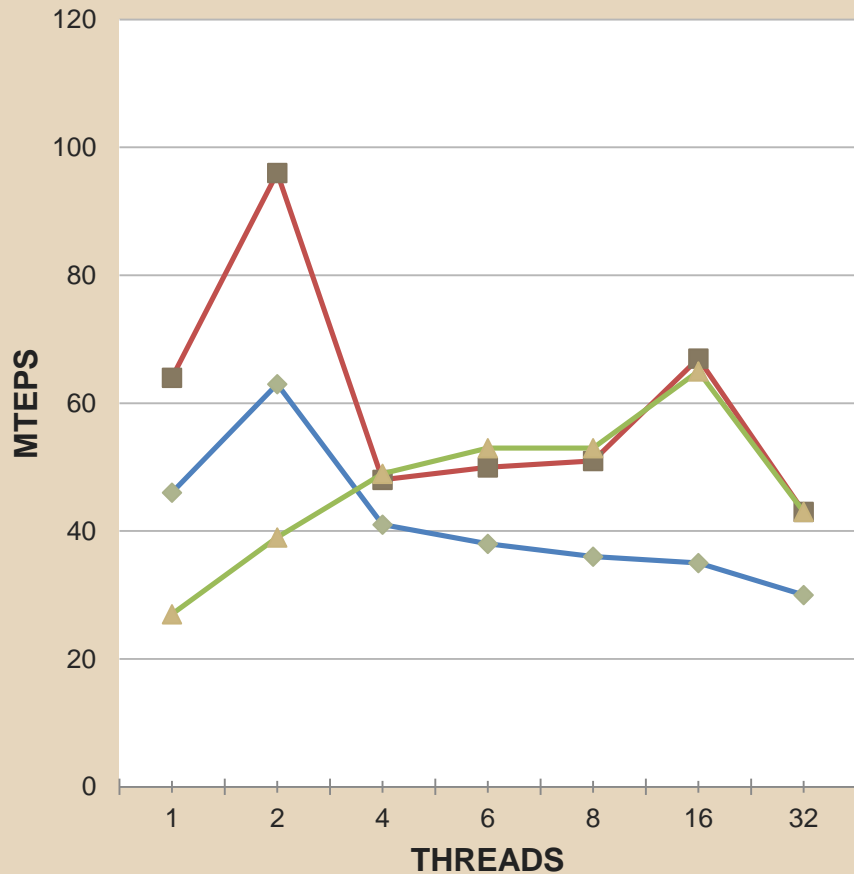
PBFS+lock free

- PBFS+critical
- Критические секции заменены на встроенные атомарные функции:
bool __sync_bool_compare_and_swap(...)

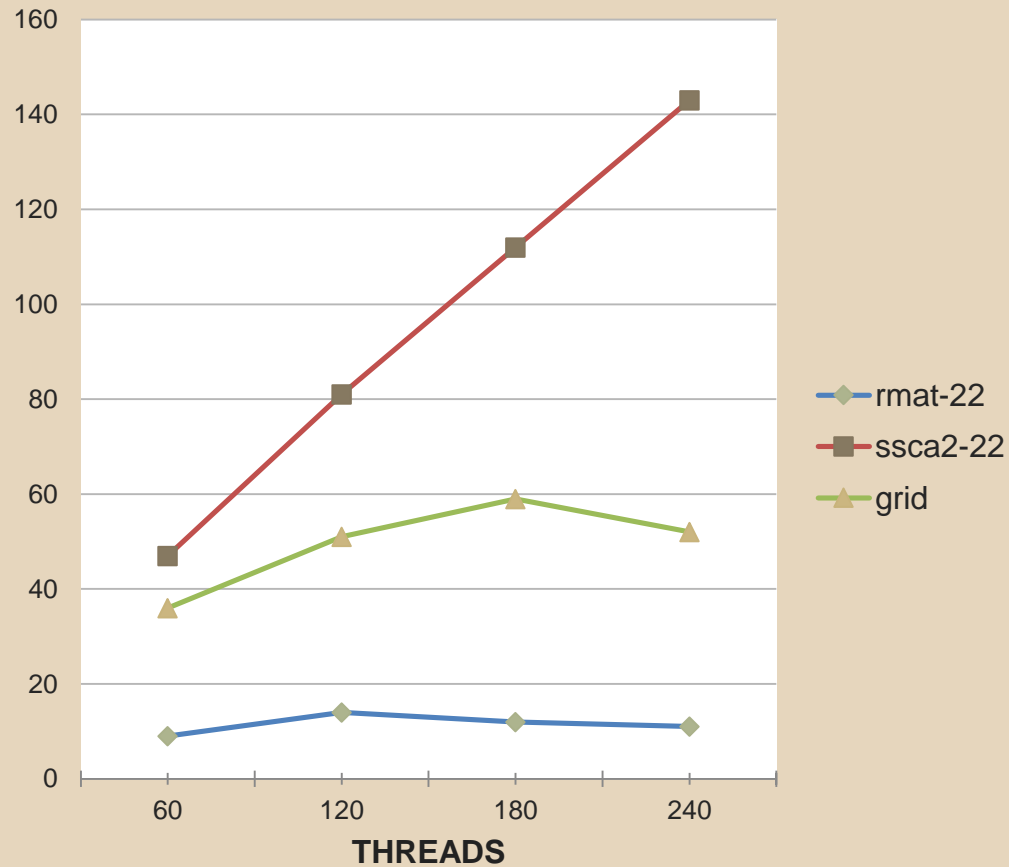
```
do {  
    while( old_root != pool[ old_root ] ) old_root = pool[ old_root ];  
    while( new_root != pool[ new_root ] ) new_root = pool[ new_root ];  
    if( new_root < old_root )  
        is_ok = __sync_bool_compare_and_swap( & pool[ old_root ], old_root, new_root );  
    else  
        is_ok = __sync_bool_compare_and_swap( & pool[ new_root ], new_root, old_root );  
} while( ! is_ok )
```

PBFS+lock free

Xeon E5-2690



Phi 7110X



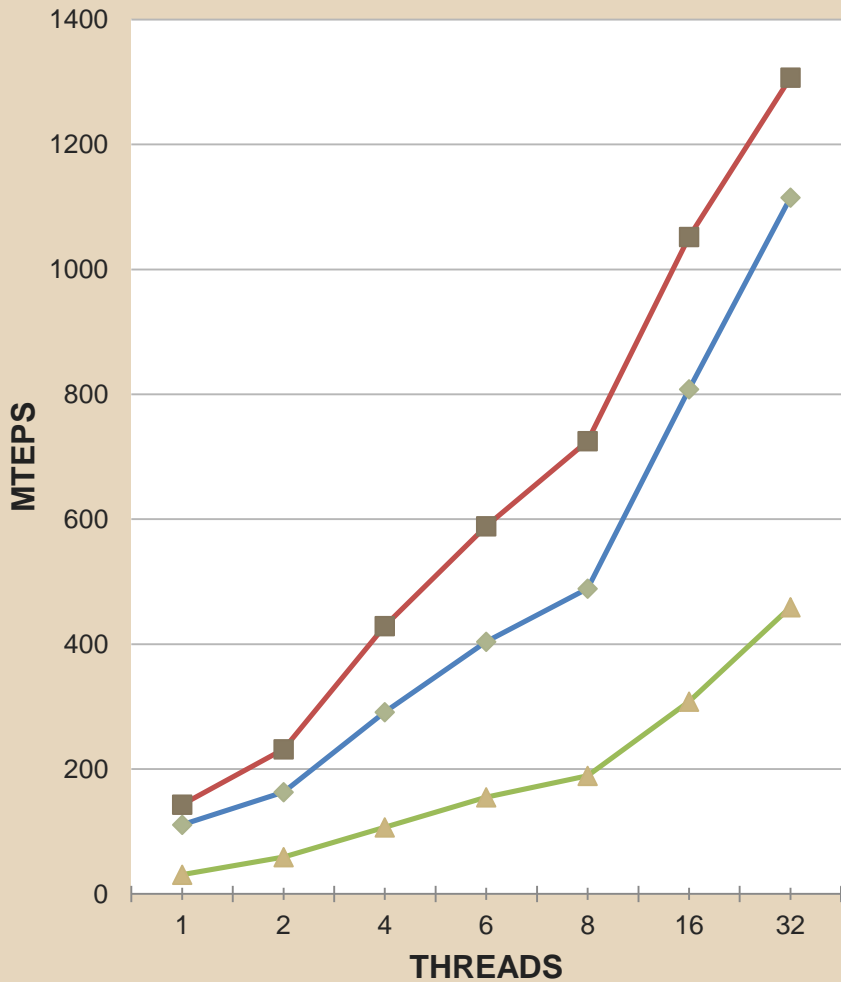
*ввиду малой производительности, этот алгоритм не будет далее рассматриваться

ABFS+SV - Hybrid

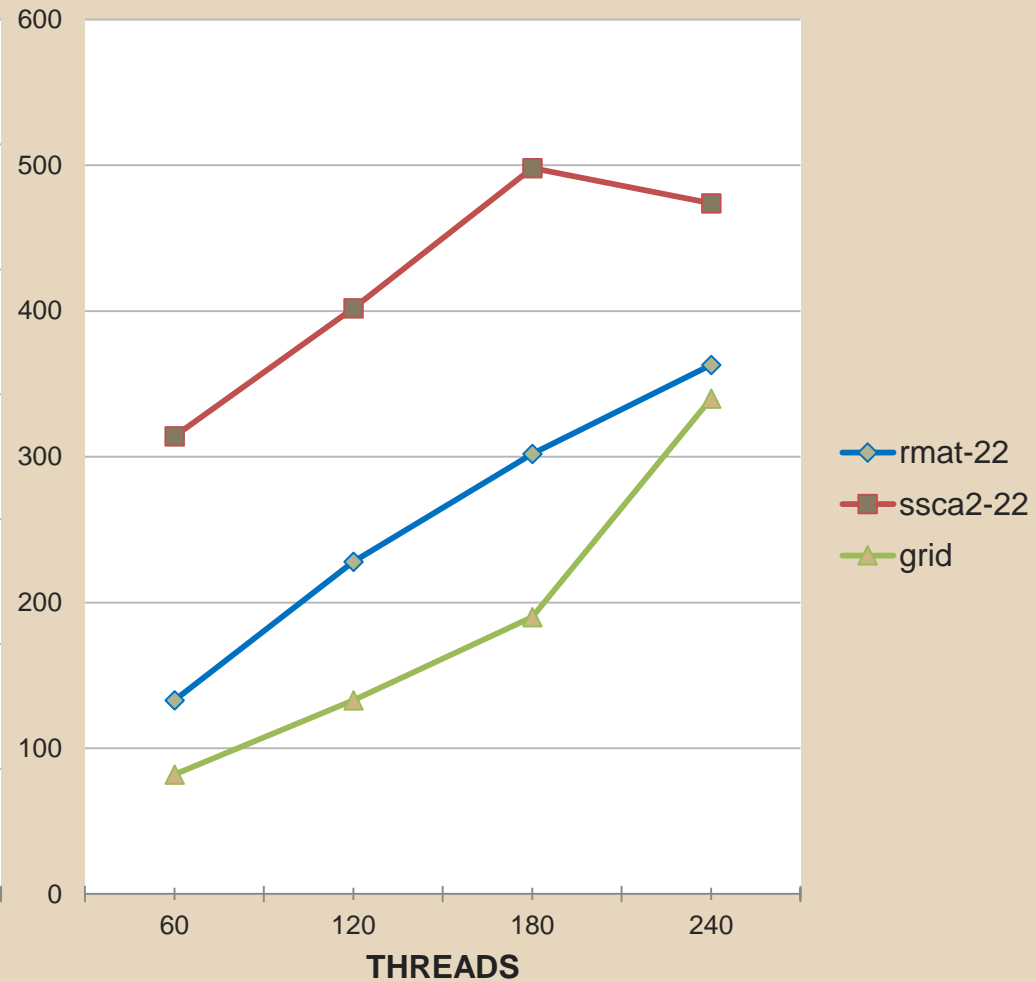
- ABFS красит большие области
 - Если поток встречает покрашенную вершину, и цвет вершины меньше текущего, то поток меняет цвет
- Первый этап алгоритма SV производит коррекцию пробегая по всем дугам (основная работа на границах)
- Pointer Jumper
- Собственно раскраска (простой цикл)

ABFS+SV - Hybrid

Xeon E5-2690

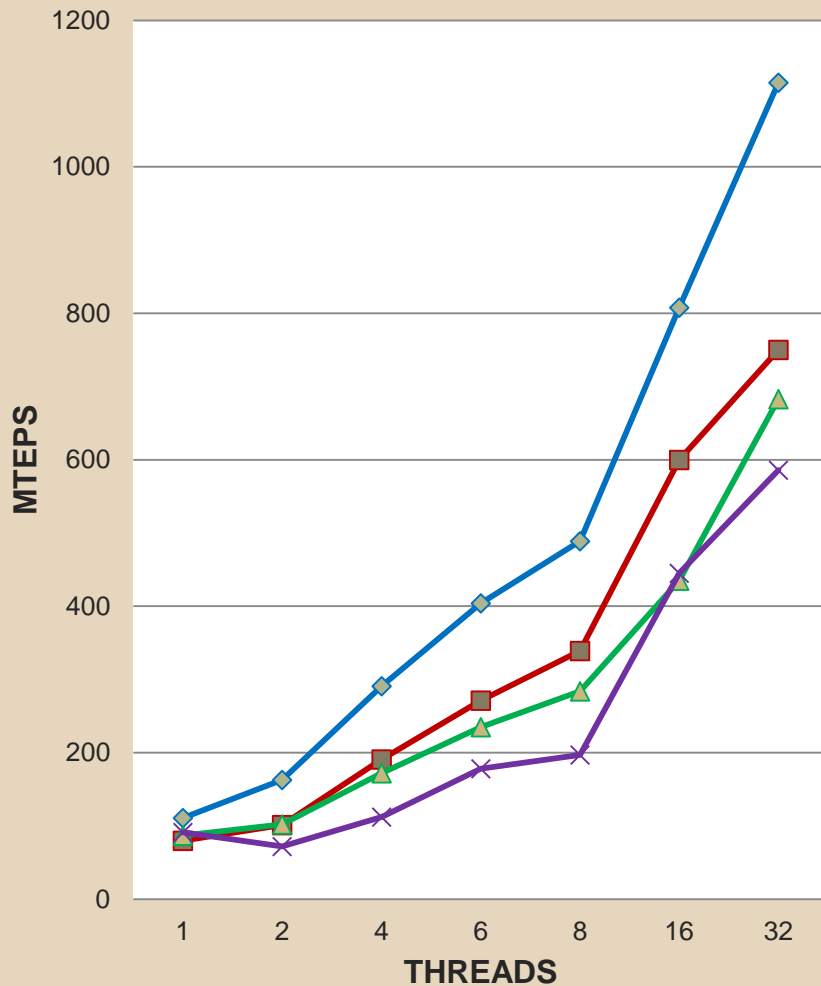


Phi 7110X

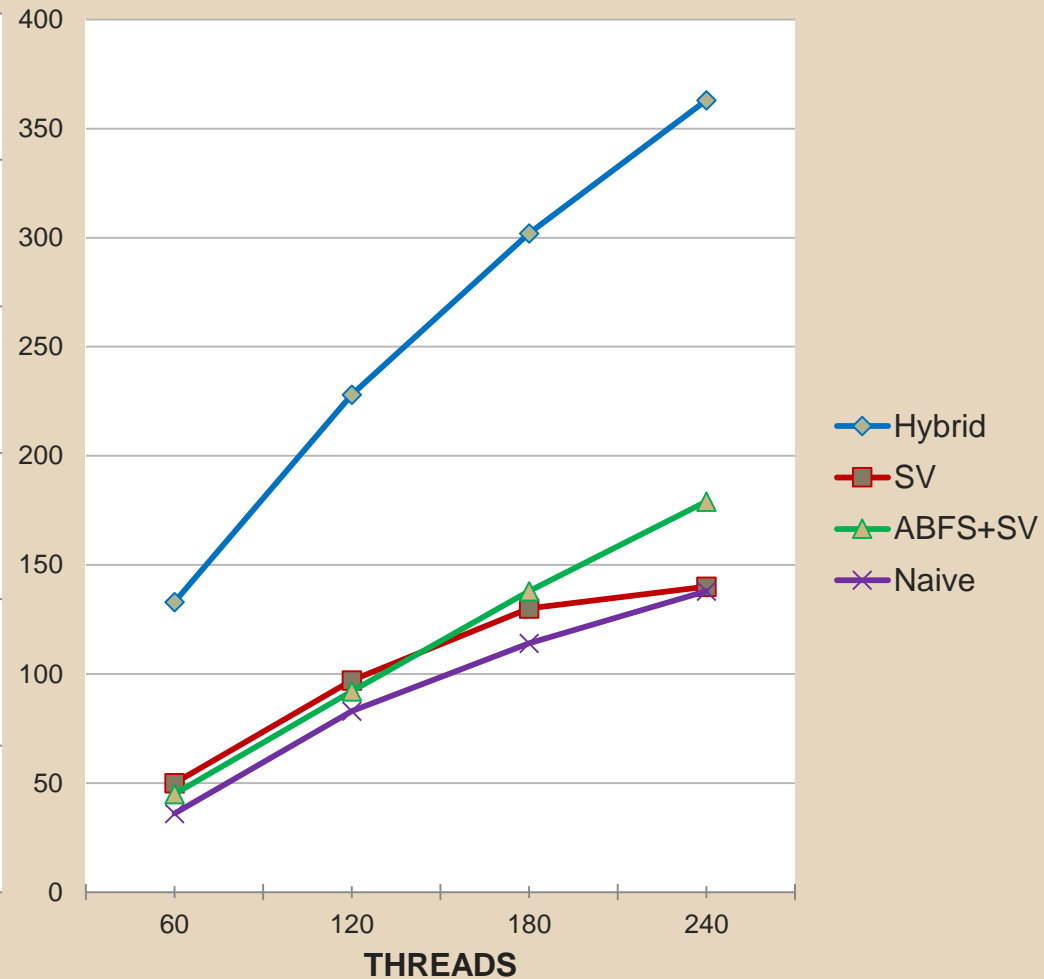


Сравнение алгоритмов на rmat-22

Хеон E5-2690

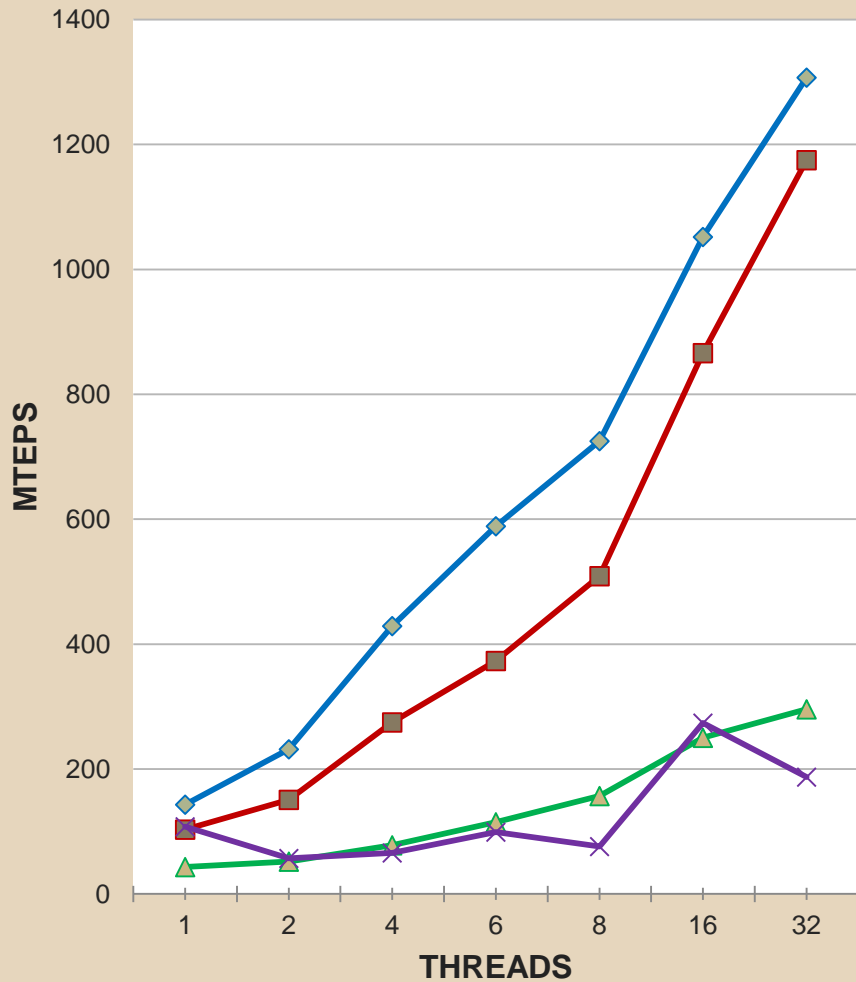


Phi 7110X

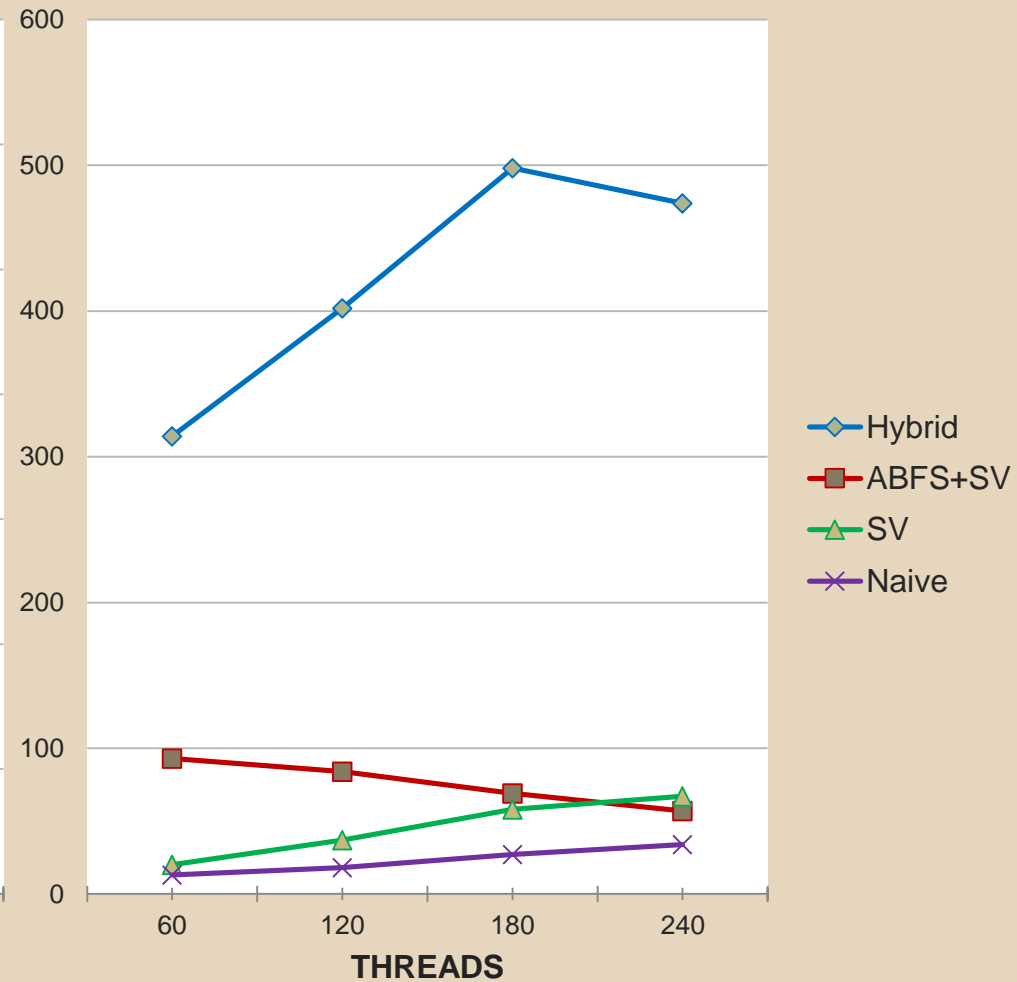


Сравнение алгоритмов на ssc2-22

Хеон E5-2690

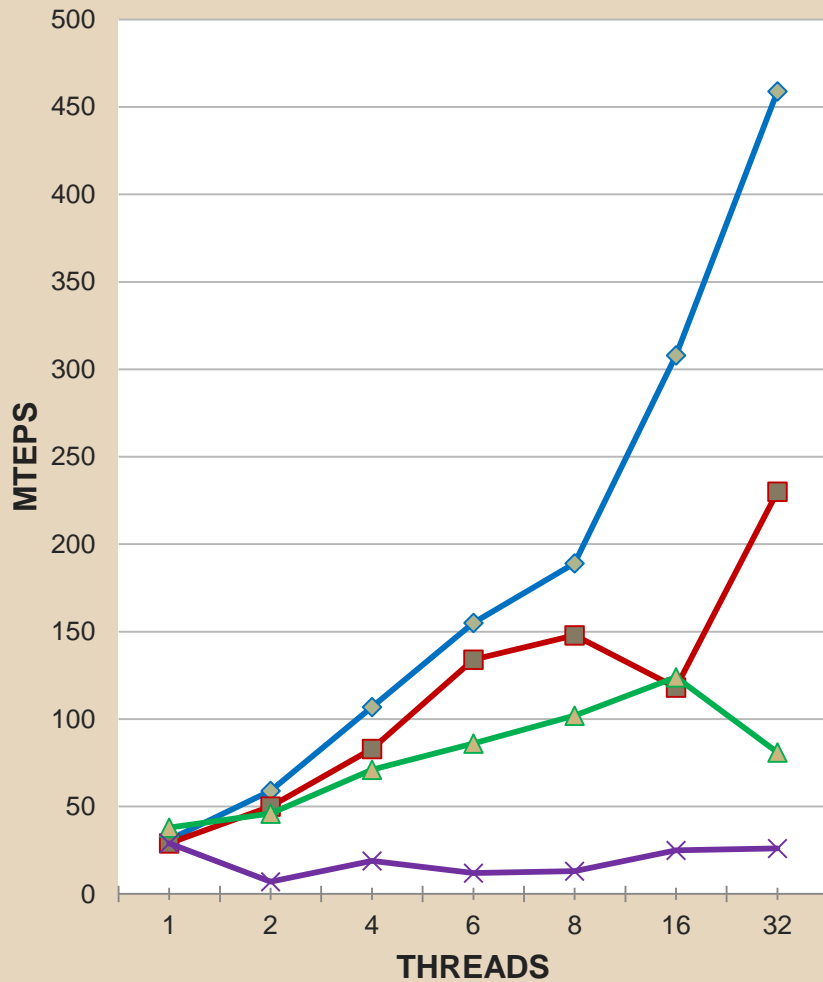


Phi 7110X

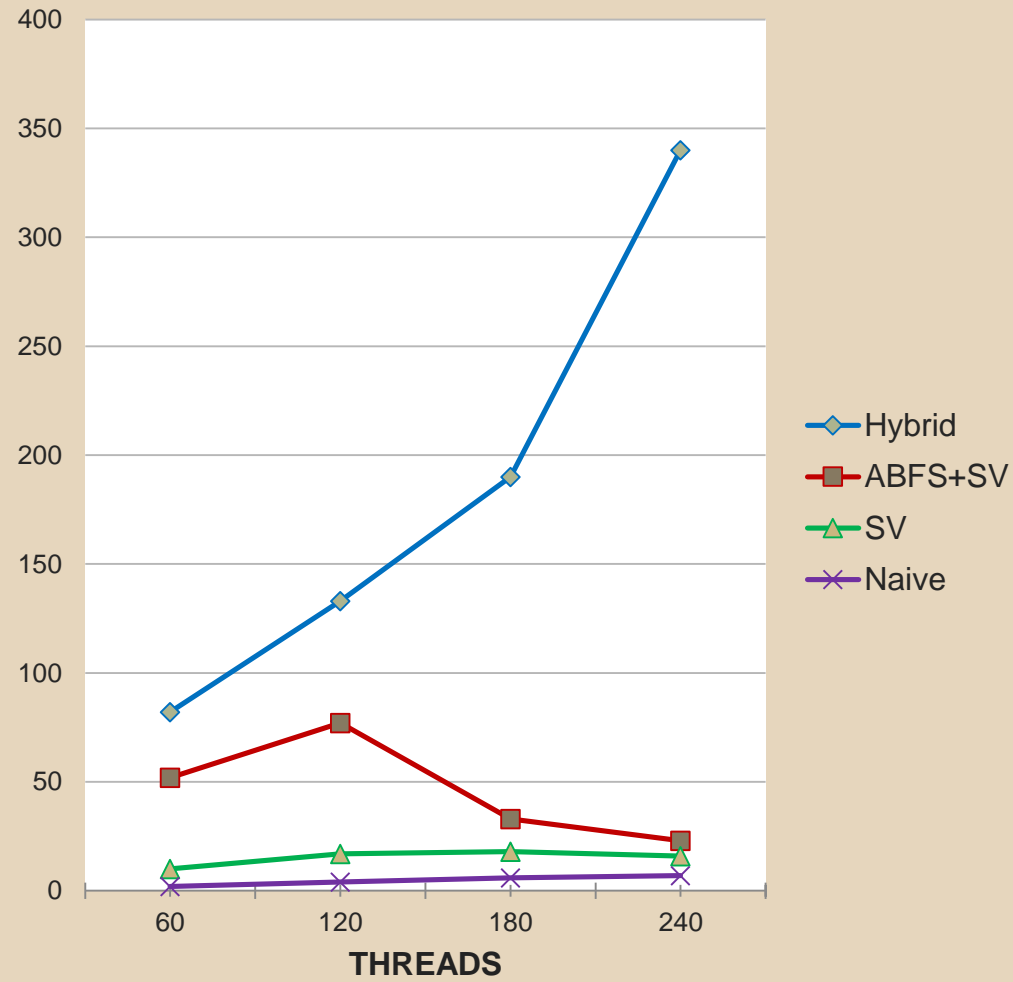


Сравнение алгоритмов на grid

Хеон E5-2690



Phi 7110X



Перспективы

- оптимизация алгоритма под Xeon Phi
- дальнейшие улучшения алгоритма
 - обрывание работы первого этапа ABFS
 - использование потоком нескольких цветов