Threads in SORD earthquake dynamic rupture code

Geoffrey P. Ely
Leadership Computing Facility
Argonne National Laboratory

MiraCon
March 6, 2013

Argonne NATIONAL LABORATORY

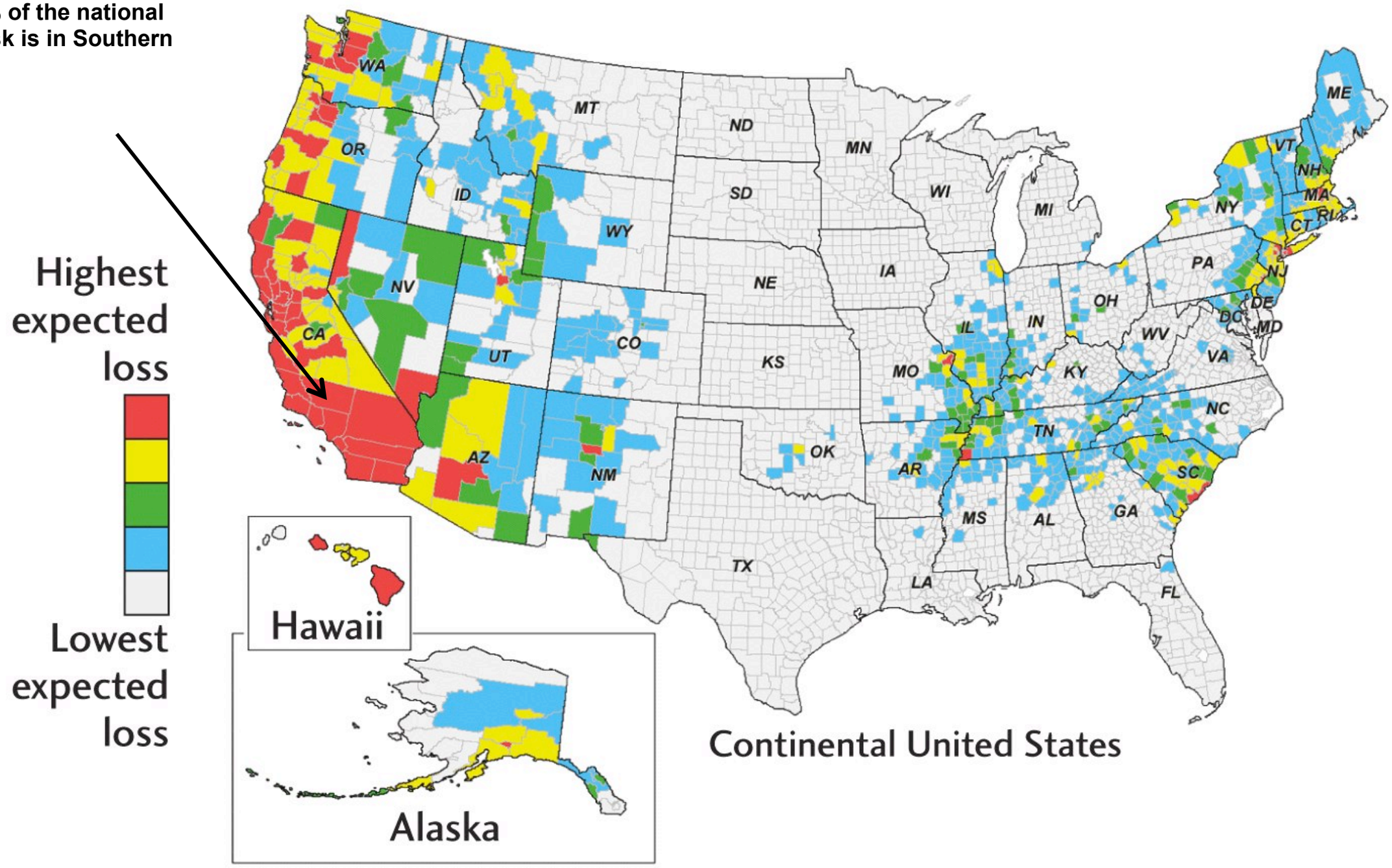
USC

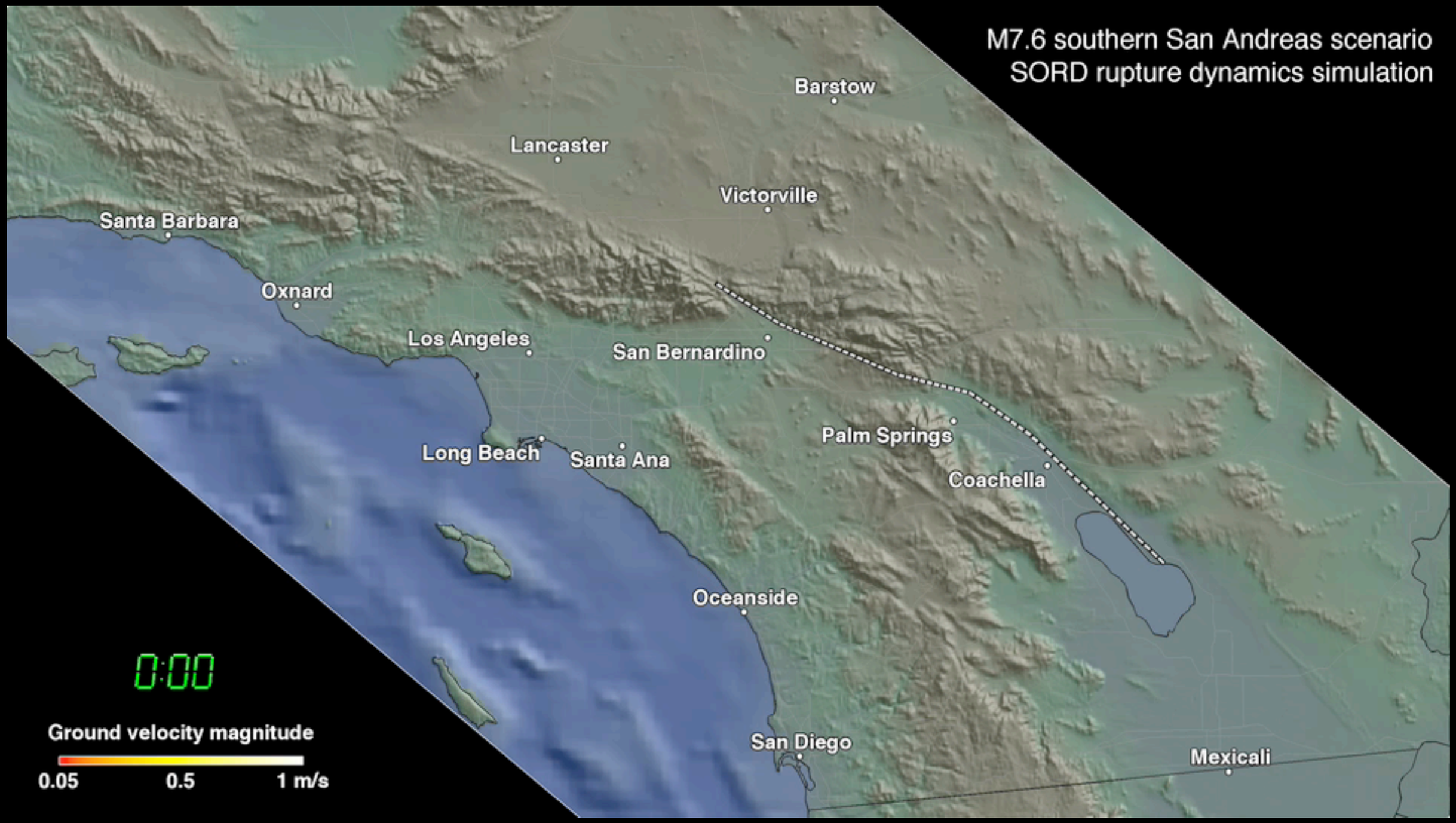
SCEC
an NSF + USGS center

U. S. Seismic Risk Map

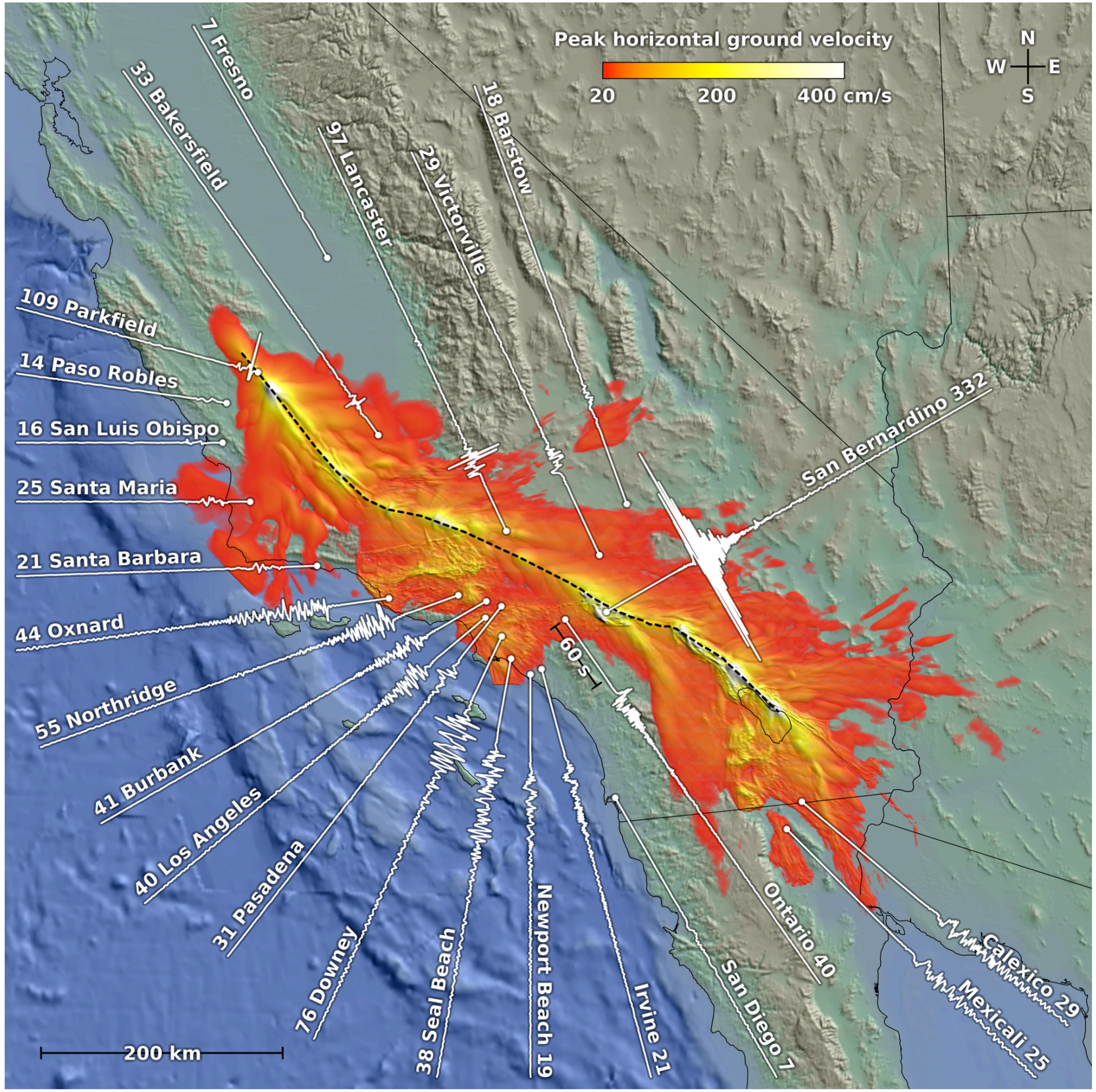
FEMA 336 Report (2000)

About 50% of the national seismic risk is in Southern California




M7.6 southern San Andreas scenario
SORD rupture dynamics simulation
Barstow
Lancaster
Victorville
Santa Barbara
Oxnard
Los Angeles
San Bernardino
Palm Springs
Long Beach
Santa Ana
Coachella
Oceanside
0:00
Ground velocity magnitude
0.05
0.5
1 m/s
San Diego
Mexicali

M8 Southern San Andreas Scenario



Equations of Motion

Viscoelastic Solid (Kelvin-Voigt Model)

$$\ddot{\mathbf{u}} = \frac{1}{\rho}\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}$$

$$\mathbf{g} = \boldsymbol{\nabla}(\mathbf{u} + \gamma\dot{\mathbf{u}})$$

$$\boldsymbol{\sigma} = \lambda\,\mathrm{trace}(\mathbf{g}) + \mu(\mathbf{g} + \mathbf{g}^{\mathrm{T}})$$

ρ	density
λ, μ	elastic moduli
γ	viscosity
$\mathbf{u}$	displacement
$\boldsymbol{\sigma}$	stress tensor

Free surface boundary condition

$$\boldsymbol{\tau} = \boldsymbol{\sigma}\cdot\hat{\mathbf{n}} = 0$$

Spontaneous Rupture Boundary Condition

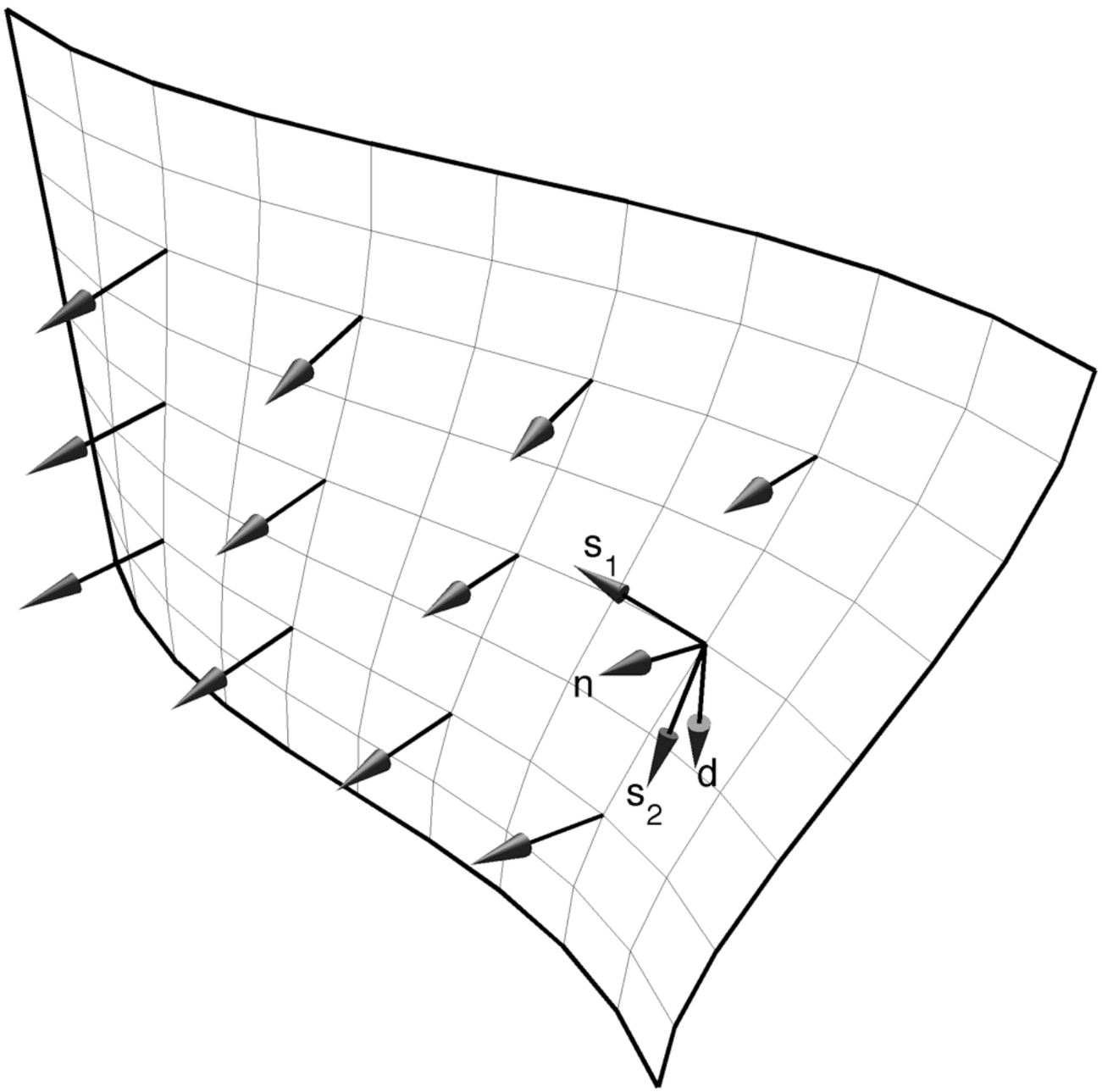
$$\mathbf{s} = \mathbf{u}^+ - \mathbf{u}^-$$

$$\boldsymbol{\tau}_n = \hat{\mathbf{n}} \cdot \boldsymbol{\sigma} \cdot \hat{\mathbf{n}}$$

$$\boldsymbol{\tau}_s = (\mathbf{I} - \hat{\mathbf{n}}\hat{\mathbf{n}}) \cdot \boldsymbol{\sigma} \cdot \hat{\mathbf{n}}$$

$$|\boldsymbol{\tau}_s| \leq \tau_c$$

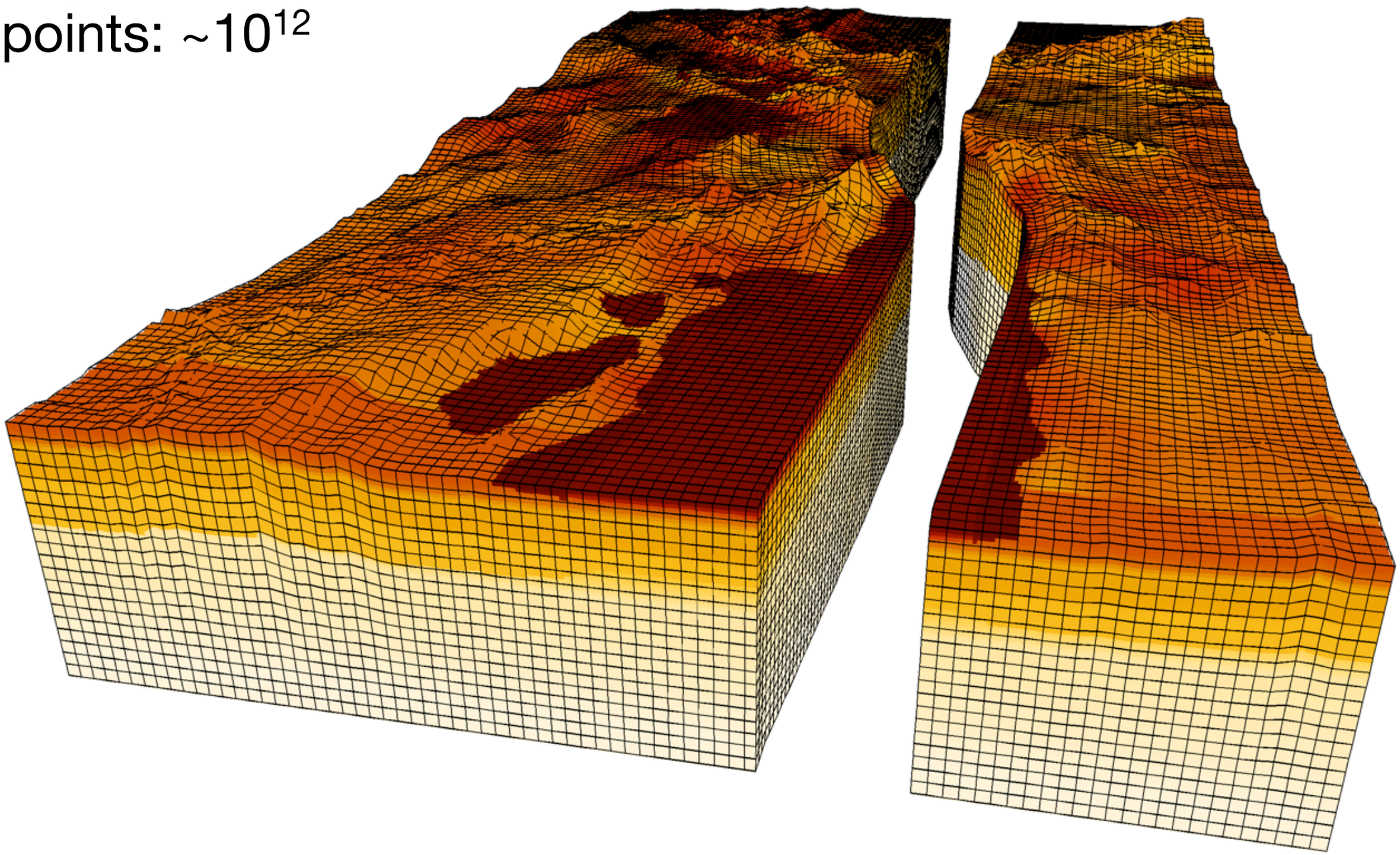
$$\tau_c \dot{\mathbf{s}} = |\dot{\mathbf{s}}| \boldsymbol{\tau}_s$$



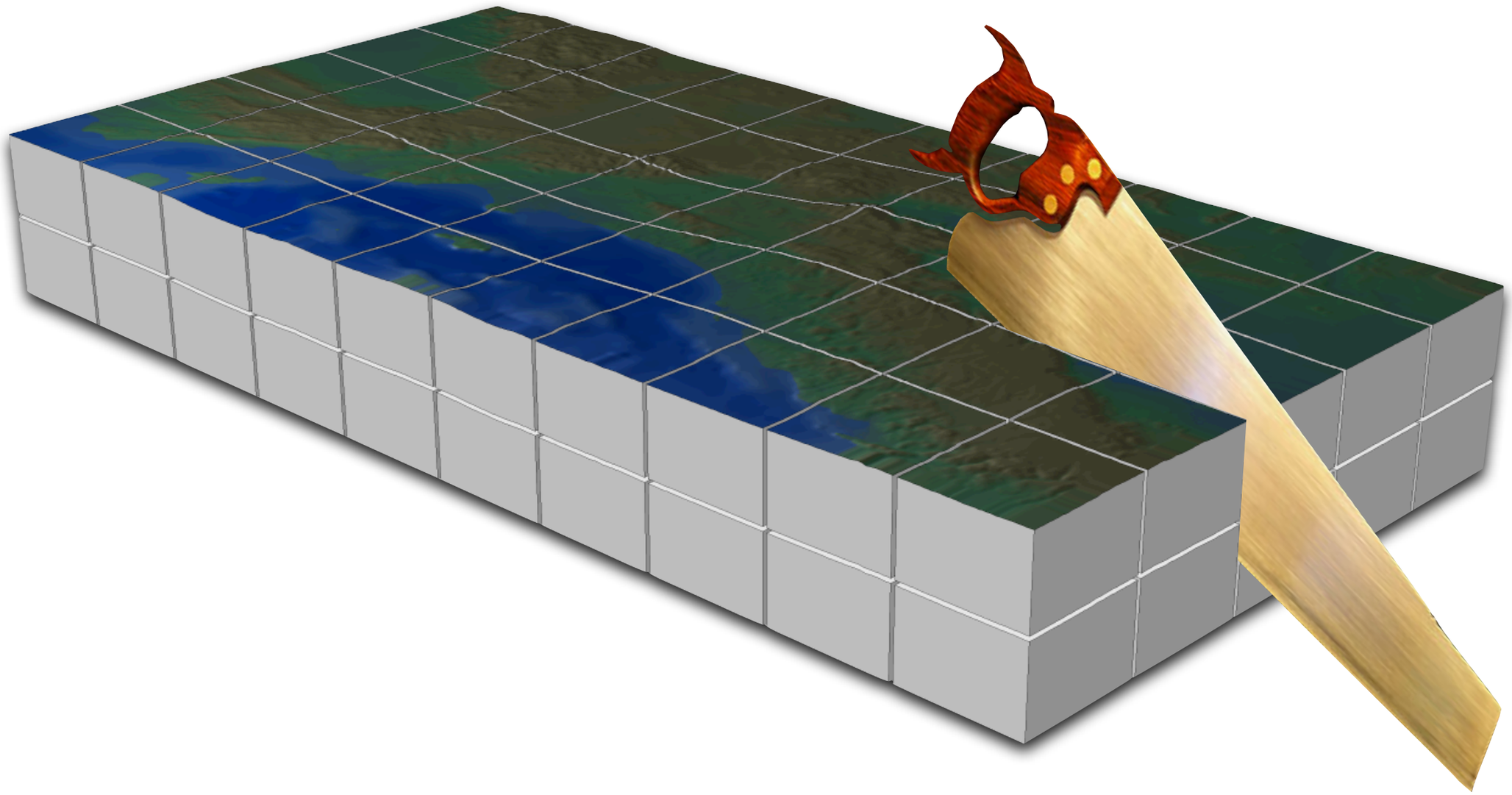
$\mathbf{s}$	slip
$\boldsymbol{\sigma}$	stress tensor
τ_n	normal traction
$\boldsymbol{\tau}_s$	shear traction
τ_c	frictional strength
$\hat{\mathbf{n}}$	surface unit normal

Simulation Scale

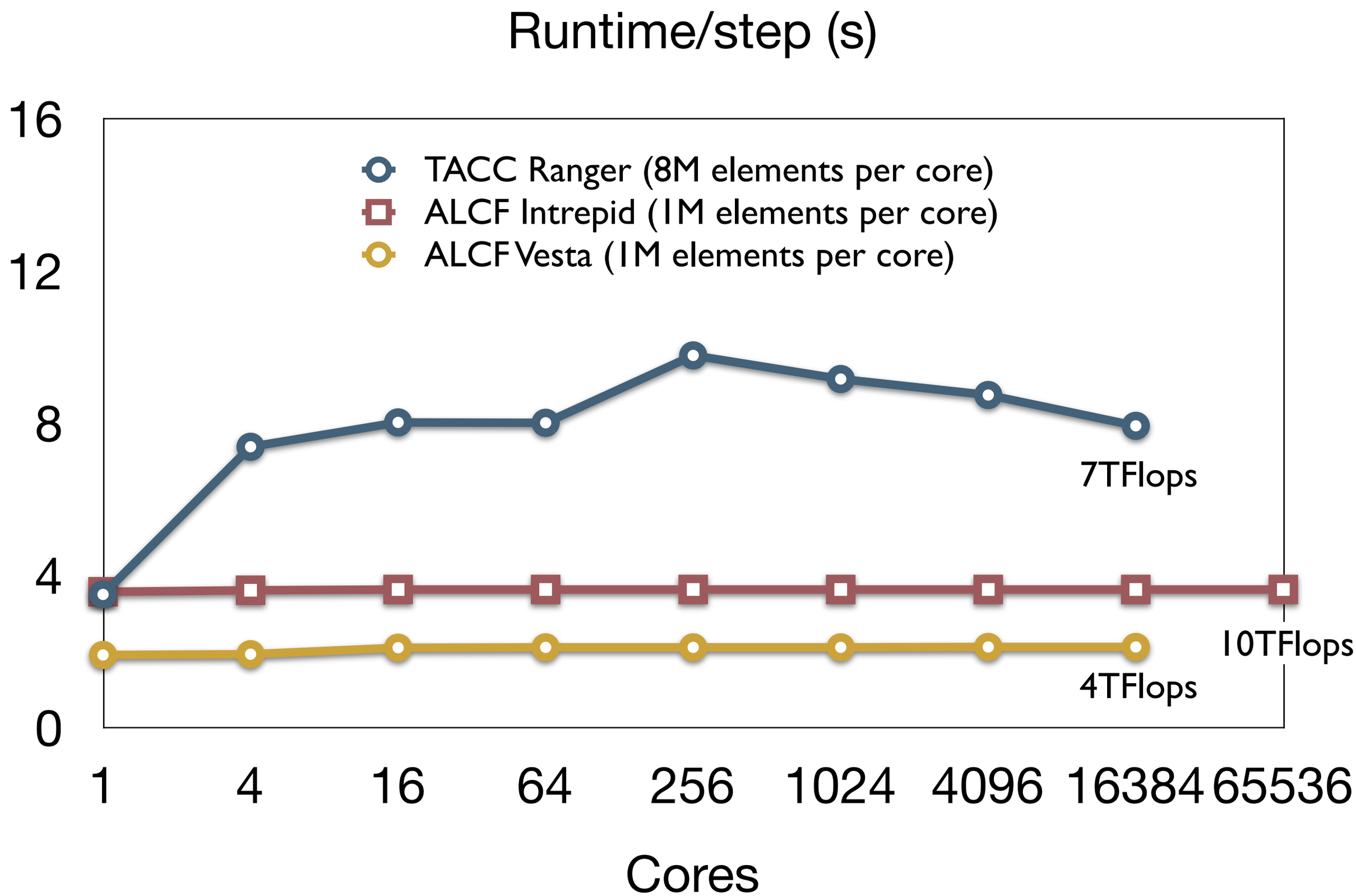
Outer length scale: ~500km
Inner length scale: ~50m
Mesh points: ~10^{12}



Domain Decomposition



Weak Scaling Benchmark



Hardware Counters with Walkup's mpi_trace Library

Pure MPI link flags:

```
-lSPI_upci_cnk
/home/morozov/HPM/lib/libmpihpm.a
/bgsys/drivers/ppcfloor/bgpm/lib/libbgpm.a
```

MPI + OpenMP link flags:

```
-lSPI_upci_cnk
/home/morozov/HPM/lib/libmpihpm_smp.a
/bgsys/drivers/ppcfloor/bgpm/lib/libbgpm.a
```

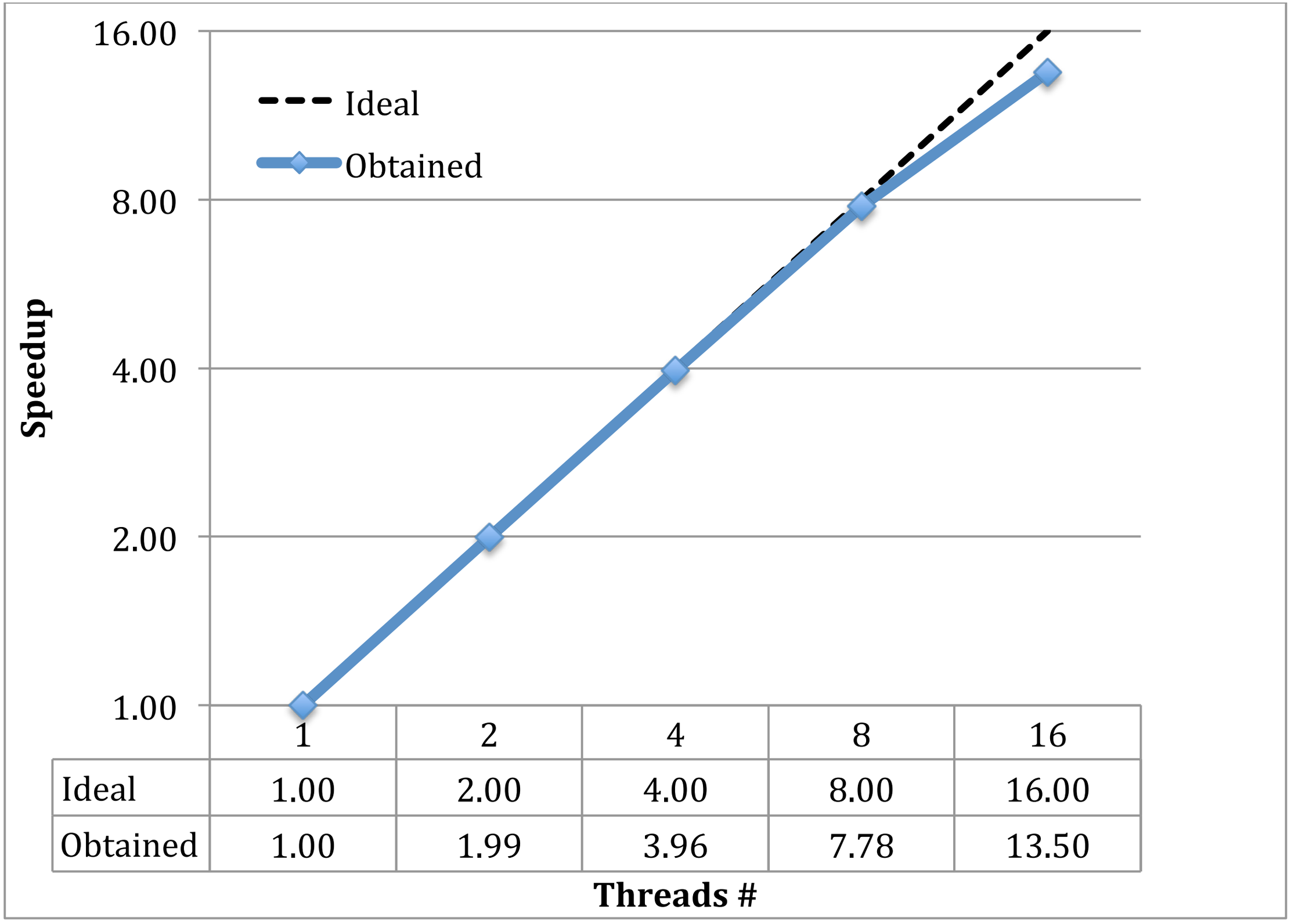
mpi_trace Output

```
=================================================================
Aggregate BG/Q counter data for jobid 10358
This report includes all 16384 processes in MPI_COMM_WORLD and a total of 16384 cores
=================================================================
-----------------------------------------------------------------
mpiAll, call count = 1, avg cycles = 60045259469, max cycles = 60360564972 :
  -- Counter values for processes in this reporting group ----
    min-value  min-rank    max-value  max-rank    avg-value   label
  4.580812e+08     127   4.609117e+08    8127   4.607698e+08  Committed Load Misses
  1.020854e+10   16256   1.034341e+10    8433   1.025143e+10  Committed Cacheable Loads
  1.033073e+07   13558   1.316181e+07   12278   1.118504e+07  L1p miss
  1.586691e+10     644   1.631809e+10    8127   1.595532e+10  All XU Instruction Completions
  6.971373e+09   16383   7.022227e+09    8127   7.021428e+09  All AXU Instruction Completions
  9.605689e+09   16383   9.676877e+09    8127   9.675758e+09  FP Operations Group 1

  Histogram of floating-point operation counts:
      flop-bin      #ranks
    9.605689e+09         4
    9.636198e+09         0
    9.641283e+09       504
    9.646368e+09         0
    9.671792e+09         0
    9.676877e+09     15876

  Derived metrics for code block "mpiAll" averaged over process(es) in the reporting group
  Instruction mix:  FPU = 30.56 %,  FXU = 69.44 %
  Instructions per cycle completed per core = 0.3807
  Per cent of max issue rate per core = 38.07 %
  Total weighted TFlops = 4.202
```

OpenMP Threading

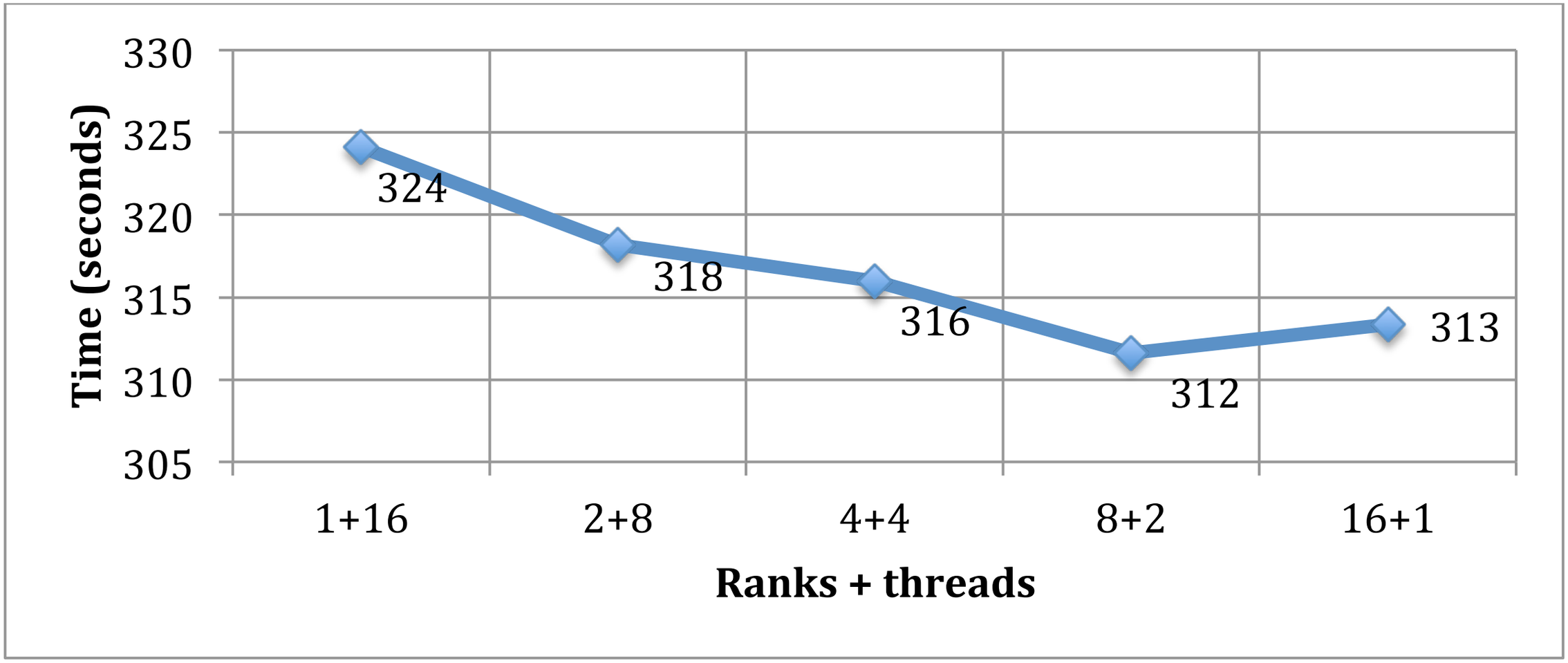
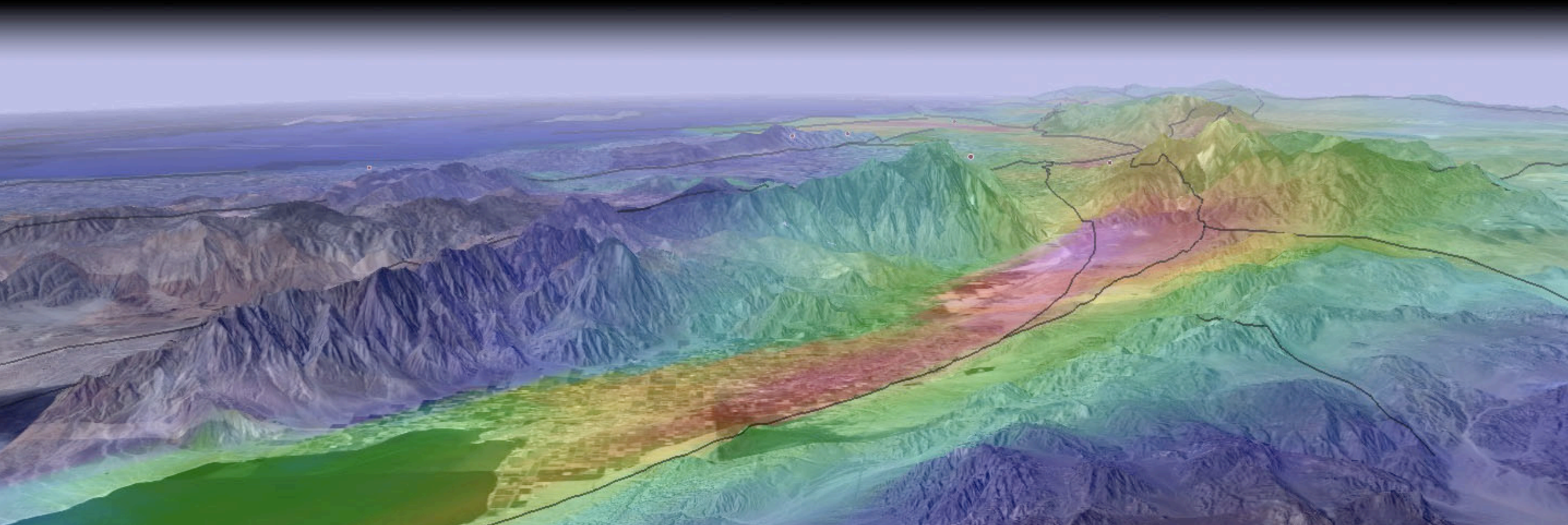
```
!$omp parallel do schedule(static) private(j, k, l)
do l = i1(3), i2(3)
do k = i1(2), i2(2)
do j = i1(1), i2(1)
    df(j,k,l) &
    = bb(j,k,l,5,a) * f(j,  k,l,i) &
    + bb(j,k,l,2,a) * f(j+1,k,l,i) &
    + bb(j,k,l,3,a) * f(j,  k+1,l,i) &
    + bb(j,k,l,8,a) * f(j+1,k+1,l,i)
end do
do j = i1(1), i2(1)
    df(j,k,l) = df(j,k,l) &
    + bb(j,k,l,4,a) * f(j,  k,l+1,i) &
    + bb(j,k,l,7,a) * f(j+1,k,l+1,i) &
    + bb(j,k,l,6,a) * f(j,  k+1,l+1,i) &
    + bb(j,k,l,1,a) * f(j+1,k+1,l+1,i)
end do
end do
end do
```

Strong Scaling



	1	2	4	8	16
Ideal	1.00	2.00	4.00	8.00	16.00
Obtained	1.00	1.99	3.96	7.78	13.50

Threads vs Ranks



The End