

Post SC12 Report

Thomas Heller & Andreas Schäfer

Friedrich-Alexander-Universität Erlangen-Nürnberg

DokKolloq, 2012.11.29

Agenda

- LibGeoDecomp + HPX
 - SC12 Demo
 - future plans (HPX plugin, PyHPC, cloud)
- SC12 (Fotos)
- exascale (Horst Simon, LBNL, VecPar 2012 talk)
- WOLFHPC talk
- LibGeoDecomp

Current Team

Stephan Helou advanced LBM implementation

Björn Meier in-situ visualization and steering

Julian Hammer auto vectorization

Jochen Keil molecular dynamics with OpenCL

Dominik Thoennes Android port

Konstantin Kronfeldner MMORPG

Thomas Heller HPX Backend

Andreas Schäfer project lead

What's New Since August?

- ≥ 100 commits
- live steering (**scalable!**)
- state on Cray still somewhat moot
- dedicated performance tests
- `Region` reimplemented
- API Changes

Reimplementation of Region

- Region
 - set of coordinates
 - compressed storage
- reimplementation
 - decrease memory consumption
 - increase speed
 - drop `std::map` in favor of flat storage

Benchmarking Original Region

Test	Family	Dimensions	Time
1	Insert	(128, 128, 128)	0.002968 s
2	Insert	(512, 512, 512)	0.052137 s
3	Insert	(2048, 2048, 2048)	0.960965 s
4	Intersect	(128, 128, 128)	0.023085 s
5	Intersect	(512, 512, 512)	0.395194 s
6	Intersect	(2048, 2048, 2048)	6.81856 s

Benchmarking the New Region

Test	Family	Dimensions	Time	Speedup
1	Insert	(128, 128, 128)	0.000591 s	5.022
2	Insert	(512, 512, 512)	0.011337 s	4.599
3	Insert	(2048, 2048, 2048)	0.213258 s	4.506
4	Intersect	(128, 128, 128)	0.484808 s	0.048
5	Intersect	(512, 512, 512)	149.2 s	0.003
6	Intersect	(2048, 2048, 2048)	eternity?	zero?

- memory consumption down by 90%
- major speedup due to elimination of excessive `malloc()`
- insert is now
 - $O(n)$ instead of $O(\log n)$ for the general case (bad)
 - $O(1)$ for appending at the end
- intersect changed from $O(n \log(n))$ to $O(n^2)$ (catastrophic!)

Benchmarking the New, Fixed Region

Test	Family	Dimensions	Time	Speedup
1	Insert	(128, 128, 128)	0.000646 s	4.4257
2	Insert	(512, 512, 512)	0.012235 s	4.23098
3	Insert	(2048, 2048, 2048)	0.227063 s	4.19547
4	Intersect	(128, 128, 128)	0.00203 s	11.3015
5	Intersect	(512, 512, 512)	0.039035 s	10.0636
6	Intersect	(2048, 2048, 2048)	0.718114 s	9.39575

- fixed intersect
 - insert only at end
 - complexity: $O(n)$
- excellent overall performance

API Changes

- original declarations (in user code)
 - topology
 - nano-steps
- new/additional declarations
 - callback API (Base vs. Fixe vs. Line)
 - stencil shape and radius
- `LinePointer` interface finished
 - abundant boundary cases
 - 7x speedup for Jacobi
- variety/complexity hidden by `UpdateFunctor`

API Canges: Example

```
class Cell
{
public:
    typedef Stencils::VonNeumann<3, 1> Stencil;
    typedef Topologies::Cube<3>::Topology Topology;
    class API : public CellAPITraits::Fixed
    {};

    static inline unsigned nanoSteps()
    {
        return 1;
    }

    template<typename COORD_MAP>
    void update(const COORD_MAP& hood, const unsigned& nanoStep)
    {
        temp = (hood[FixedCoord< 0, 0, -1>()].temp ...
    }

    double temp;
};
```

What's next?

- flat_array library
- complimented by FixedNeighborhood
- adapt dendrite kernel from Aoki-sensei's lab
- benchmarks on ORNL's Titan