ExaShark+GASPI

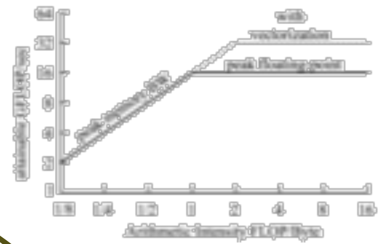
Reducing the burden to program large HPC systems since 2014

Tom vander Aa

Imen Chakroun, Mirko Rahn, Christian Simmendinger and Roel Wuyts &
The EXA2CT EU Project

EXA2CT

www.exa2ct.eu



Solvers that scale to ExaScale

10^{18}

Programming models that scale to ExaScale

TBB

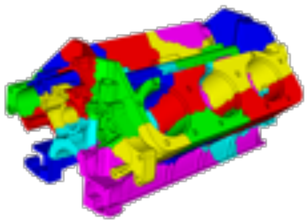
CILK

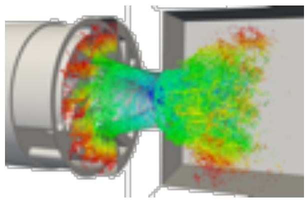
PATUS

GPI

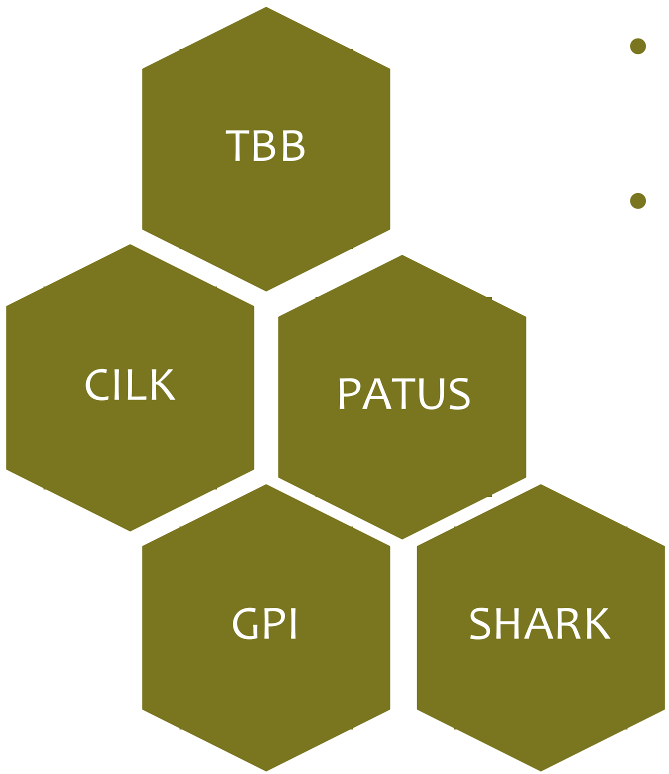
SHARK

Using relevant real-life **proto applications**

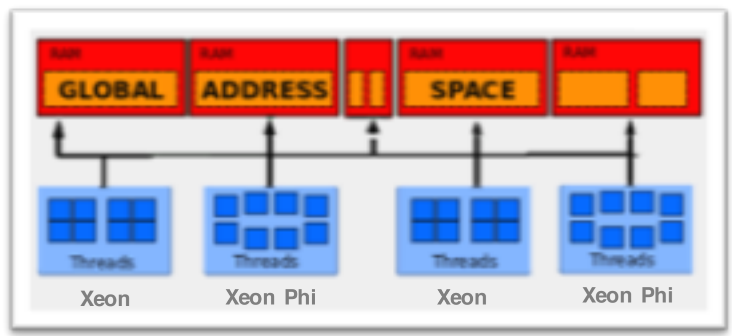




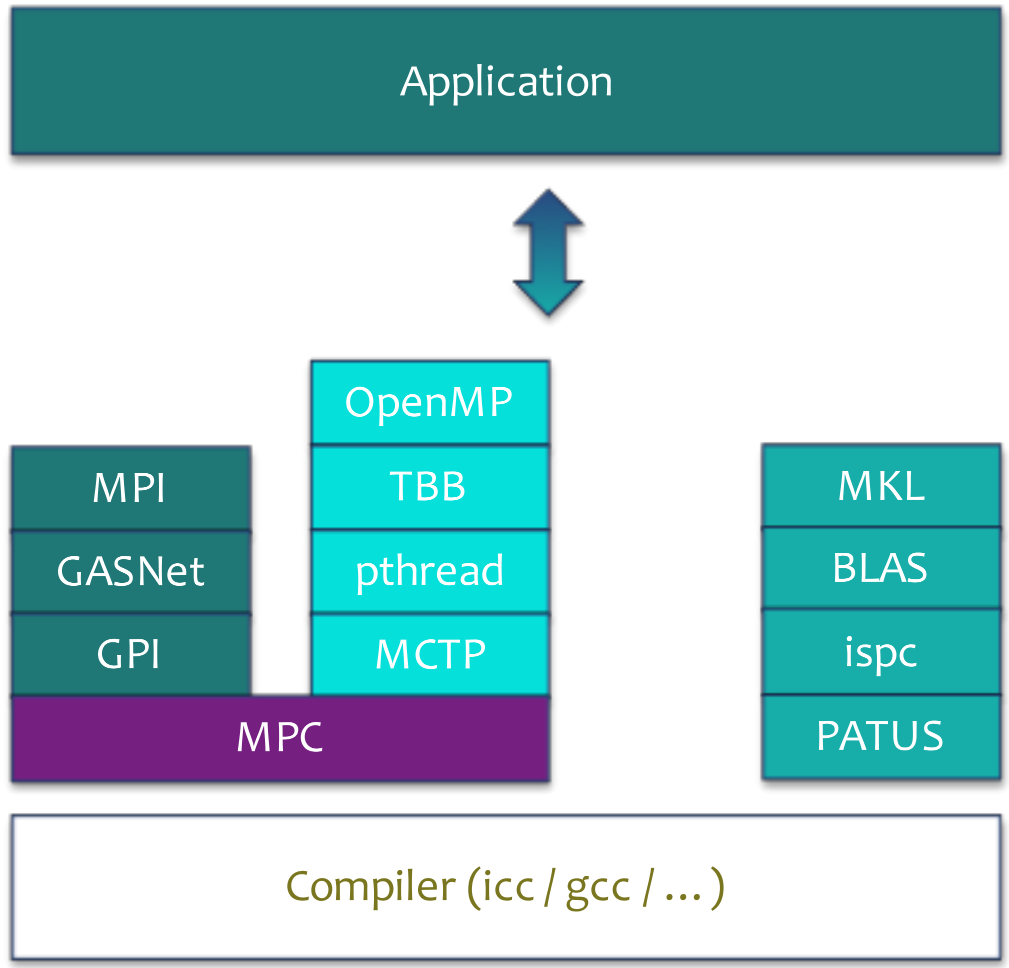
Programming Models



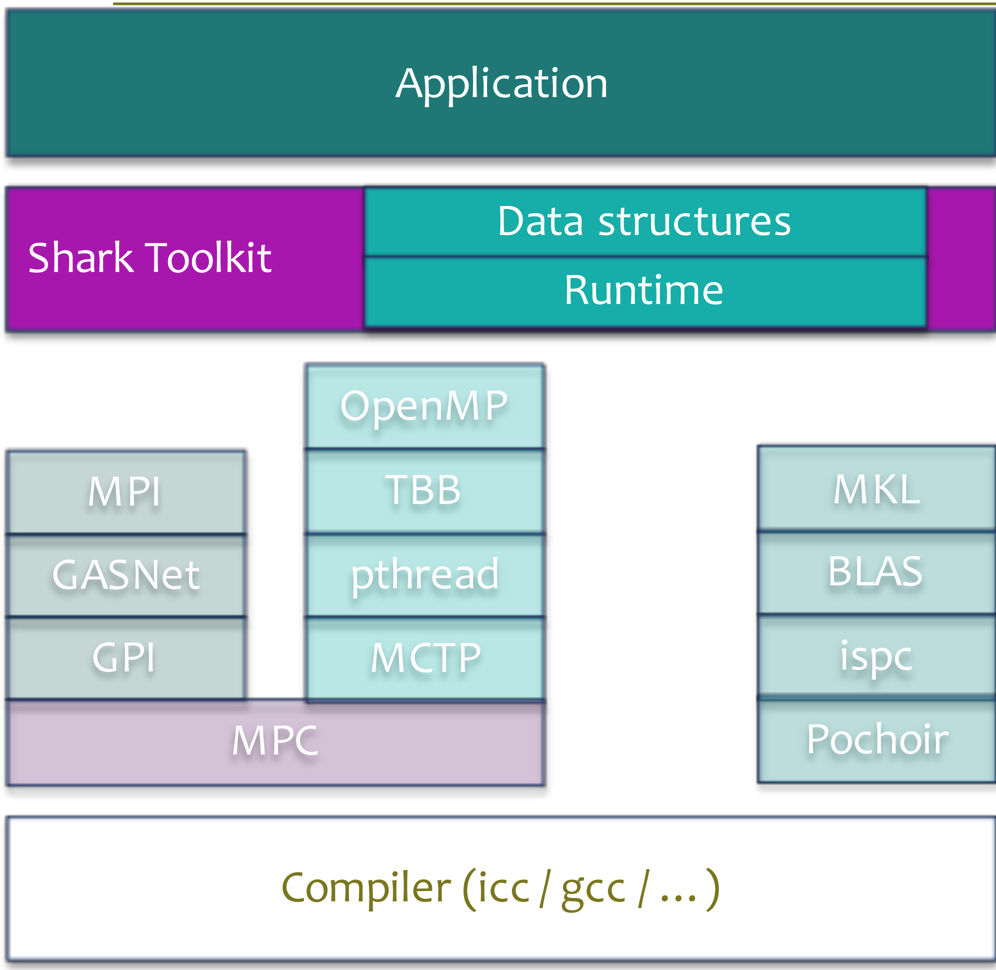
- PATUS -- stencil compiler
- ExaSHARK -- n-dimensional grids
- GPI-2 – Asynchronous PGAS



Complicated Programming Stack



The Shark in the Middle



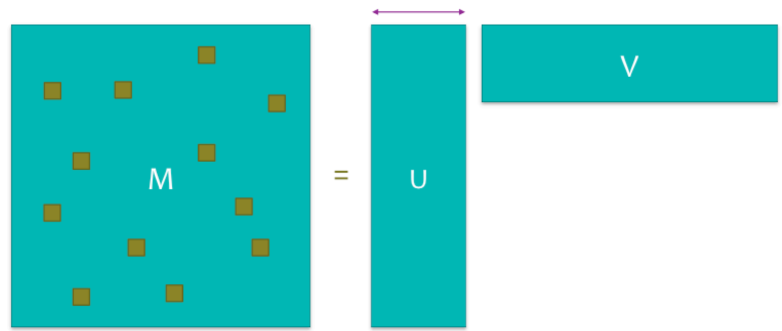
PGAS-style grids:
N-dimensional distributed grids with local operations
Specific comm. patterns
Hybrid parallelism

Inspired by:
Global Array (GA) Toolkit

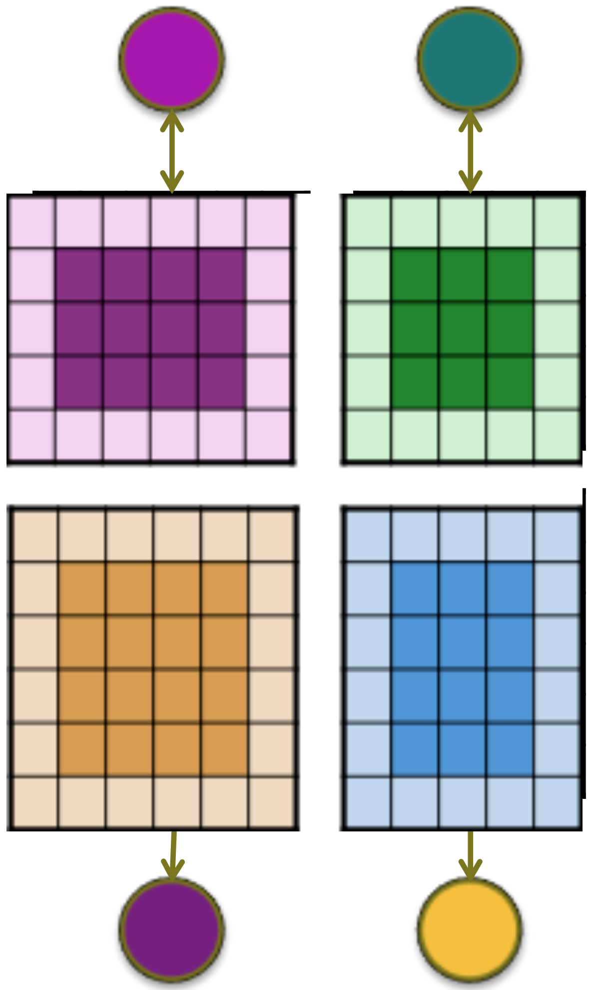
Pacific Northwest NATIONAL LABORATORY

A bit about Shark

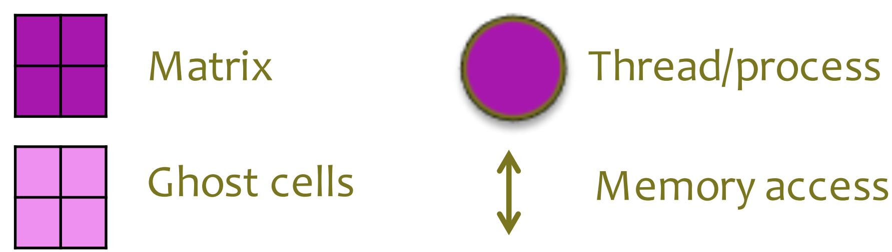
- Shark Technology
  - Shark Basics
  - C++11 features
  - Supported backends
- Applications built with Shark
  - Solvers
  - Benchmarks
  - HelSim PIC simulator
  - MACAU Recommender
- The road to ExaShark



GlobalArrays are Key

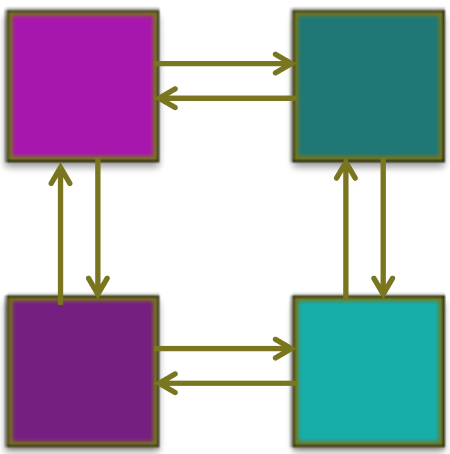


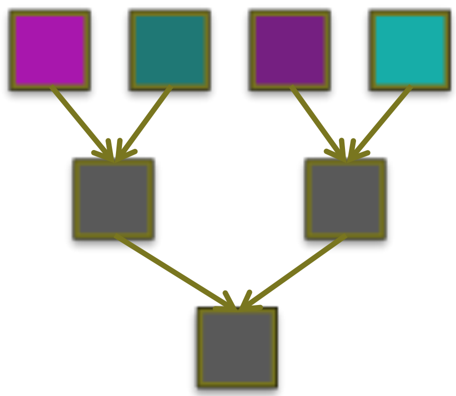
- GlobalArray:
  - Automatically or manually distributed
  - Ghost Borders
- Data-parallel Iterations
  - Locality Aware
  - C++ Expression Templates

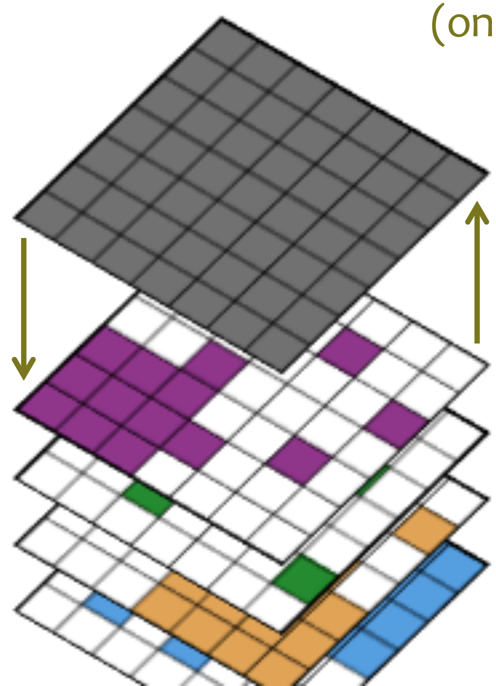


Shark Communication Patterns

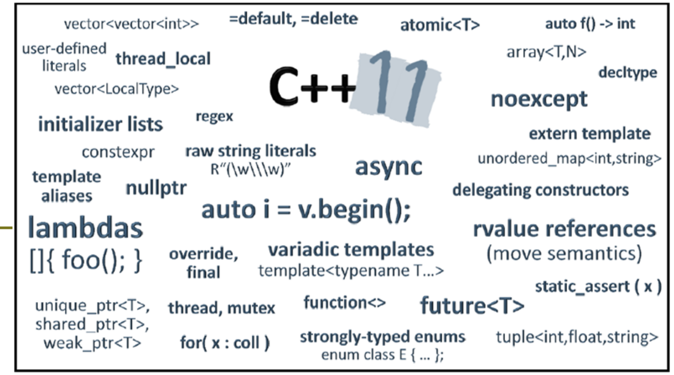
1. Ghost updates	geometric
2. Reductions	global
3. Gather/scatter with local array masks	long-distance (collective)
4. Get/put/accumulate RMA	long-distance (one-sided)







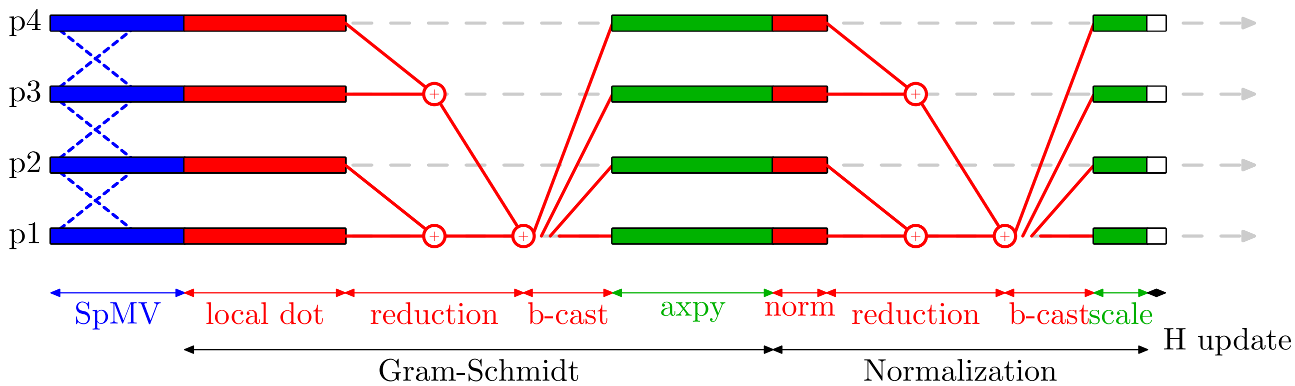
Built on C++11

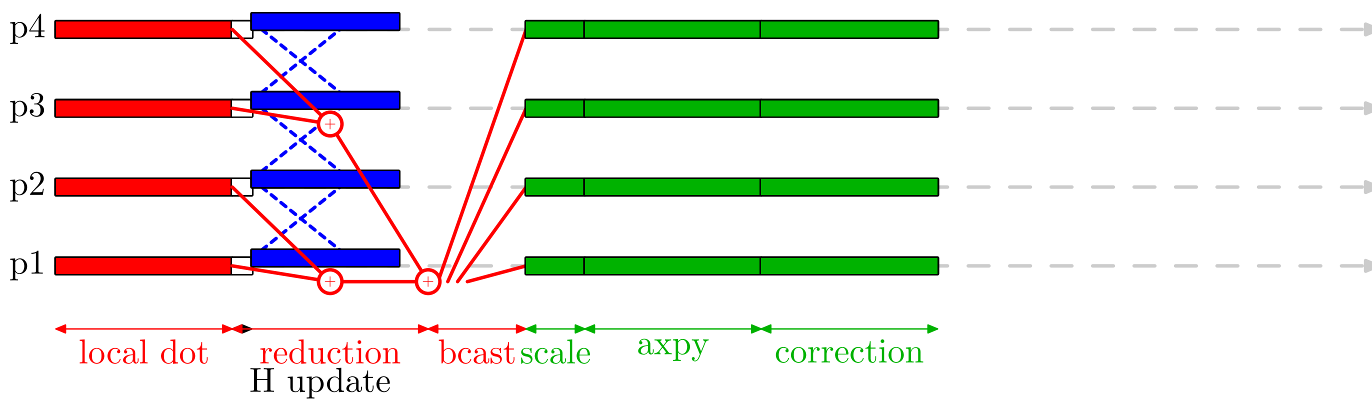


- Extensive template programming
  - Arbitrary number of dimensions
  - Any C++ type
  - Expression templates
    - Linear algebra ops in natural syntax
    - Implicitly data-parallel

- Non-blocking communication with Future<>

Shark for Solvers



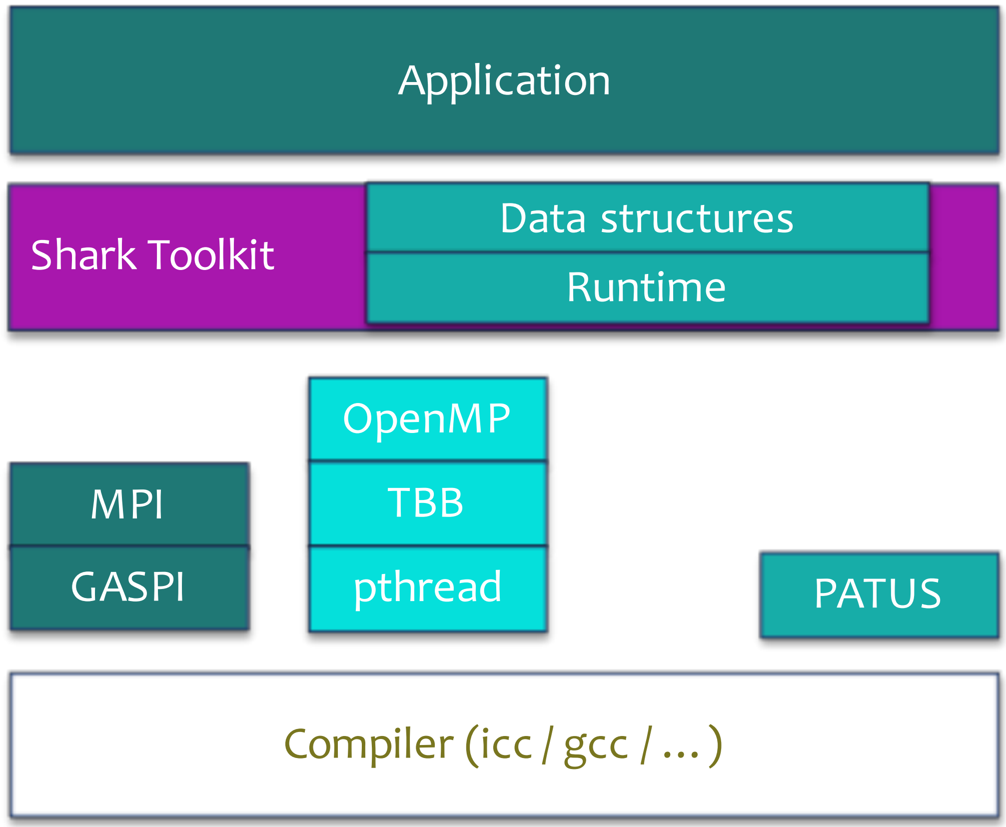


Nice Natural Syntax

1. $r = b - Ax^{(0)}$
2. $\rho_0 = ||r||_2$
3. $k = 0,\ p = r,\ x = x^{(0)}$
4. **while** $\rho \geq \epsilon$ **and** $k < k_{max}$
5. $w = Ap$
6. $\alpha = \rho_k^2/(p^T w)$
7. $x = x + \alpha p$
8. $r = r - \alpha w$
9. $\rho_{k+1} = ||r||_2$
10. $\beta = \rho_{k+1}^2/\rho_k^2$
11. $p = r + \beta p$
12. $k = k + 1$

```
r = b - Amult(x);
rho = norm2(r);
p = r;

for(k = 0; k < maxit; k++) {
    if(rho <= tol)
        break;

    w = A*p;

    alpha = rho*rho / dot(p,w);
    x = x + alpha * p;
    r = r - alpha * w;

    rho_old = rho;
    rho = norm2(r);

    beta = rho*rho / (rho_old*rho_old);
    p = r + beta * p;
}
```

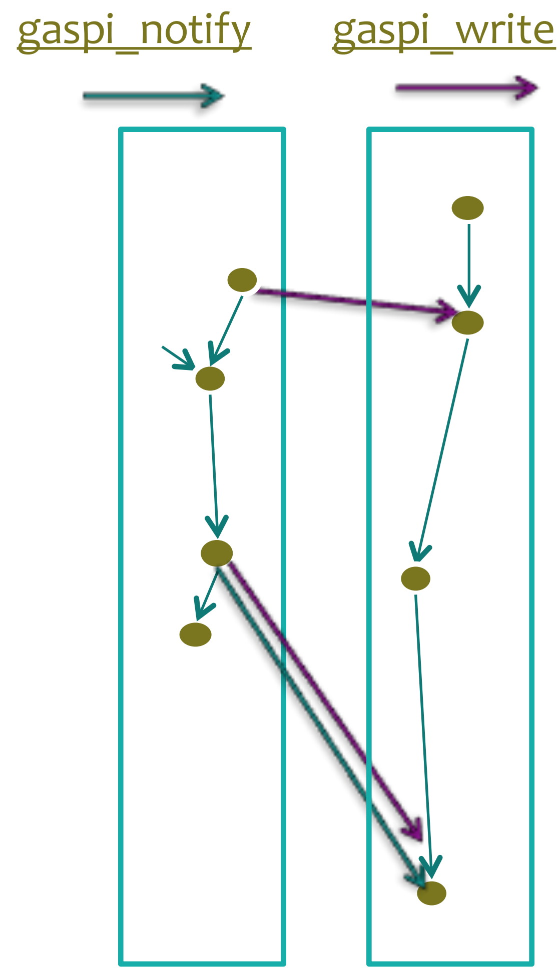
Shark Supports Many Backends



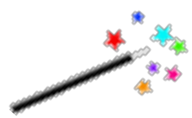
GASPI in a nutshell

PGAS API - designed to be

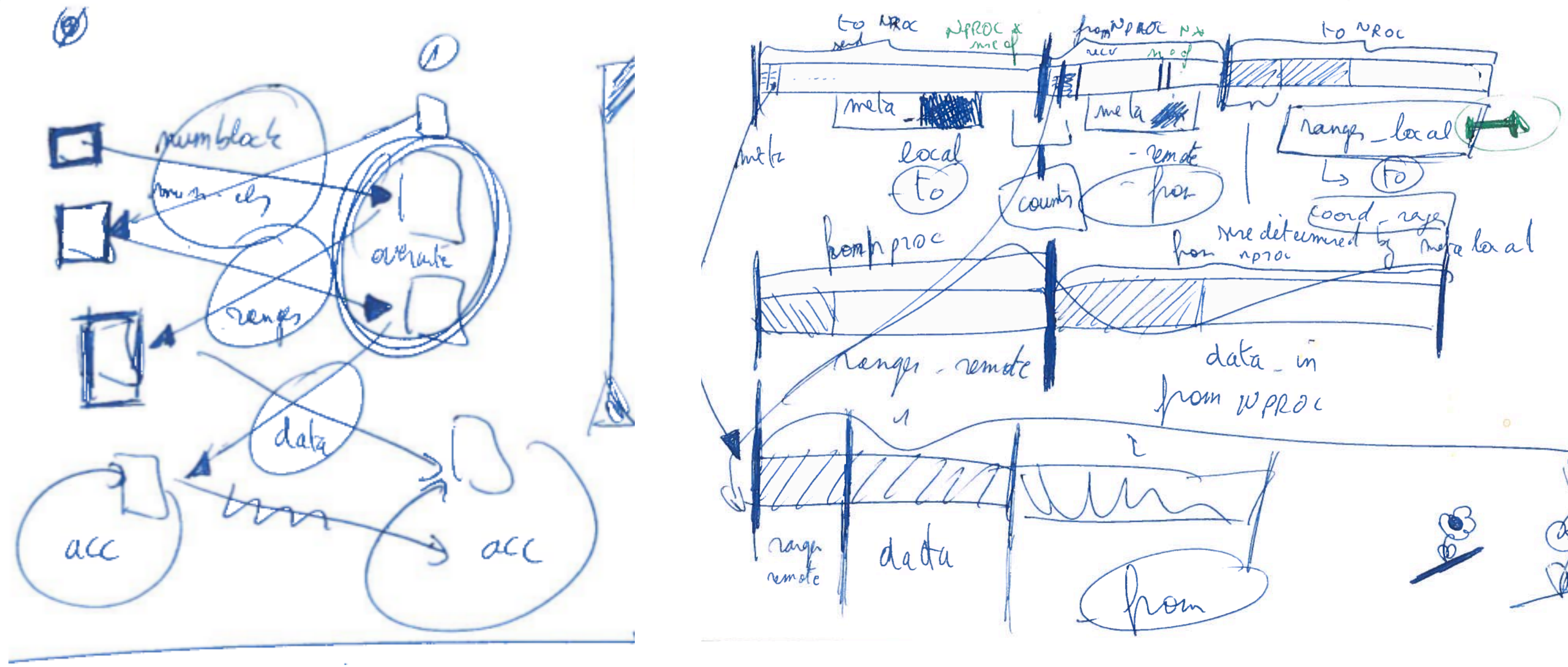
- Multithreaded
- Heterogeneous architectures
- NUMA architectures
- Global asynchronous dataflow
- Interoperability with MPI



SHARK+GASPI = Easy+Difficult

	ExaShark	GASPI
Trivial	GlobalArray()	gaspi_segment_create
Enlightening	Future<>	gaspi_notify/gaspi_notify_wait
Minimal Thinking	Put, Get	gaspi_write_block
PITA	Scatter, Gather, Accumulate	RequestQ<AccContext<GlobalArray<ndim,T>>, AccRequest<GlobalArray<ndim,T>>> q(AccContext<GlobalArray<ndim,T>>{src,acc});
Impossible	IDs	 Voodoo Magic

SHARK+GASPI, the mess



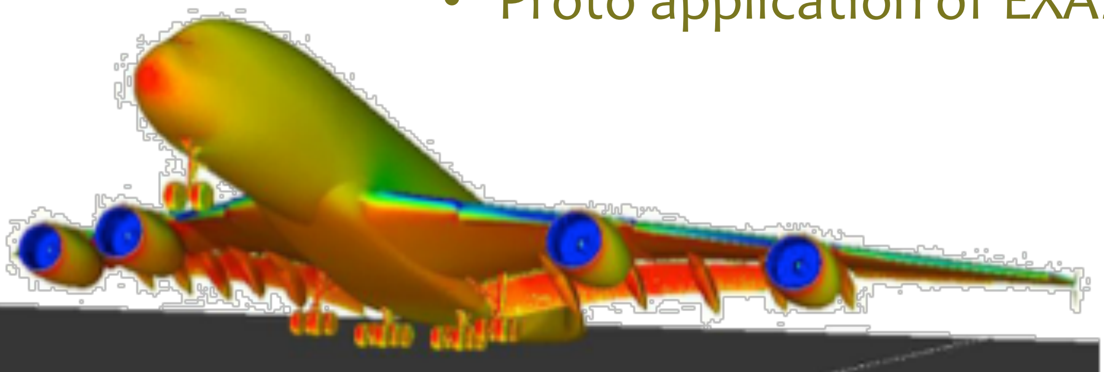
From Shark to ExaShark

- Shark Technology
  - Shark Basics
  - C++11 features
  - Supported backends

- The road to ExaShark
  - ExaScaling is Hard
  - Pure GASPI/MPI vs Shark
  - Scaling Shark

Strong Exa-Scaling is Hard

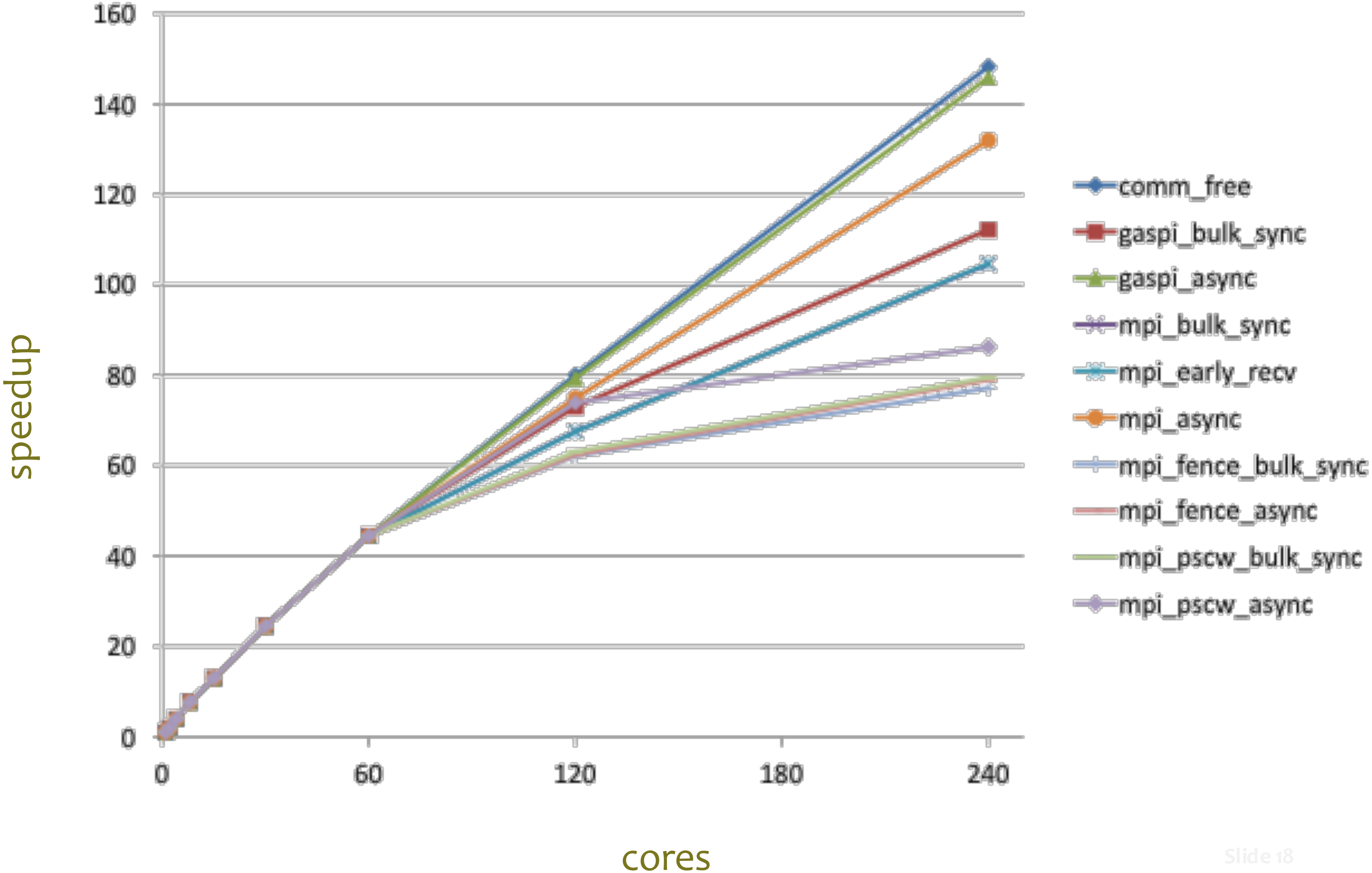
- CFD Application
  - Today: 50M mesh points
  - In ten years: 500M
- ExaScale
  - Many more cores, less memory per core
  - 50 mesh points per core
- CFD Proxy Application
  - Proto application of EXA2CT



T · · Systems ·

DLR Deutsches Zentrum für Luft- und Raumfahrt e.V. in der Helmholtz-Gemeinschaft

CFD-Proxy on Xeon-Phi



Strong Exa-Scaling is Possible

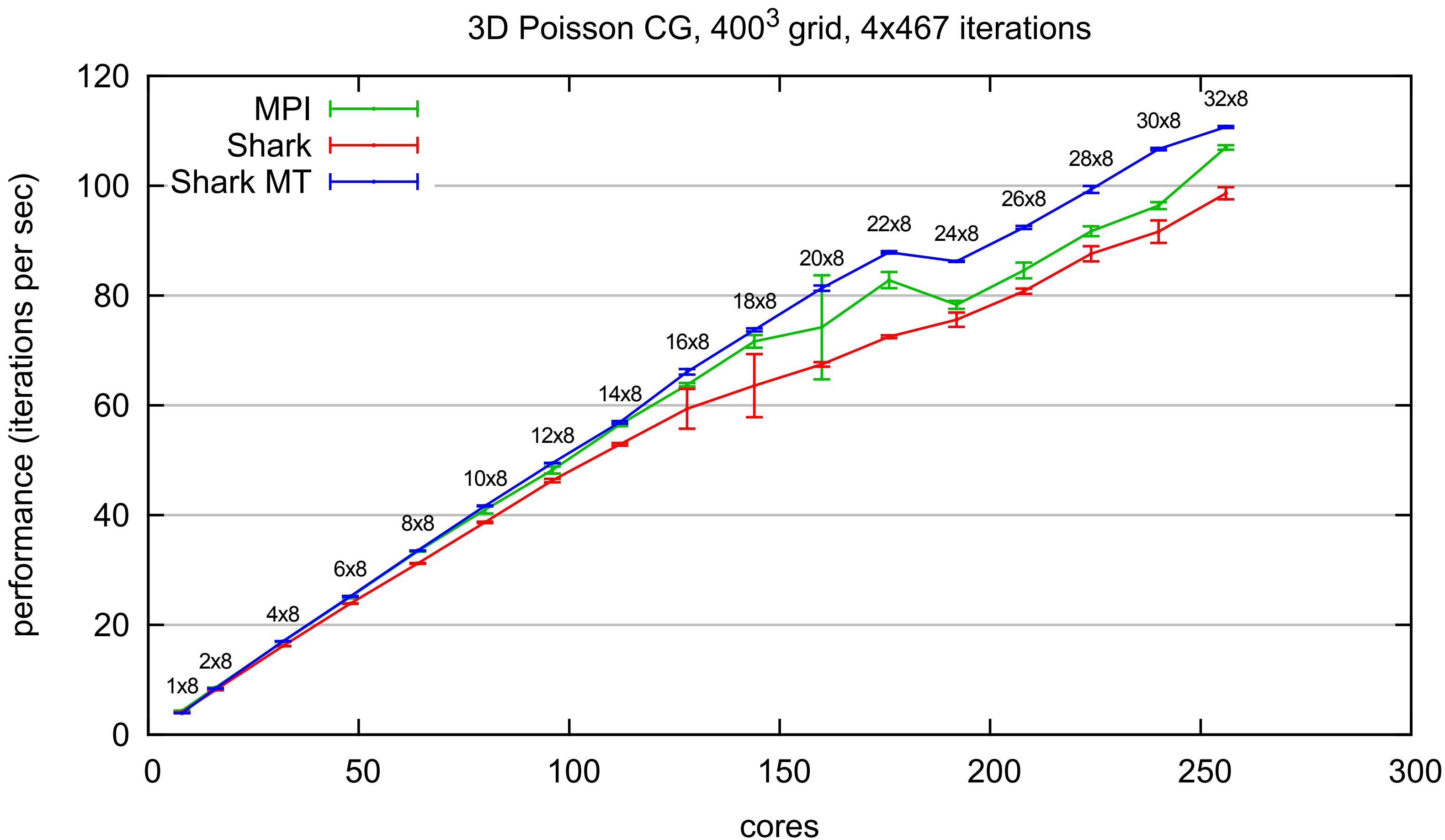
Don’ts

- Bulk Synchronous
- Single Threaded Communication
- MPI Data Types

Dos

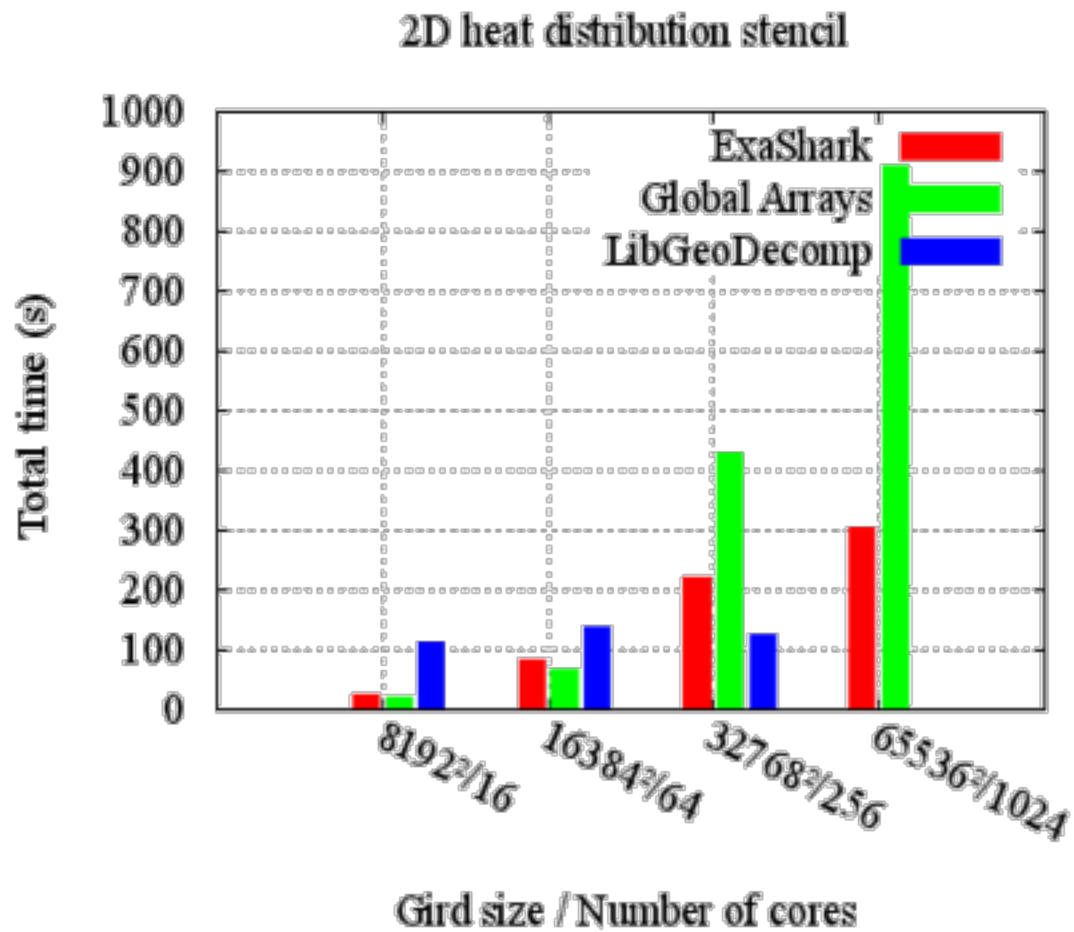
- Completely Asynchronous
  - GASPI write+notify
  - MPI ISend/IRecv
- Thread-to-thread communication
- Multi-threaded packing

SHARK on par with pure MPI

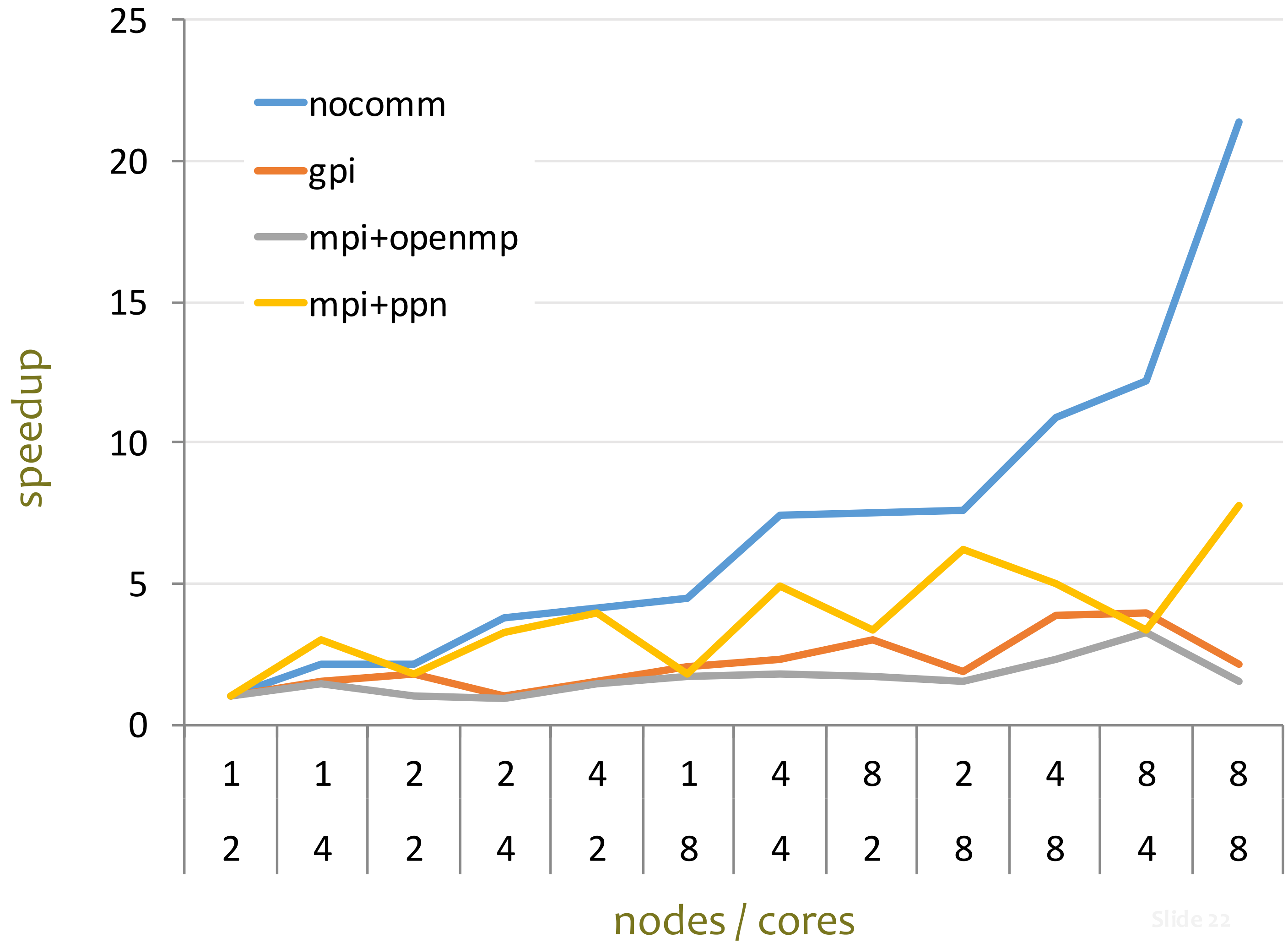


HPC15 Paper

Weak scaling on stencils



ExaScaling IS hard

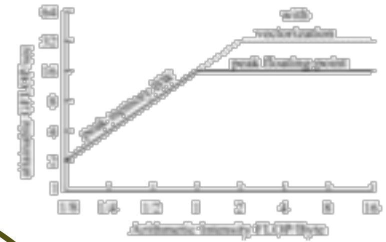


Conclusion on SHARK

- Modern high-level library for n-dim. grids
  - Support for many programming paradigms
  - Natural syntax thanks to C++
  - Performance good at node-level
- Much room for improvement
  - Active development
  - Support for GASPI/MPI Hybrid
  - Improve Scaling
    - More fine-grained a-synchronization
    - Multi-threaded Communication

EXA2CT

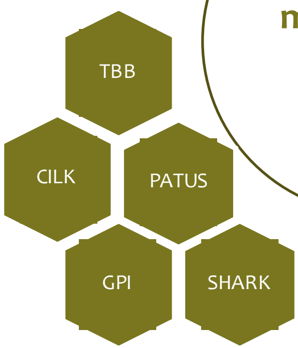
www.exa2ct.eu



Solvers that scale to ExaScale

10^{18}

Programming models that scale to ExaScale



Using relevant real-life **proto applications**