Volcanic ash transport using parallel, adaptive Ash3d

Donna Calhoun (Boise State University)

Collaborators : Melody Shih (NYU),
Roger Denlinger (USGS), Larry Mastin (USGS)
Hans Schwaiger (USGS)

SIAM Parallel Processing
March 7-10, 2018
Tokyo, Japan

2010 Eyjafjallajökull

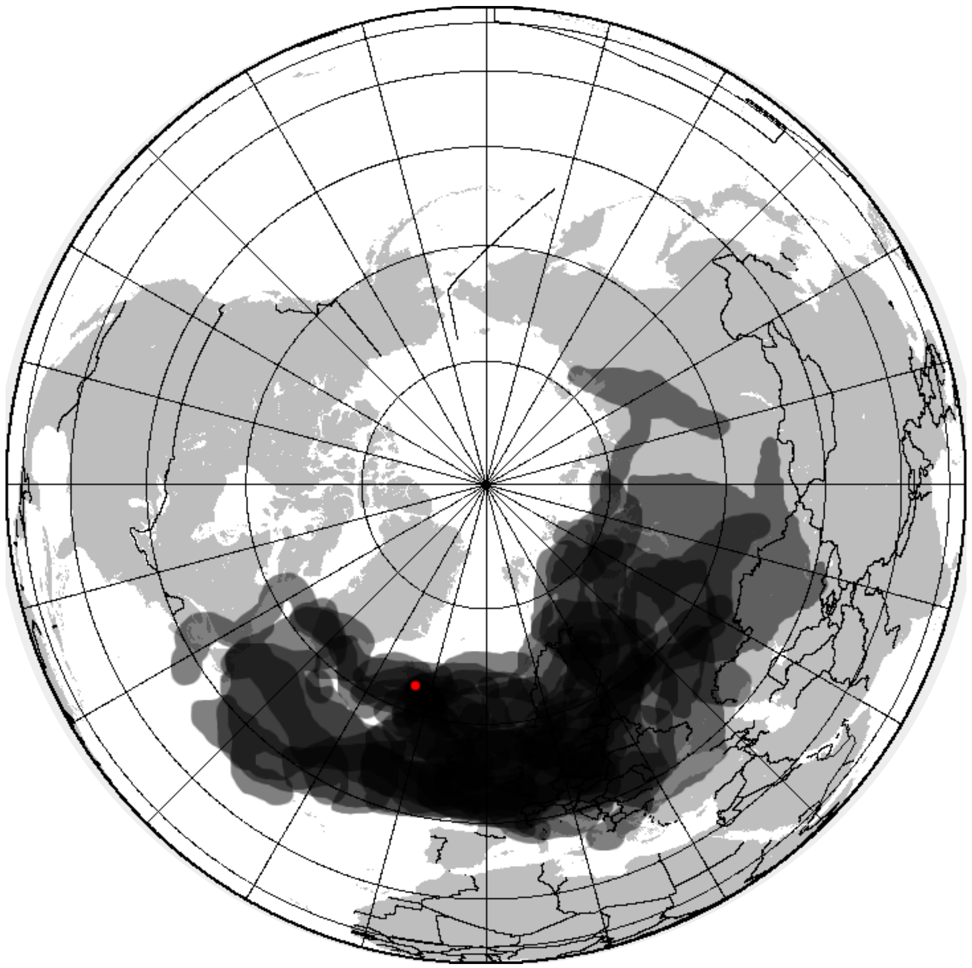
The April 15-21, 2010 Ejyafjallajokull eruption in Iceland caused unprecedented disruption to civil European airspace.

- 25 countries and over 4 million passengers were affected
- Cost to the airlines were measured in the billions of Euros (Oxford-Economics, 2010)

On April 20th, the previous "zero-ash-tolerance" policy were substituted for policies allowing for low level concentrations of ash in commercial airspace.

As a result numerical models must now be able to report ash concentration levels accurately.

Volcanic eruption 2010



2010 Eruption in Iceland

Ash3d - Ash cloud forecasts

USGS science for a changing world

USGS HOME
Contact USGS
Search VHP

Volcano Hazards Program | Find a U.S. Volcano | Assess | Prepare | Forecast | Activity | Products | Observatories | About

Volcano Updates | Monitoring | Hazards | Education | Multimedia

- About
- Volcano Updates
- Monitoring
- Hazards
- Preparedness
- Education
- Multimedia
- Products
  - Geologic Maps
  - Hazard Assessments
  - Ash Hazards Info
  - Landslide/Lahar Models
  - Coordination Plans

USGS provides volcanic ash cloud forecasts and ashfall information.

Volcanic ash is a far-reaching hazard that can affect structures, power generation and transmission, water districts, ground and air transportation, agriculture, and human health. The USGS Volcano Hazards Program offers several resources aimed at forecasting, remediating, and reporting volcanic ash.

There are specific actions to take during and after ashfall events.

The **Volcanic Ash Working Group** is a partnership between several international organizations, including the USGS, that offers *practical information about and actions to take before, during, and after volcanic ash events.*

VOLCANIC ASHFALL IMPACTS WORKING GROUP

Volcanic Ashfall Impacts Working Group offers comprehensive information about coping with ashfall.

Ash3D provides forecasts of ash clouds and ashfall.

The USGS provides forecasts of expected ash dispersion (ash clouds) and deposition (ash fall) from volcanic eruptions using a numerical atmospheric transport model

-170° -160° -150° -140°
Volcano: Shishaldin
Eruption start: 2015 12 03 19:57 UTC
Plume height: 13.7 km asl
Model valid on: 2015.12.04.1459UTC

Ash3d - Ash cloud forecasts

The Ash3d Volcanic Ash Dispersion Model

During explosive eruptions, volcanic ash can disrupt downwind populations by causing breathing problems, clogging air filters, shorting out power systems and making transportation difficult. Volcanic ash clouds can threaten air traffic by sandblasting windscreens, clogging pitot tubes, and in severe cases, causing jet engines to shut down.

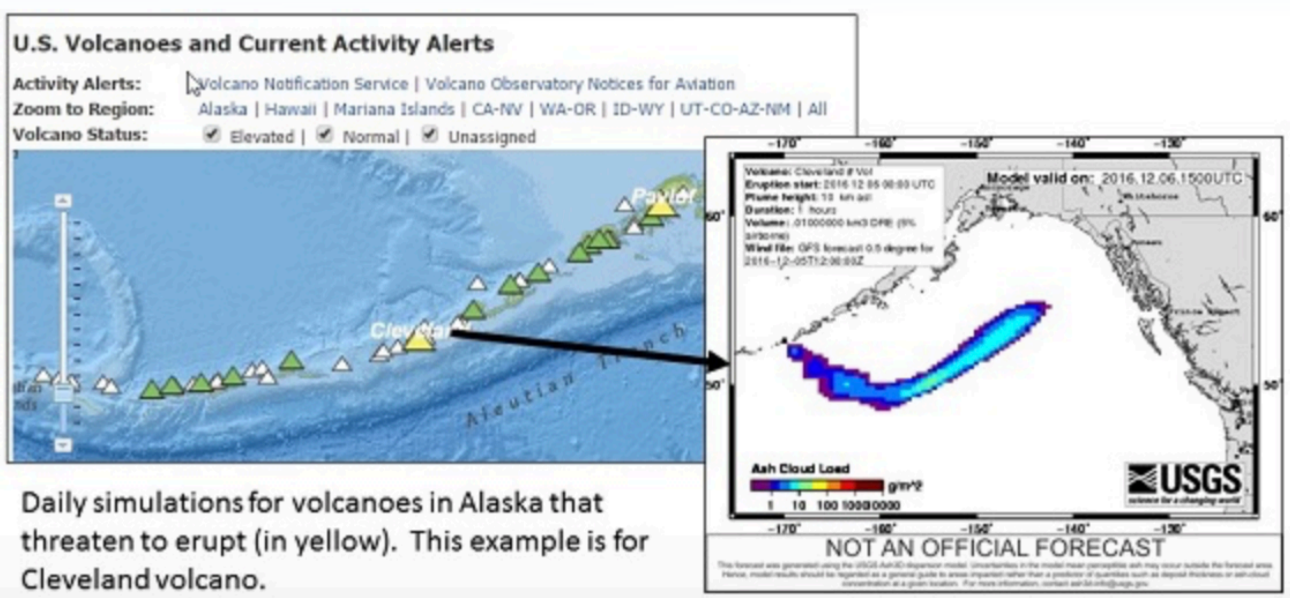
Before and during eruptions, it is important to anticipate where ash clouds might move and deposits might fall. For this purpose, the U.S. Geological Survey has developed the Ash3d model. Ash3d uses three-dimensional, time-varying wind fields

To assist other scientists. *Ash3d is used by dozens of scientists around the world, in volcano observatories, government agencies, and universities, to improve understanding of ash-producing volcanic processes and associated hazards.*

When volcanoes threaten to erupt, the USGS runs daily simulations to anticipate where ash clouds may move or deposits might land. Model results are publicly posted, for example, at the Alaska Volcano Observatory.

Ash plume drifting downwind from Cleveland Volcano, Alas

Inte



Daily simulations for volcanoes in Alaska that threaten to erupt (in yellow). This example is for Cleveland volcano.

USGS science for a changing world

Ash3d | Public Run Results | Resources

Login

Username

Password

Login

USGS - Ash3d daily forecasts

USGS science for a changing world

USGS Home | Contact USGS | Search USGS

Ash3d | Public Run Results | Resources | Login/Request Account

Ash3d Public Run Results

Baker

Baker today	Eruption Start: 2018-03-02 17:51:00 UTC	Deposit Results
Baker today	Eruption Start: 2018-03-02 17:50:00 UTC	Airborne Results

Bogoslof

Bogoslof Aware	Eruption Start: 2018-03-02 17:38:00 UTC	Deposit Results
Bogoslof Aware	Eruption Start: 2018-03-02 17:38:00 UTC	Airborne Results

Cleveland

Cleveland Aware	Eruption Start: 2018-03-02 17:39:00 UTC	Deposit Results
Cleveland Aware	Eruption Start: 2018-03-02 17:39:00 UTC	Airborne Results

Ash3d - details

$$q_t + \nabla((\mathbf{u} + v_s)q) = Q(\mathbf{x}, t)$$

Model

- **Wind fields** interpolated in space and time from re-analysis data available from various weather services.
- **Source term** modeled with a Suzuki distribution (Suzuki, 1983).
- **Multiple grain sizes** tracked; Fall velocity and deposition modeled
- Extensive library for reading **meteorological data** in many re-analysis formats

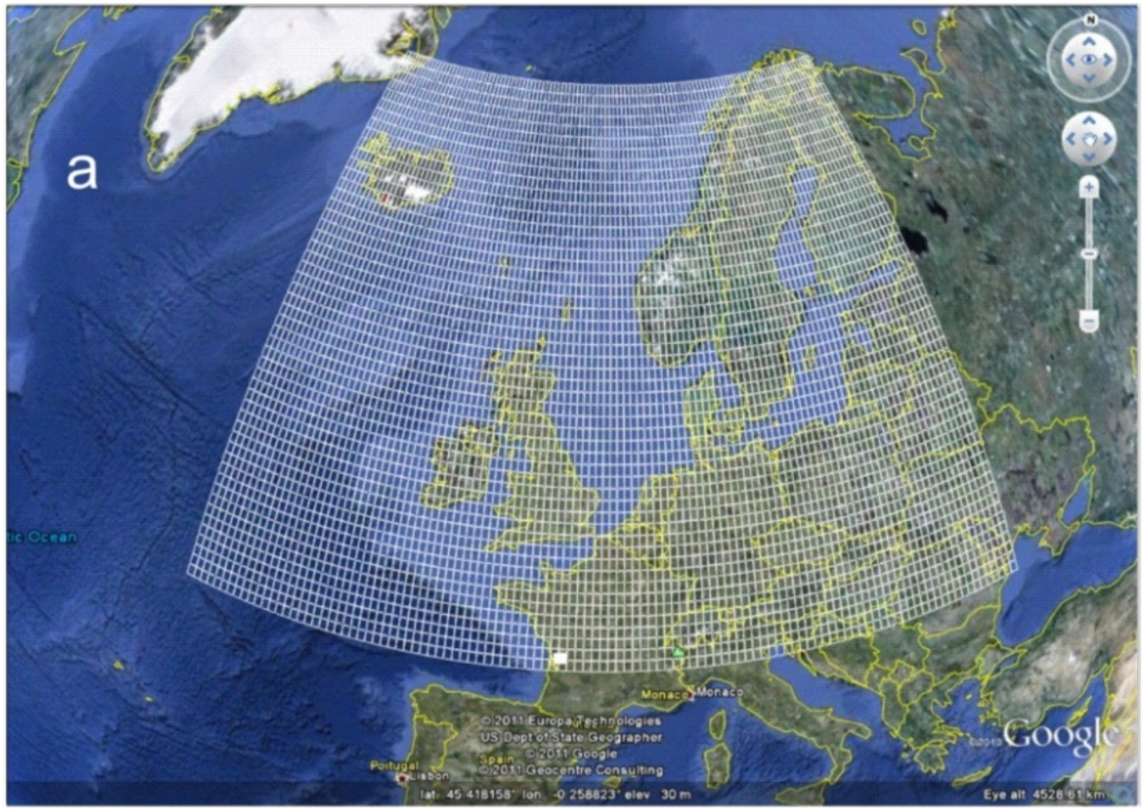
Numerical scheme.

- Explicit second order finite volume upwind schemes (CTU, DCU) and Lagrangian schemes.
- **Serial** F90 code run on USGS servers through web portal
- Source code available upon request

Computational domain

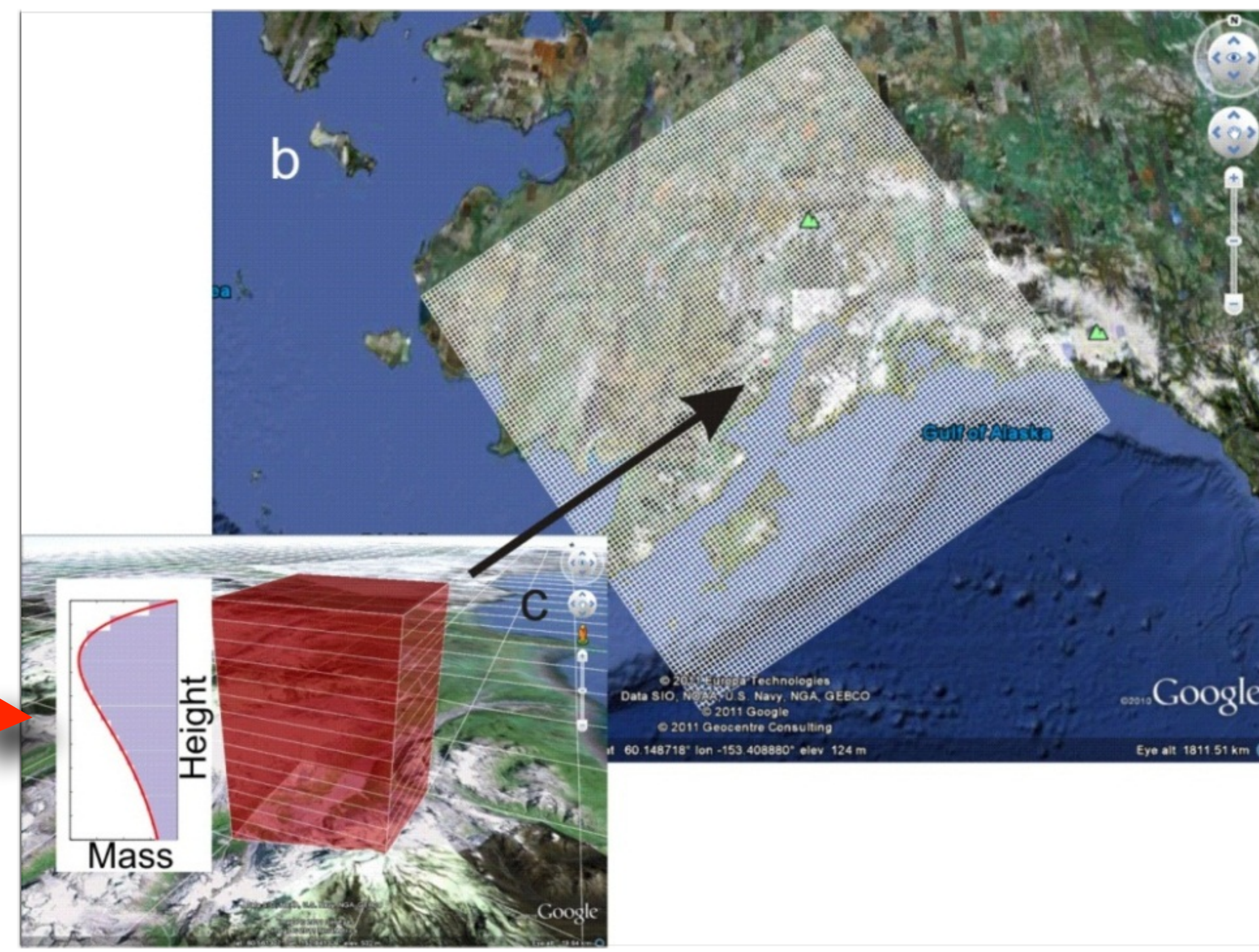
- 3d **regional lat-long** computational domain; 25 cells in the vertical

Ash3d - computational domain

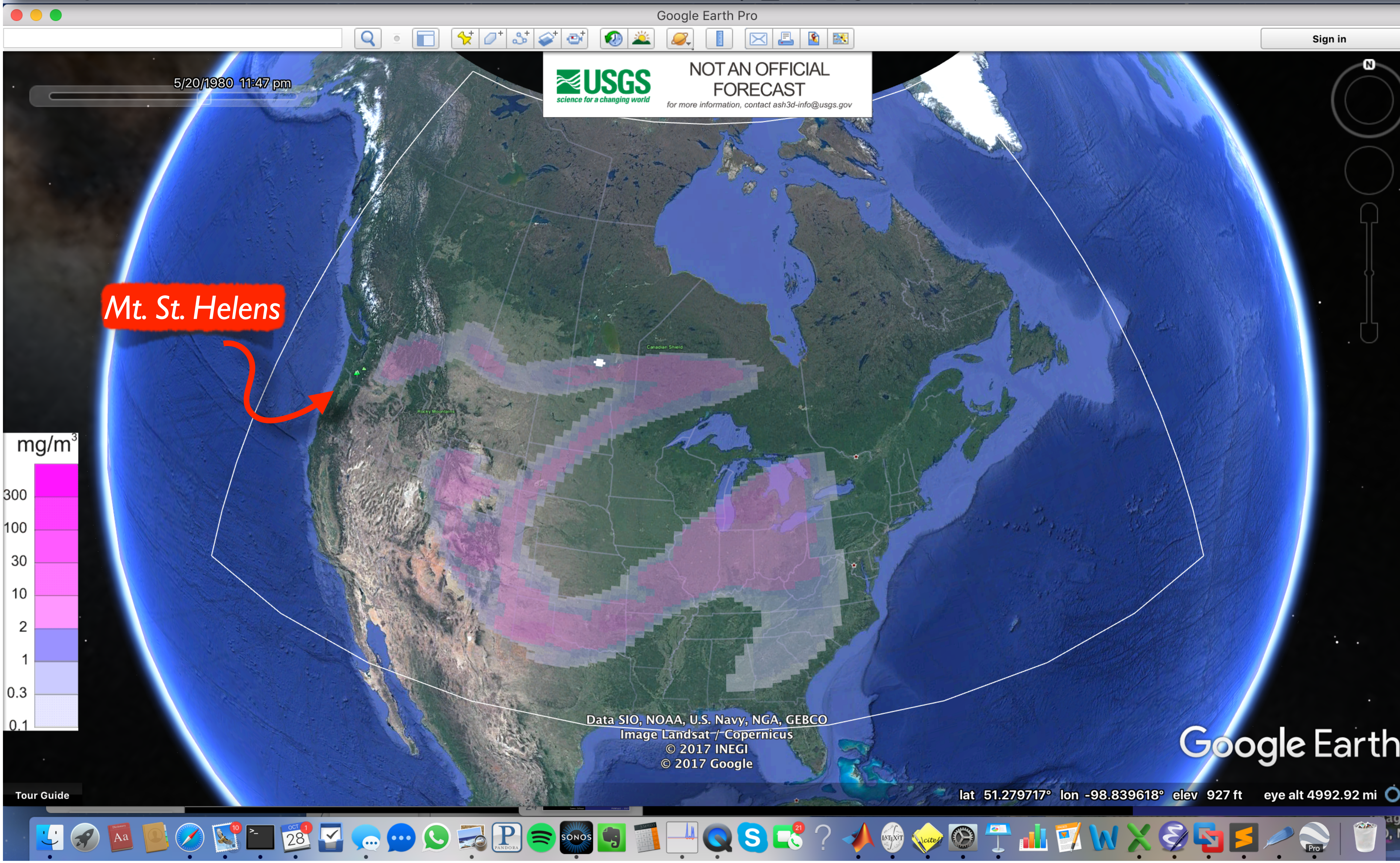


Users Guide and Reference to Ash3d -- A Three Dimensional Model for Eulerian Tephra Transport and Deposition, USGS Open File Report 2013 - 1122

Source term model :Ash initialized in single column



Ash3d : Results



Volcanic ash transport models

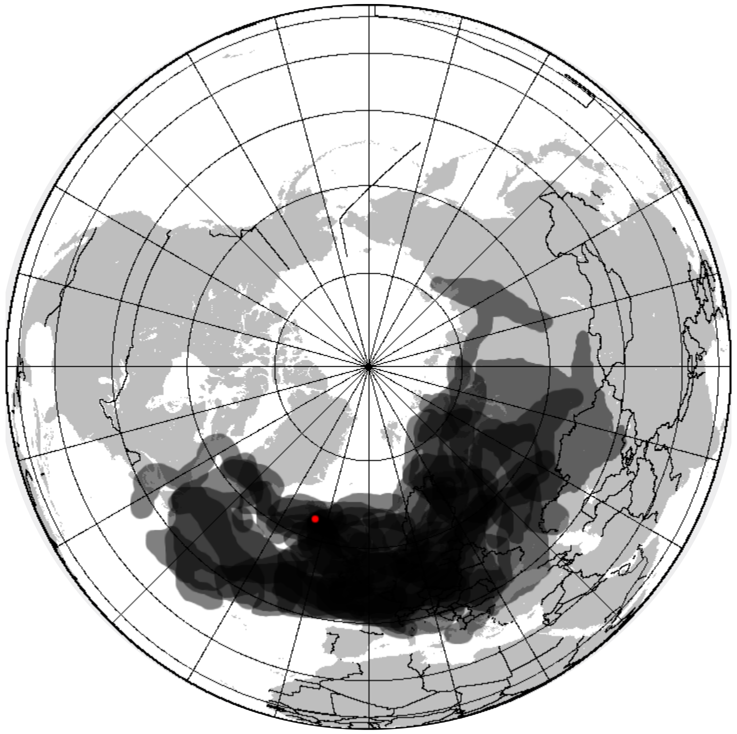
Eruptions lasting several days or weeks can place heavy demands on computational resources.

- Observational evidence supports the idea that there is considerable variation in concentration levels in tracers in the atmosphere, not just large diffuse clouds of ash.
- Important to track small scale structures in smooth, even coarse-resolution wind fields (Behrens, MWR 2000)

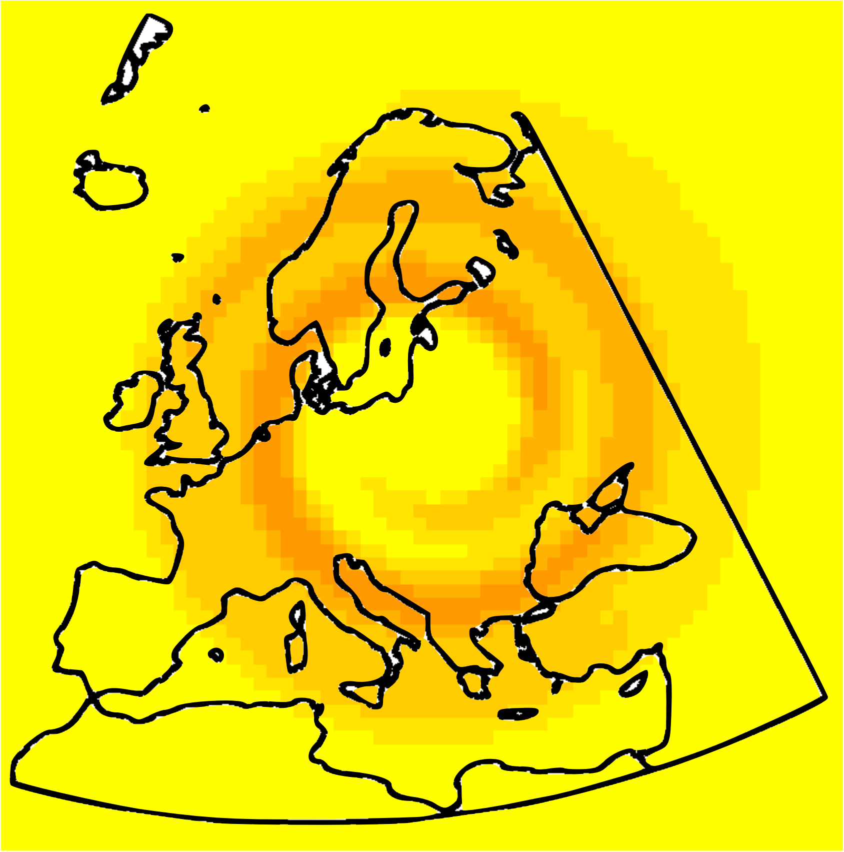
Ash plume drifting downwind from Cleveland Volcano, Alaska on May 3, 2006. Public photo taken from the International Space Station.

Volcanic eruption 2010

Horizontal scale ~ O(5000km)

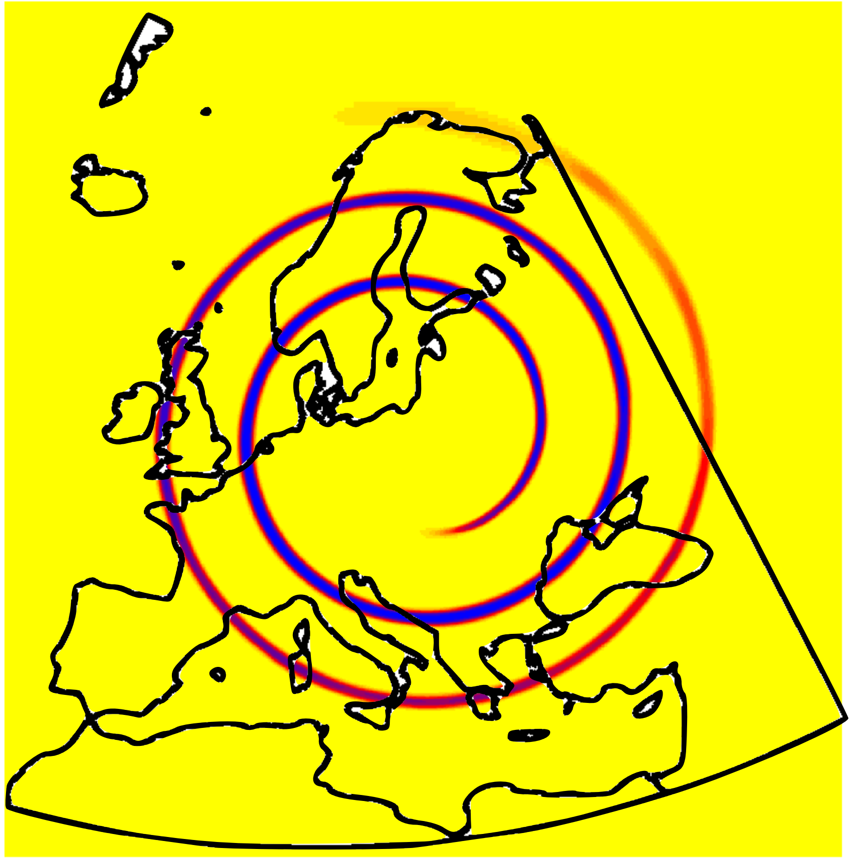


2010 Eruption in Iceland



64 x 64 (~80 km resolution)

Evolution has to be tracked over several days



1024 x 1024 (~5 km resolution)

Ash3d

Ash3d is an obvious candidate for including parallel, spatial adaptivity.

- **Obvious refinement criteria** : Use ash threshold level to determine ash location (“feature based refinement”)
- Advection problem with explicit time stepping is **relatively easy** to convert to adaptive code
- **Cartesian grid** can be decomposed into subproblems in an obvious way

Benefits

- **Shorter run times** : multiple runs for uncertainty analysis and forecasting
- Get **higher resolution** in reasonable, operational time frames

However, developing a parallel, adaptive code is non-trivial!

ForestClaw Project

A parallel, adaptive library for logically Cartesian, mapped, multi-block domains

Features of ForestClaw include :

- Uses the **highly scalable p4est** dynamic grid management library (C. Burstedde, Univ. of Bonn, Germany)
- Each leaf of the quadtree contains a fixed, uniform grid,
- Optional multi-rate time stepping strategy,
- Has **mapped, multi-block** capabilities, (cubed-sphere, for example) to allow for flexibility in physical domains,
- **Modular design** gives user flexibility in extending ForestClaw with Cartesian grid based solvers and packages.
- Uses essentially the same algorithmic components as patch-based AMR (e.g. Berger-Oliger-Colella)

Thanks to NSF for supporting this work

www.forestclaw.org

Extending ForestClaw

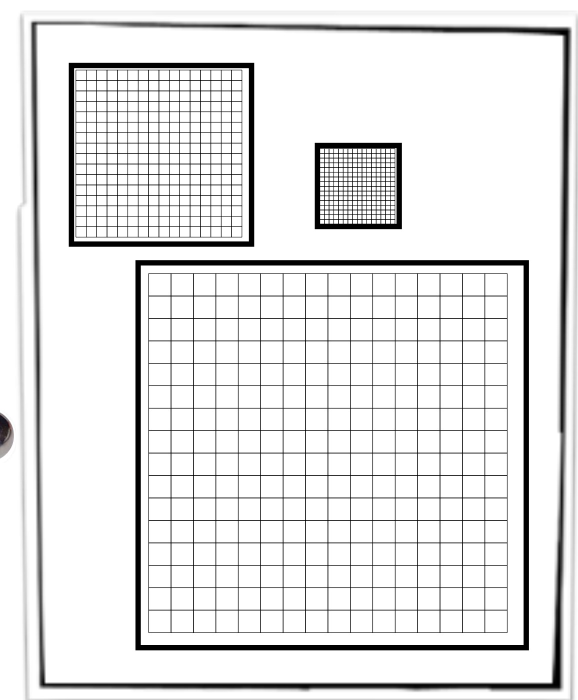
Core routines

p4est

ForestClaw

Time stepping, dynamic grid management, input/output.

Patches



Tasks (tagging, building patches, etc) customized through use of function pointers stored in virtual tables

Extensions

clawpack4.6

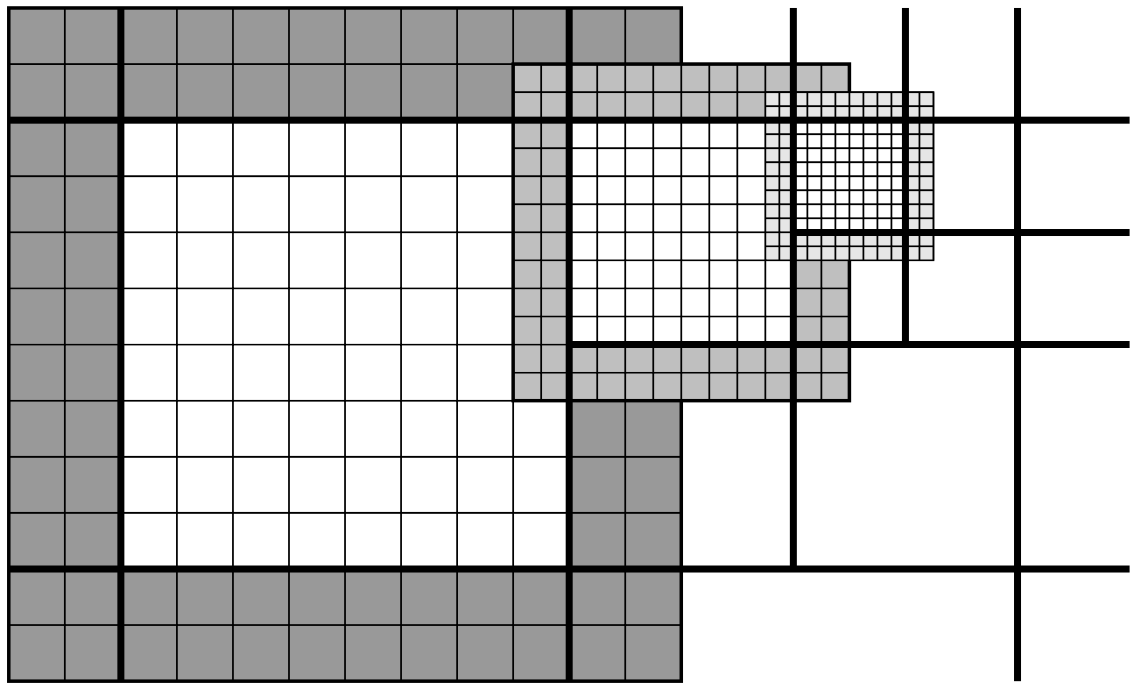
clawpack5

geoclaw

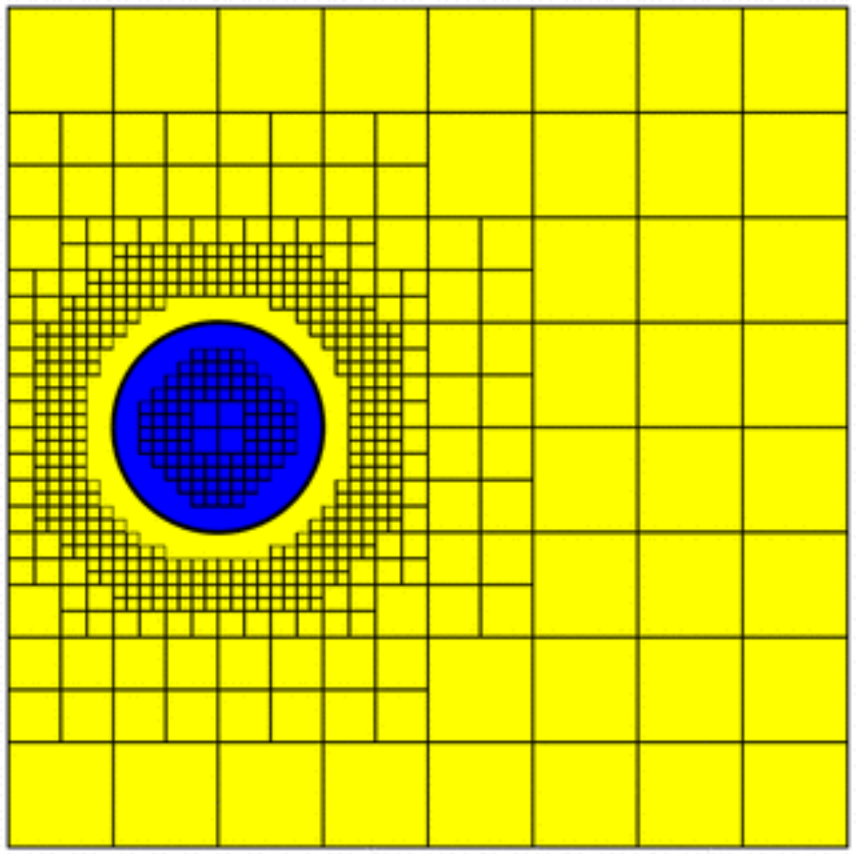
ash3d

ForestClaw is mostly C; solvers extension libraries left largely untouched (in original Fortran)

ForestClaw adaptivity



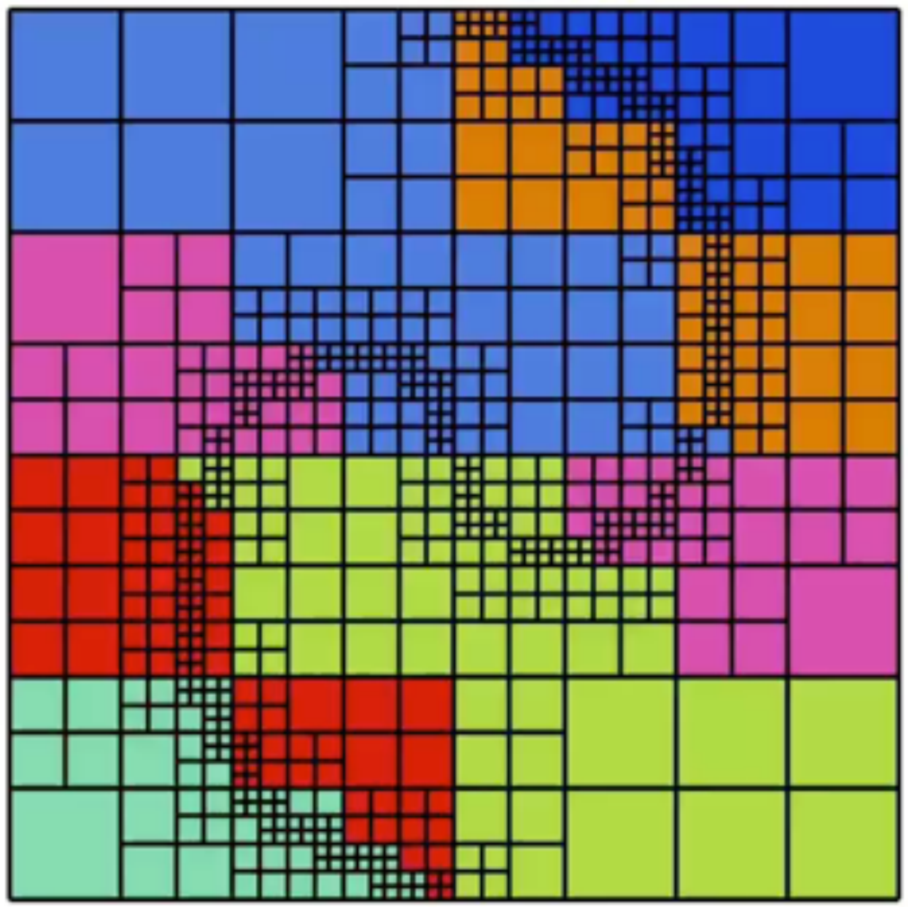
Each quadrant is a single logically grid, designed for finite volume or finite difference solvers.



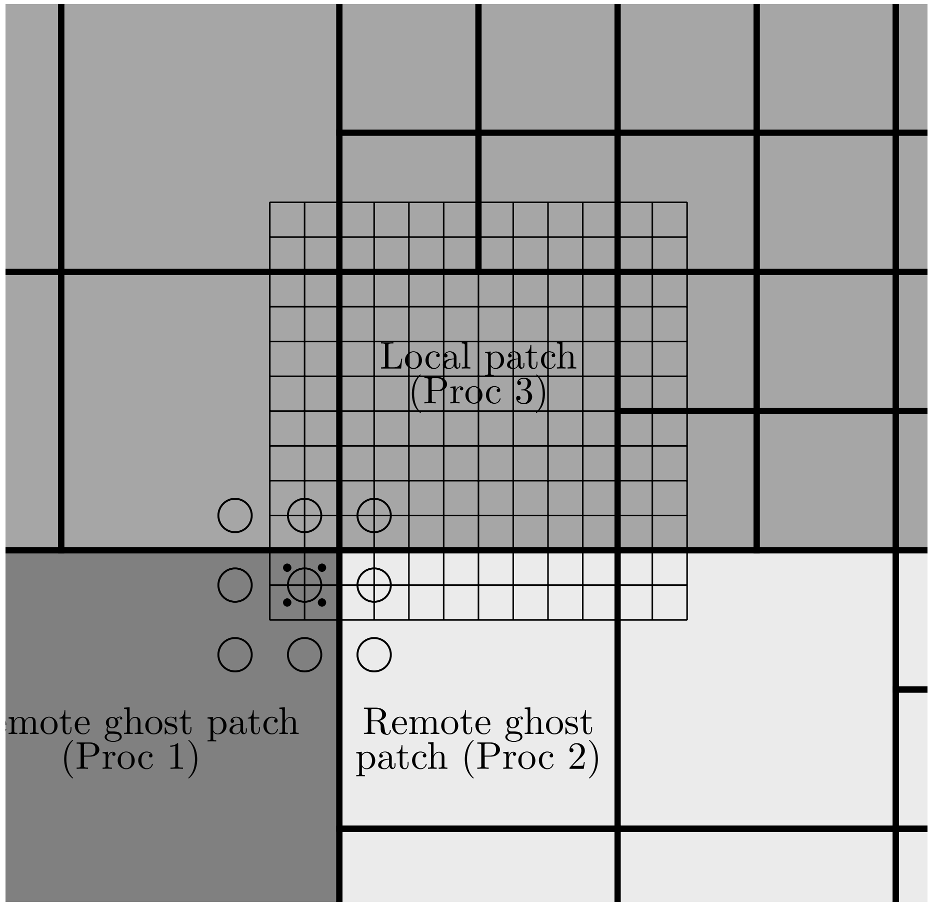
(Behrens, MWR 2000)

Regridding, connectivity done using p4est (www.p4est.org)

ForestClaw - Parallelism



p4est : Load balancing using a space filling curve; scaling shown to 16K processors



Fine grid corner ghost cells at corners where 3 or more processors meet

D. Calhoun and C. Burstedde, *"ForestClaw : A parallel algorithm for patch-based adaptive mesh refinement on a forest of quadtrees", (submitted), 2017. [arXiv: 1703.03116])*

ForestClaw + Ash3d

How will Ash3d benefit from a ForestClaw implementation?

- **Domain decomposition** for better cache performance (even if run on a single processor)
- **Parallelism using space filling curve**. Even a uniformly refined grid can benefit from fact that number of processors need not divide number of grids evenly (17 grids on 4 processors is automatically handled).
- **Dynamic spatial adaptivity.** Track thin filaments and accurately model ash spatial distribution of concentration.
- Benefit from **ForestClaw build system** (autoconf/automake)

Goal is to use Ash3d essentially as is and run solvers in ForestClaw patches

Ash3d - code

```
subroutine MesoInterpolater(TimeNow,Load_MesoSteps,Interval_Frac)

!     Subroutine that interpolates to obtain the current wind field
!
         use precis_param
         use global_param
         use solution
         use wind_grid
         use time_data
         use mesh
         use Source
         use Tephra
         use atmosphere
         use MetReader
```

```
module AdvectionHorz

use precis_param
use global_param

use AdvectionHorz_CTU
use AdvectionHorz_DCU
use AdvectionHorz_SL

contains

!*******************
```

Source.f90

```
module Source

use precis_param
use global_param

integer, parameter :: MAXCustSrc = 10

real(kind=ip) :: x_volcano, y_volcano
real(kind=ip) :: lat_volcano, lon_volc
real(kind=ip) :: z_volcano
```

```
subroutine alloc_arrays

use precis_param
use global_param
use Source
use mesh
use solution
use Output_Vars
use time_data
use wind_grid
```

Initialize ForestClaw patches

Original Ash3d code

```
MODULE ash3d
  REAL(kind=8), POINTER :: concen(:,:,:,:)
  INTEGER(KIND=4) :: n
END
```

Called for every patch

```
SUBROUTINE ALLOC_ARRAY()
  USE ash3d
  IMPLICIT NONE
  ALLOCATE(concen(0:n+1,0:n+1,0:n+1,2))
END
```

ForestClaw code

```
struct patch
{
    double *concen;
};
```

```
void init_patch(patch* p)
{
  ALLOC_ARRAY();
  ATTACH_C_PTRS(&p->concen);
}
```

```
SUBROUTINE ATTACH_C_PTRS(fc_concen_ptr)
  USE iso_c_binding
  USE ash3d
  IMPLICIT NONE

  TYPE(C_PTR) :: fc_concen_ptr

  fc_concen_ptr = C_LOC(concen)
END
```

Attach C ptr to module array

Update solution on each patch

ForestClaw code

```
void update(patch* p)
{
    ATTACH_MOD_PTRS(&p->concen);
    ADVECTION_HORZ();
}
```

Original Ash3d code

```
SUBROUTINE ADVECTION_HORZ()
  USE ash3d_
  IMPLICIT NONE

  concen(:,:,:,1) = concen(:,:,:,0) + dt*(horz. update)
END
```

```
SUBROUTINE ATTACH_MOD_PTRS(fc_concen_ptr)
  USE iso_c_binding
  USE ash3d
  IMPLICIT NONE

  TYPE(C_PTR) :: fc_concen_ptr
  REAL(KIND=8), POINTER :: ptr(:)

  CALL C_F_POINTER(fc_concen_ptr,ptr,[1])
  concen(0:n+1,0:n+1,0:n+1,2) => ptr
END
```

Attach module array to data associated with C ptr

Lessons learned

What helped make this doable project?

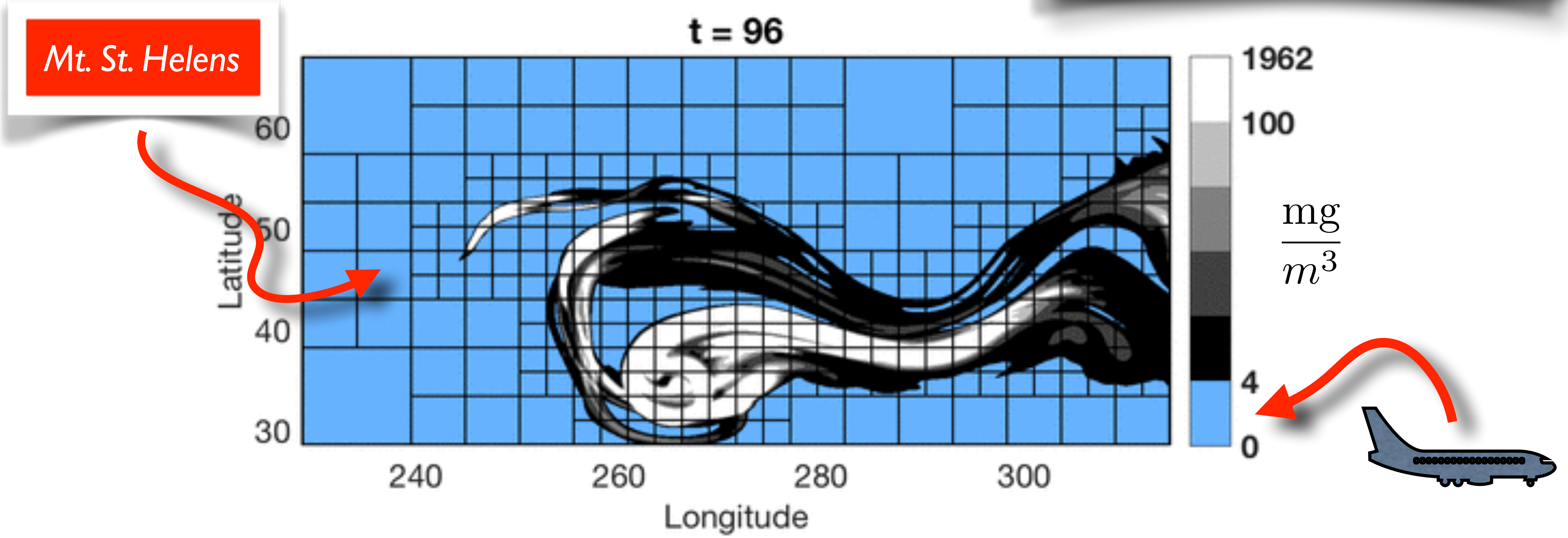
- The original Ash3d code has very **clearly defined relationship** between subroutines and module arrays,
- **Finite volume solver** based very similar to those already built into ForestClaw,
- **Extensive re-factoring of ForestClaw** made it easy to include 3d Ash3d patches in the 2d ForestClaw infrastructure

Other considerations

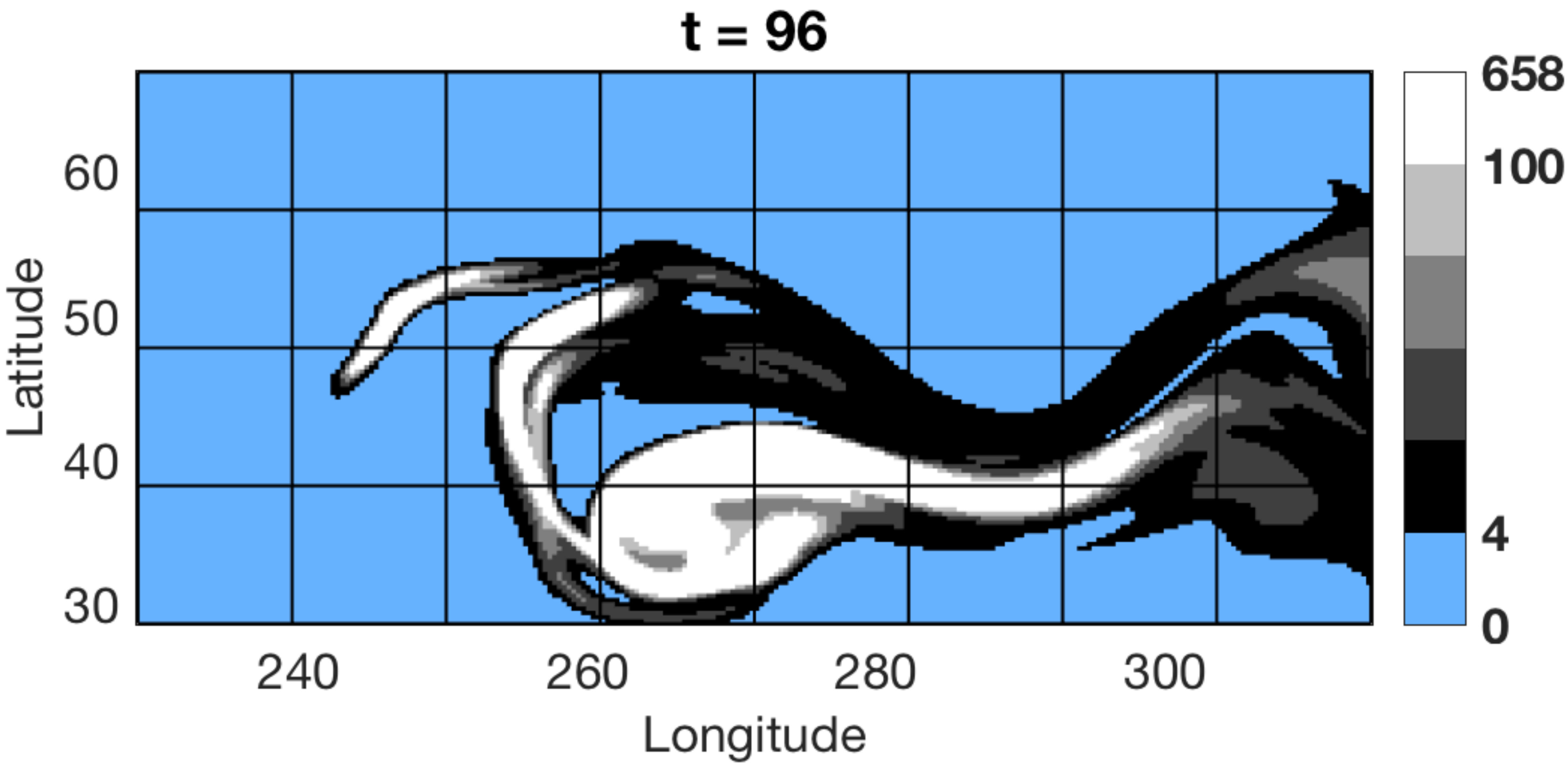
- **Boundary conditions** had to be modified to distinguish between interior patch boundaries and original physical patch boundaries
- Control **console output** by manipulating file units to selectively suppress output, depending on user-defined verbosity levels and MPI rank.
- **Coordination** of ForestClaw **parameters** and original Ash3d parameters
- **Output written in serial**, using original NETCDF file handling routines
- **Conservation** at coarse/fine boundaries
- Autoconf/Automake **build system** doesn't play so well with F90 modules

Mt. St. Helens

- May 18, 1980; largest eruption in lower 48 since 1915
- 1024 x 512 x 25 effective resolution
- 32x32x25 blocks (Surface to volume ratio = 0.25)
- 96 hours simulation time; results averaged in the vertical
- 16 virtual CPUS; 64 GB Ram; RHEL “cloud server”

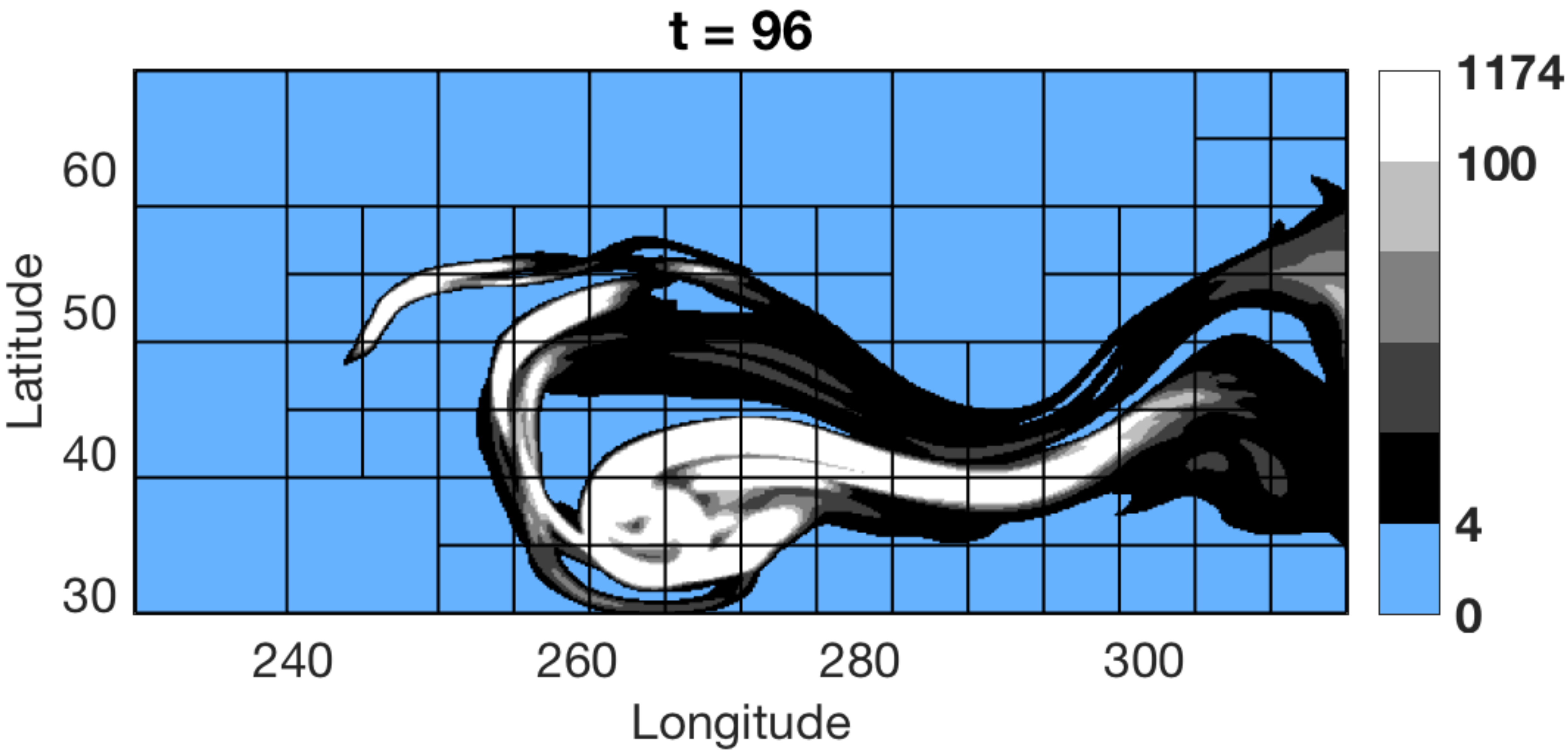


Adaptive Ash3d



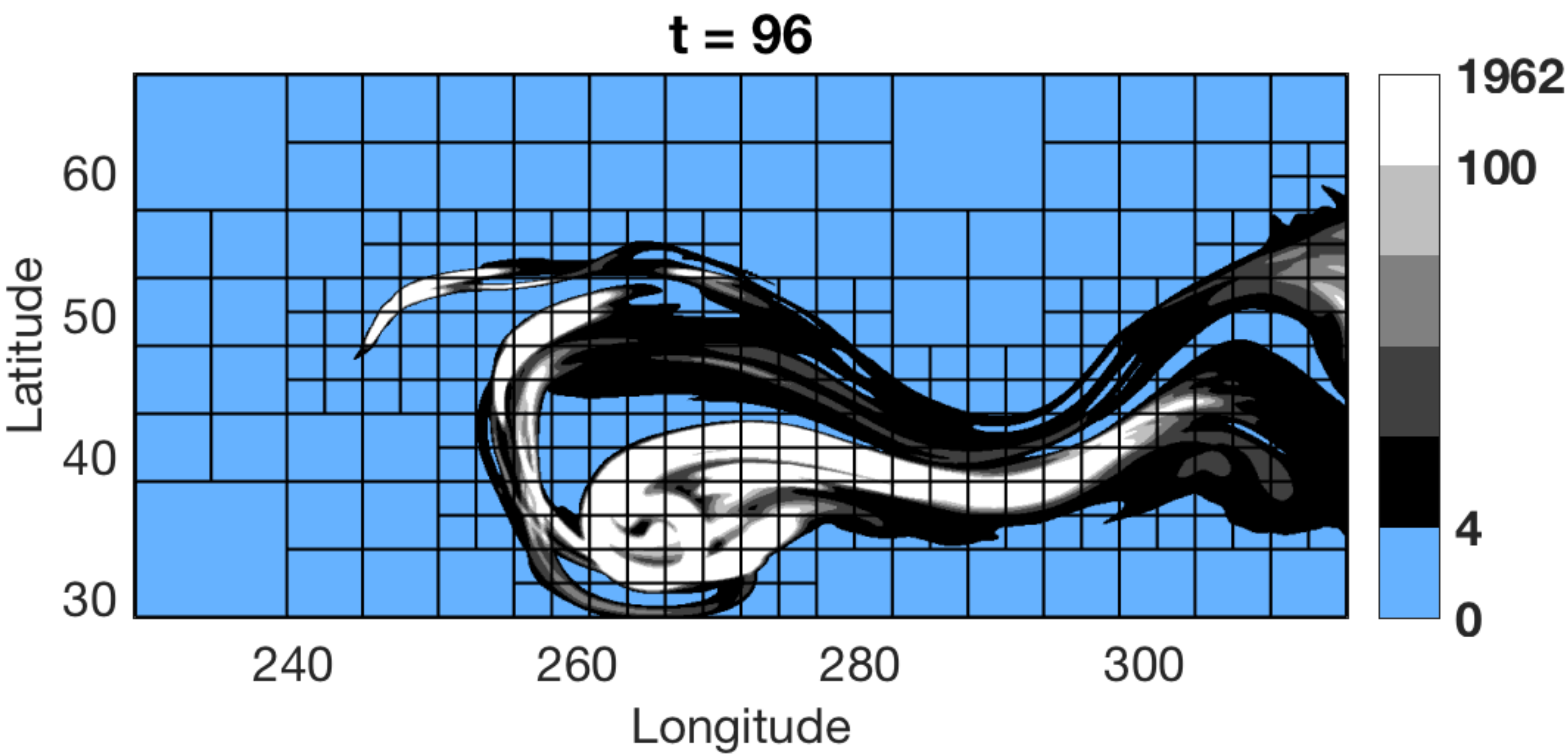
256 x 128 x 25

Adaptive Ash3d



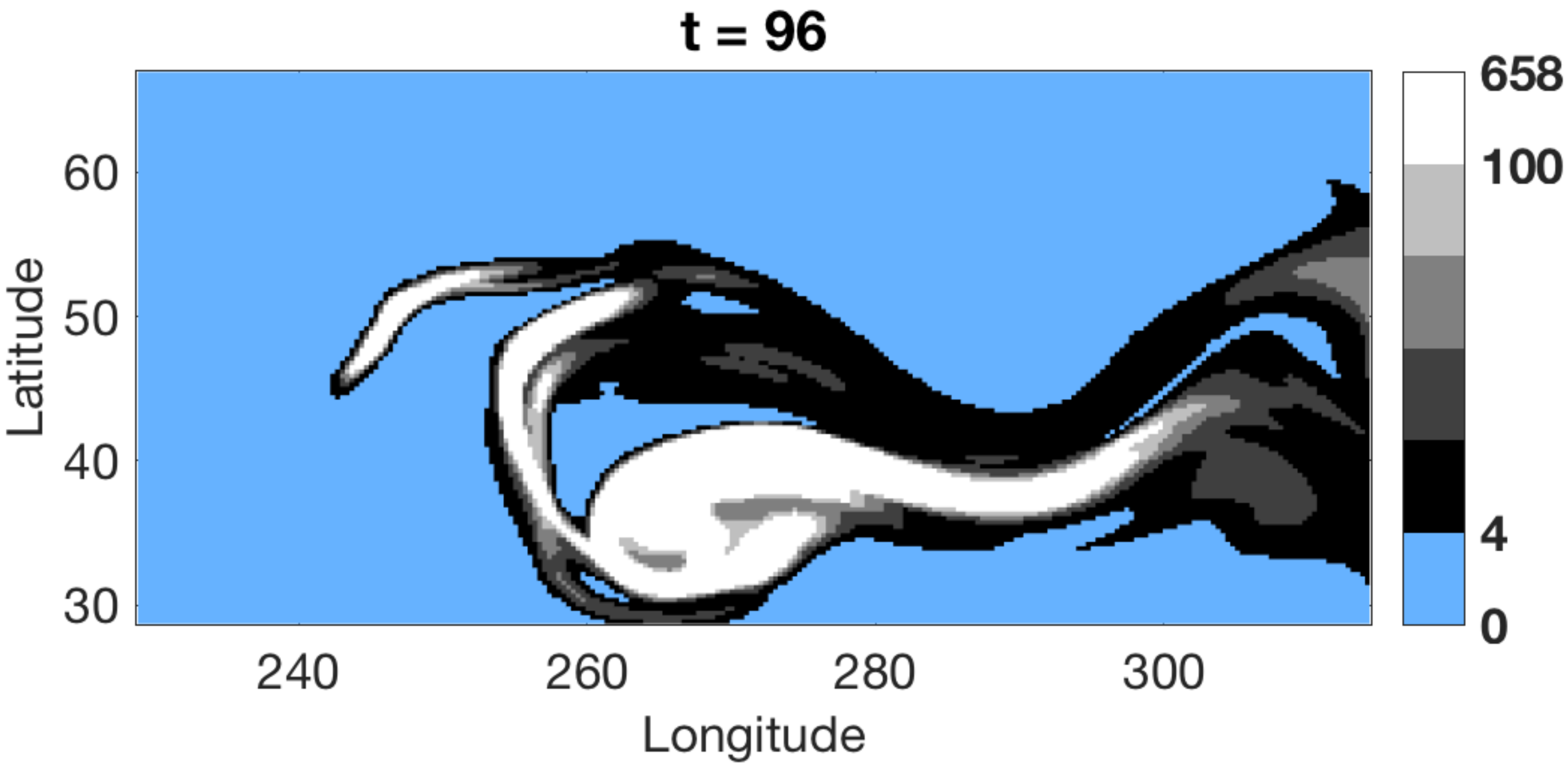
512 x 256 x 25

Adaptive Ash3d



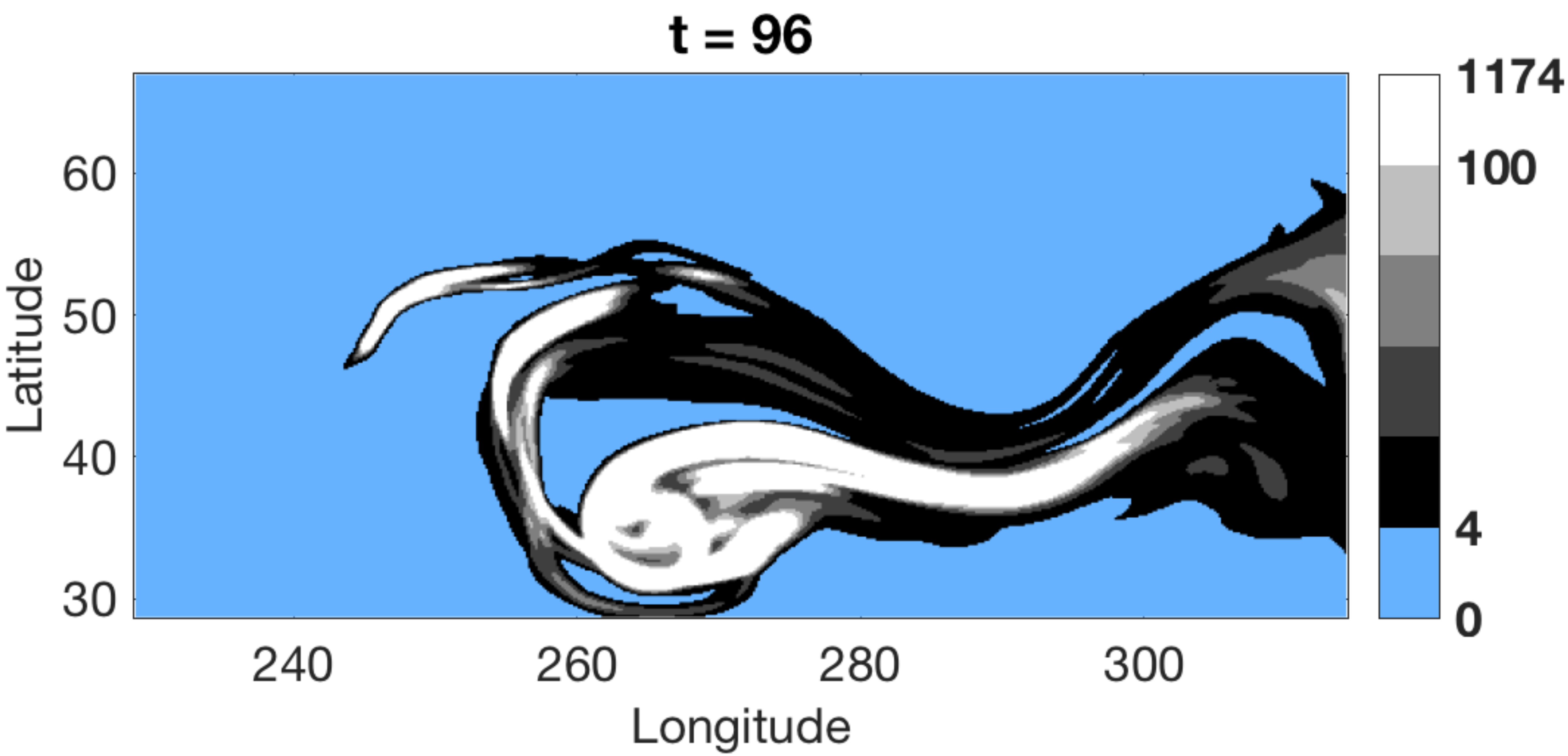
1024 x 512 x 25

Adaptive Ash3d



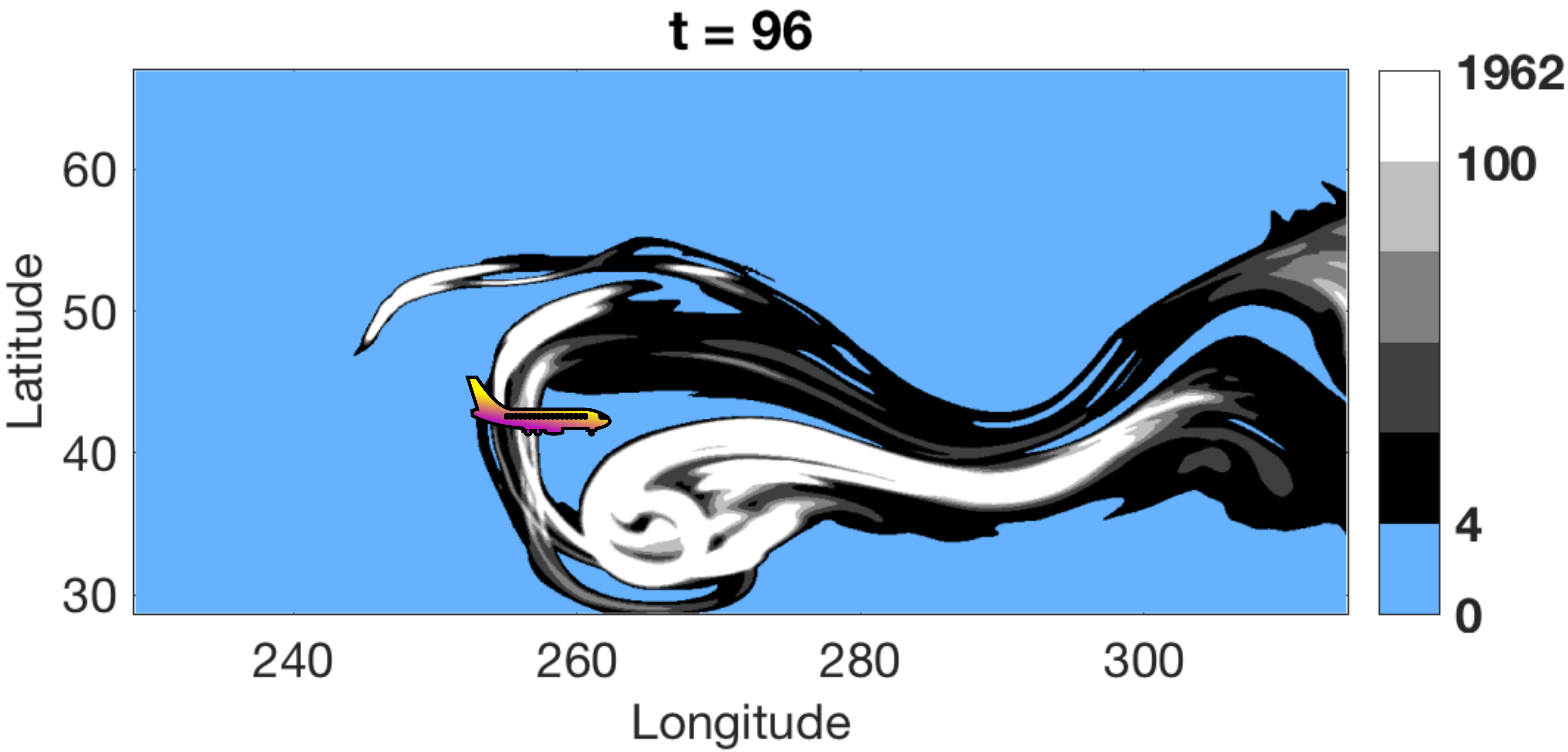
256 x 128 x 25

Adaptive Ash3d



512 x 256 x 25

Adaptive Ash3d

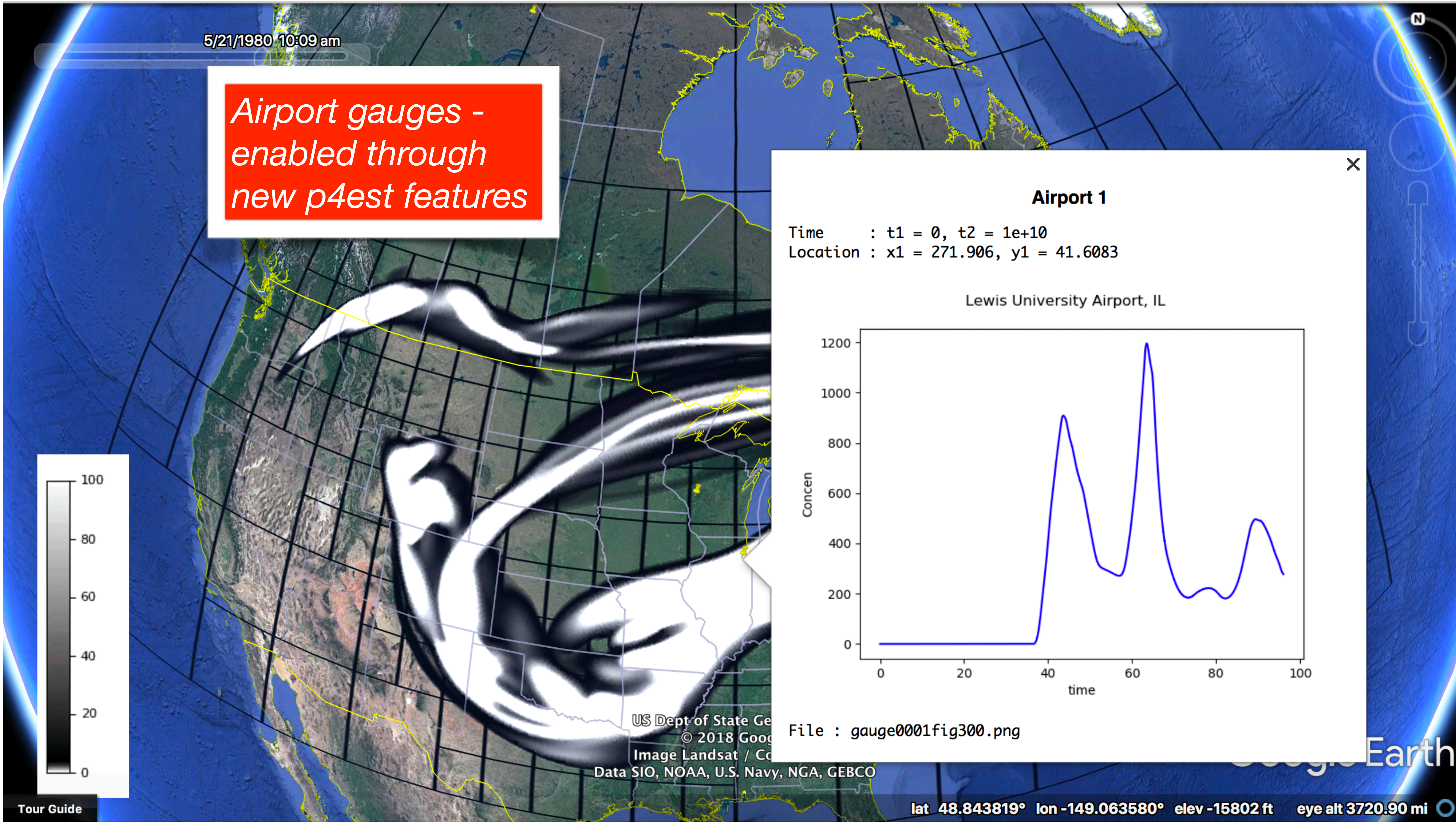


1024 x 512 x 25

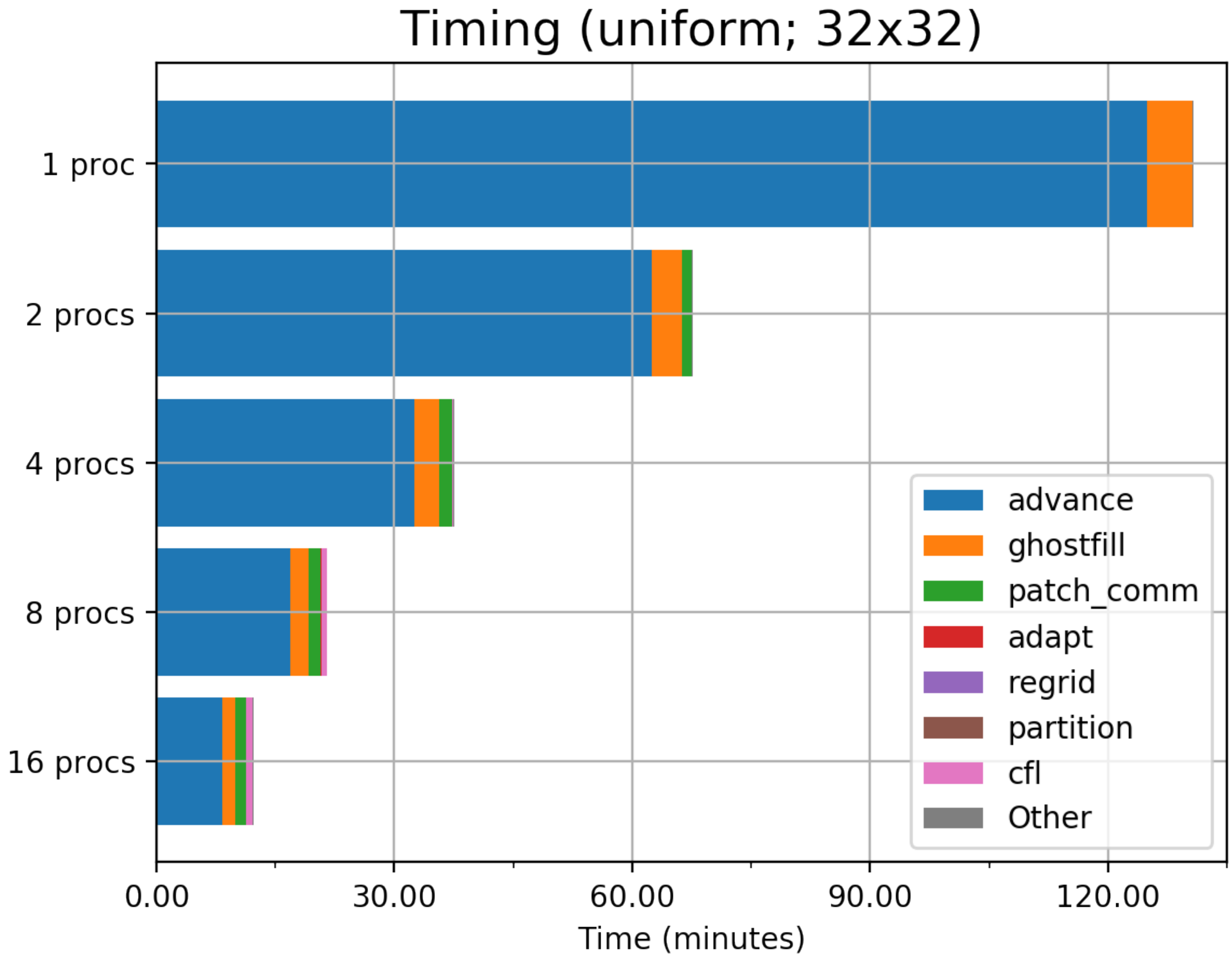
Mt. St. Helens

Google Earth Pro File Edit View Tools Add Window Help
42% Fri 10:33:30 PM calhoun
Google Earth Pro
5/20/1980 5:00 am
Spokane
Ritzville
Yakima
Mount St. Helens
Mount St. Helens' May 18, 1980 Ash Fallout Distributon in U.S.
2 inches to 5 inches
½ inch to 2 inches
Trace to ½ inch
100
80
60
40
20
0
US Dept of State Geographer
© 2018 Google
Image Landsat / Copernicus
Data SIO, NOAA, U.S. Navy, NGA, GEBCO
Google Earth
Tour Guide
eye alt 6000.11 mi

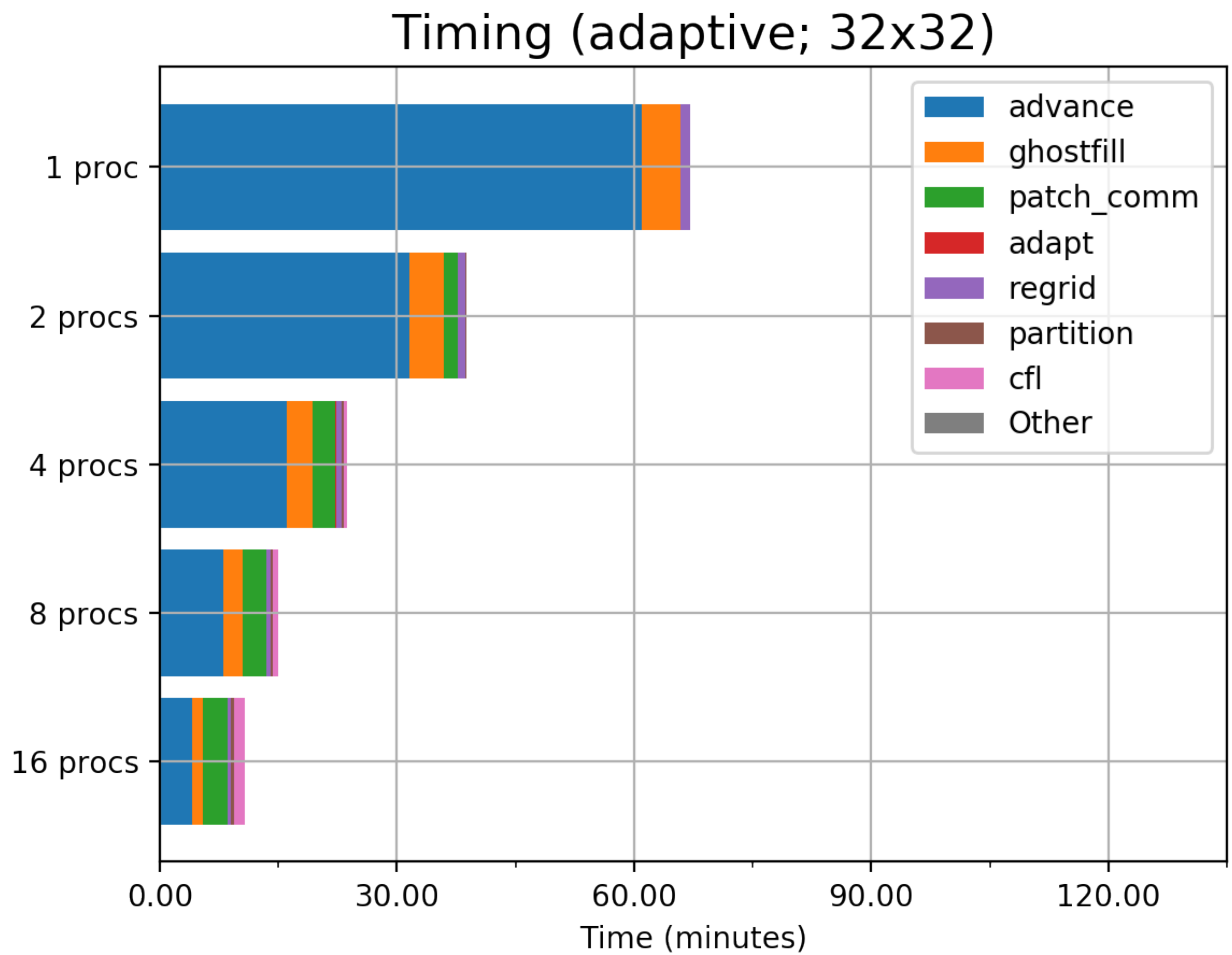
Mt. St. Helens


5/21/1980 10:09 am
Airport gauges - enabled through new p4est features
100
80
60
40
20
0
Tour Guide
Airport 1
Time : t1 = 0, t2 = 1e+10
Location : x1 = 271.906, y1 = 41.6083
Lewis University Airport, IL
Concen
1200
1000
800
600
400
200
0
0
20
40
60
80
100
time
File : gauge0001fig300.png
Data SIO, NOAA, U.S. Navy, NGA, GEBCO
lat 48.843819° lon -149.063580° elev -15802 ft eye alt 3720.90 mi

Parallel results (512x256x25)



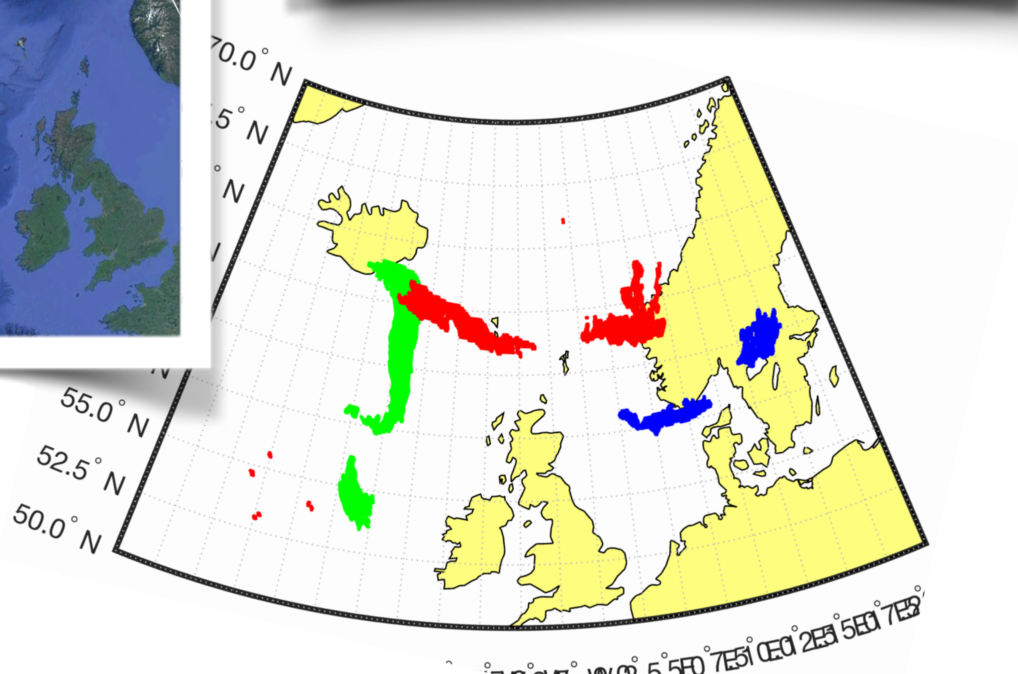
Parallel results (512x256x25)

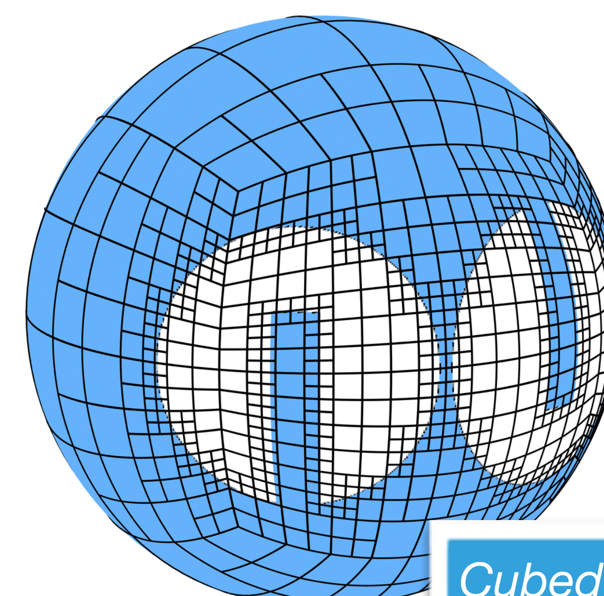


Conclusions

- **Fortran is alive and well**, and widely used. Modern features improve interoperability with other languages
- **Parallel adaptive capabilities** can be added without extensive refactoring
- Interesting **software engineering question** : Can code be set up to look as much like the native legacy code as possible, with an adaptive, parallel option?
- **Encourage thinking** about what faster runs/higher resolution can do for operational forecasting, prediction, development of hazards maps and so on.

Comparisons with satellite data

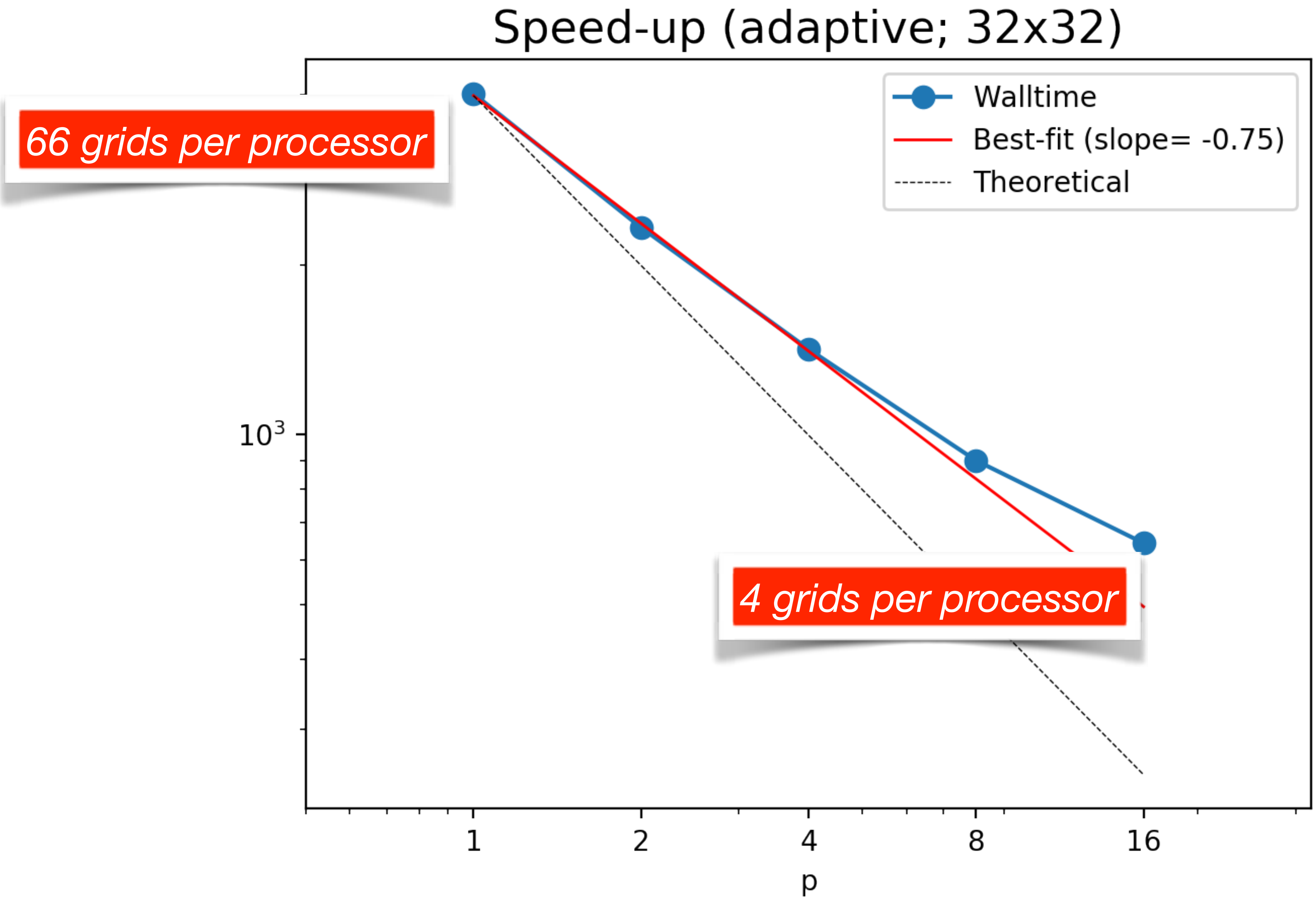




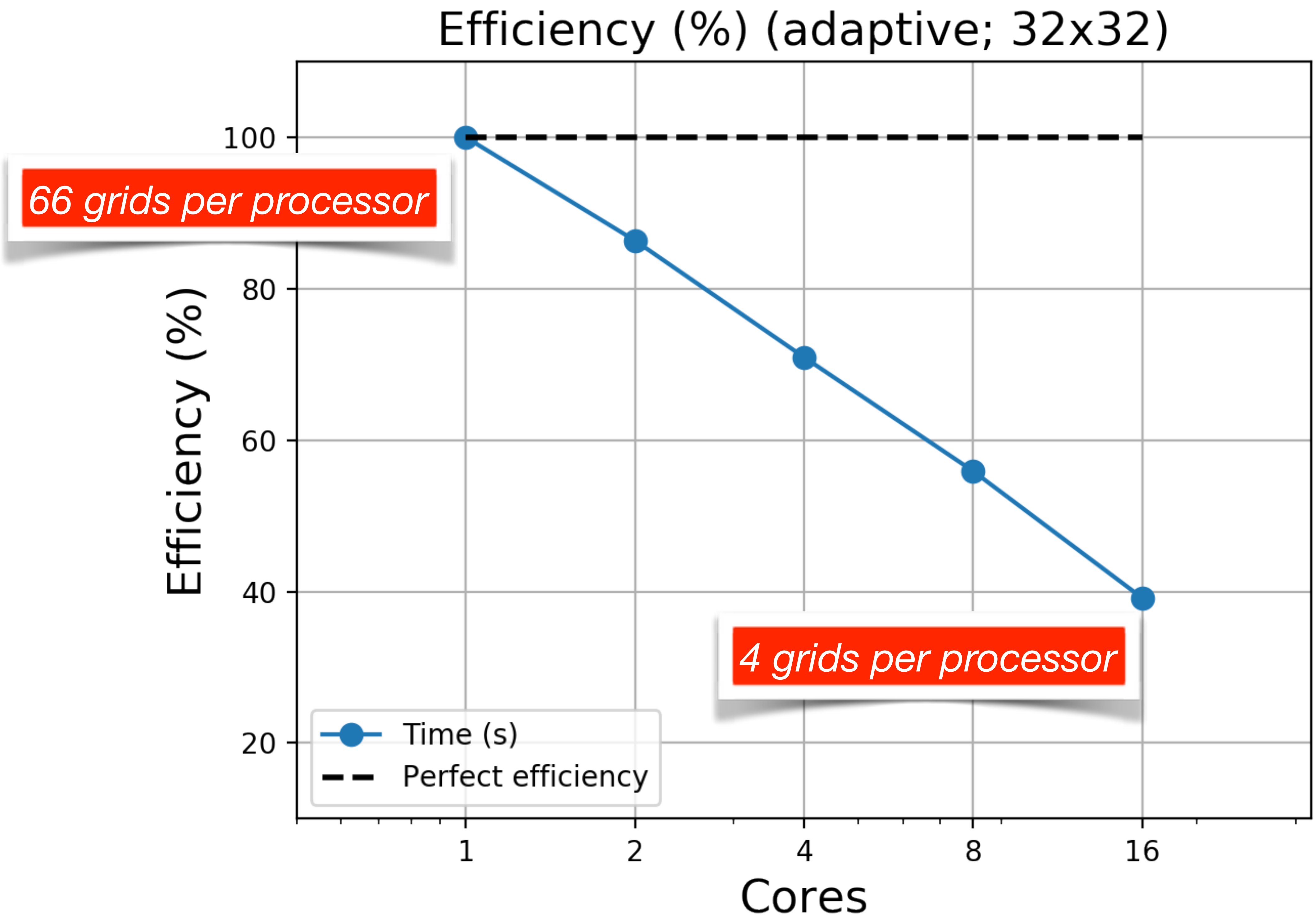
Cubed-sphere grid for global simulations

www.forestclaw.org www.github.com/ForestClaw

Strong scaling



Efficiency


Efficiency (%) (adaptive; 32x32)
100
80
60
40
20
Efficiency (%)
66 grids per processor
4 grids per processor
Time (s)
Perfect efficiency
1
2
4
8
16
Cores