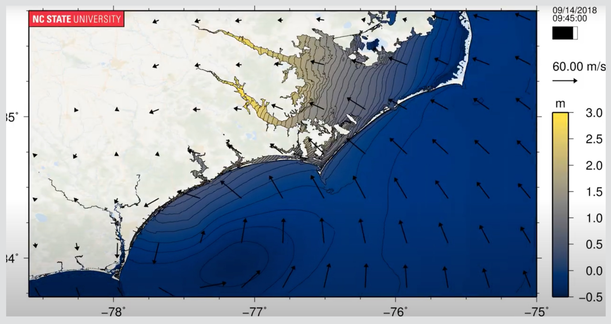
Scalability of Widely Used Coastal Flooding Model

By Sydney Crisanti[1], J. C. Dietrich[1]

1. Department of Civil, Construction, Environmental Engineering, North Carolina State University

1. Introduction

- Coastal storms can push water into inland regions, flooding coastal communities
- Numerical models are used to predict flooding due to these storms in real time
    - The Advanced Circulation (ADCIRC) model is used to predict flooding of coastal areas during extreme weather
- Must be fast and efficient so emergency personnel can better prepare and react
- Utilization of large computing cluster
- Massive computation must be completed within 1-2 hours

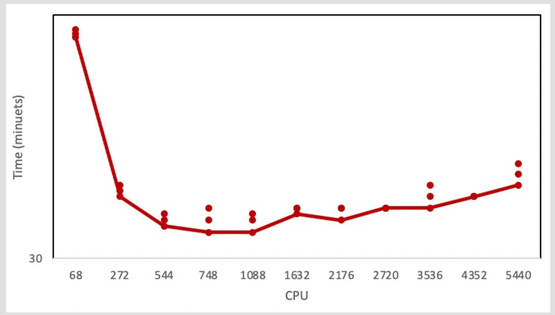


ADCIRC- North Carolina Coast during Hurricane Florence

3. Results

Low-Resolution

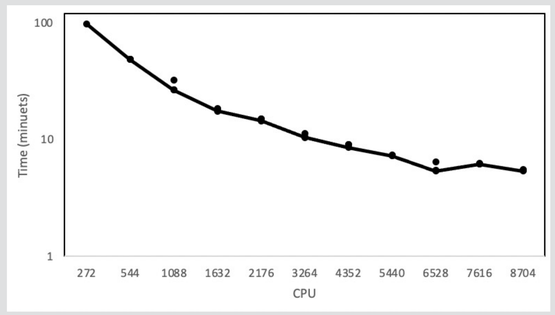
Figure 1



Low-Resolution EC2001 model presents strong scaling from 68 cores to 544 cores

High-Resolution

Figure 2



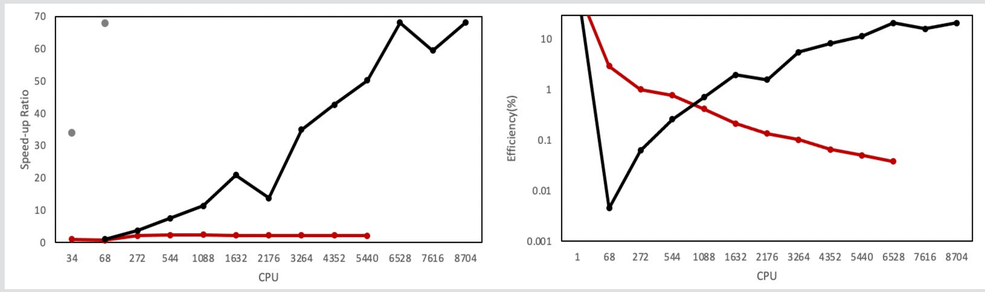
High-Resolution SACS model presents strong scaling from 272 cores to 6528 cores

Overall

- Both high- and low-resolution model break from ideal 1:1 ratio of the speed-up ratio almost immediately
- High-resolution model has higher efficiency than low-resolution
    - At 1 core, theoretically 100% efficiency due to parallelization
- Both break strong scaling at about 600 points per core
- High-resolution model is faster after 500 cores
- Low-resolution model is faster before 500 cores

Figure 3

Figure 4



- Red line is the low-resolution model, Black line is the high-resolution model
- Figure 3 represents speed-up ratio with an optimal 1:1 line in grey circles
    - $S_p=T_1/T_p$
        - Where T_1 is the timing for which the model ran at one node, Tp is the timing for which the model ran at p cores
- Figure 4 portrays efficiency,
    - $E_p=S_p/p$
        - Where p is the number of cores

2. Methods

- Ran Simulations on high- and low-resolution models
    - High resolution has high amounts of points increasing the accuracy and detail of the model prediction
- Building Scaling Models
    - Monitor the speed of the code over various central processing units or cores
- Utilization of High-Performance Computing resources
    - 25th fastest Supercomputer Stampede2 via Texas Advanced Computing Center
- Measuring Efficiency
    - Scaling curves
    - Speed-up ratio
    - Efficiency ratio

4.Conclusions

- High-resolution model is 20 times more efficient
- High-resolution model is most efficient at 6528 cores
- Low-resolution model is most efficient at 544 cores
- Independent of resolution strong scaling breaks at 600 points per core

Future Work

- Why does low-resolution model break linear scaling after 544 cores?
- How do different compilers impact scaling?