Scaling Parallel Epidemic Simulations using Charm++

Abhinav Bhatele, Jae-Seung Yeom, Nikhil Jain, Chris J. Kuhlman, Yarden Livnat, Keith R. Bisset, Laxmikant V. Kale, Madhav V. Marathe

SC17 Denver, CO | hpc connects.

SC '17 ◆ November 16, 2017

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

LLNL-PRES-742360

Approach: individual-based simulation

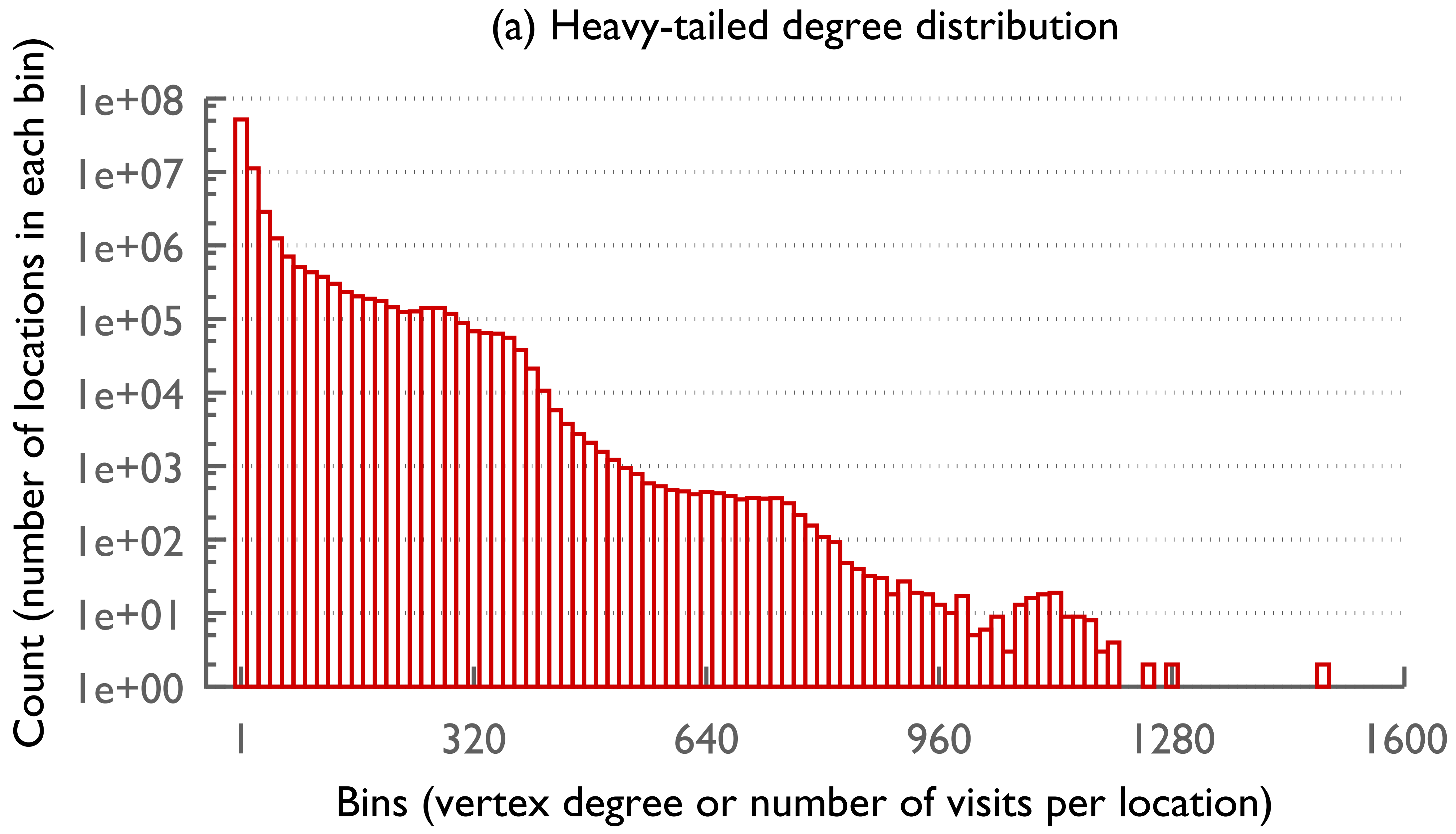
- Agent-based modeling to simulate epidemic diffusion
- Models agents (people) and interactions between them
- People interact when they visit the same location at the same time
- These "interactions" between pairs of people are represented as "visits" to locations
- A bi-partite graph of people and locations is used

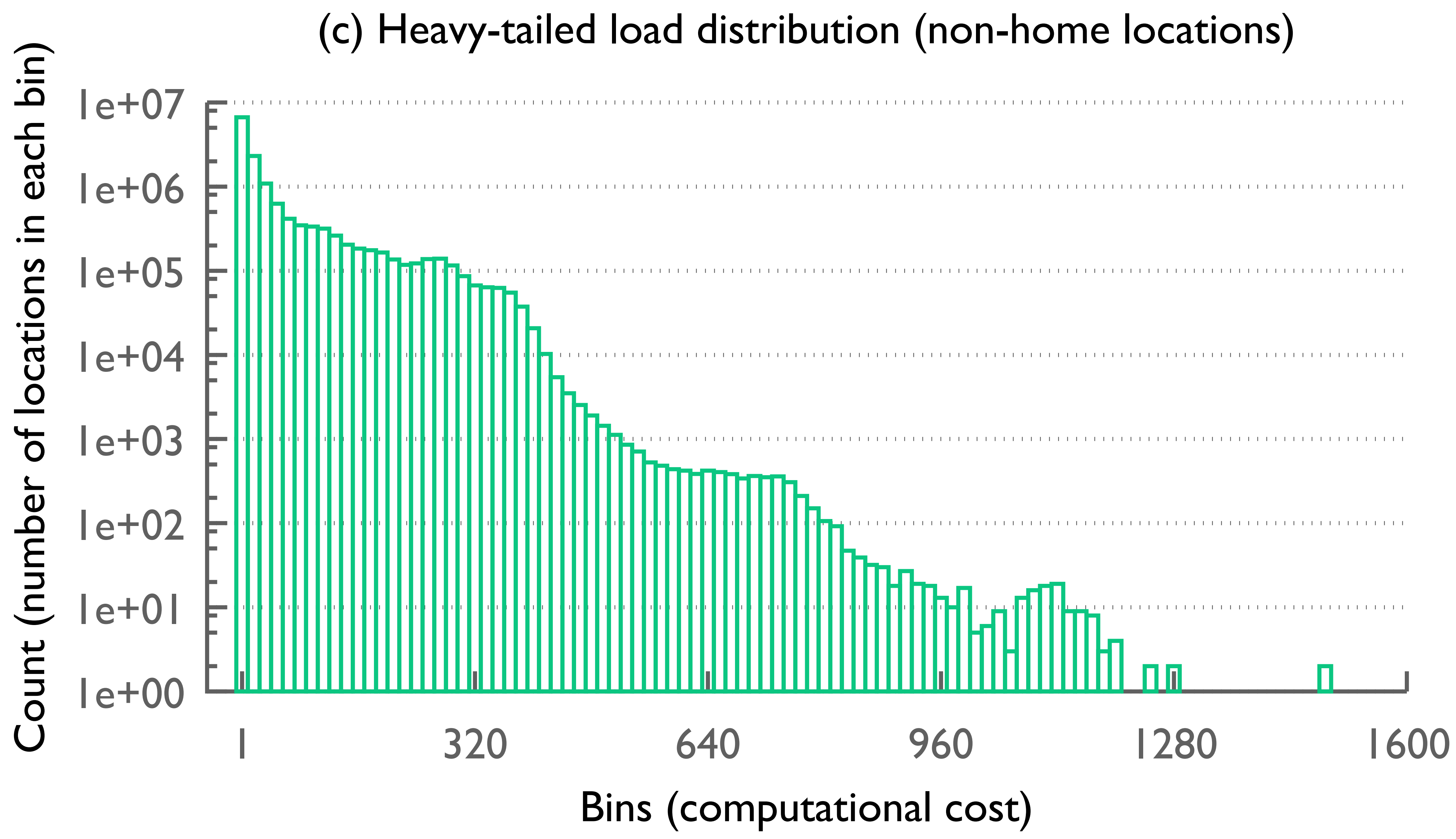
EpiSimdemics: Parallel implementation

- All the people and locations are distributed among all processes
- Computation can be done locally in parallel
- Communication when sending visit and infection messages
- Uses Charm++, a message-driven model

```
while d ≤ d_max do
    for p ∈ P do
        Evaluate scenario trigger conditions;
        Update health state h_p, if necessary, and reevaluate triggers;
        foreach v ∈ V_p ( visit schedule of p) do
            Send visit message m to location l;
        end
    end
    for l ∈ L do
        foreach m destined for l do
            Determine the sublocation l_s to visit;
            Create an arrival and departure event for each visit;
            Put the events into the event queue q_e of l;
        end
        Reorder q_e by the time of event in ascending order;
        foreach e ∈ q_e do
            if e is arrival then
                Put p into sublocation l_s;
            else
                Remove p from sublocation l_s;
                foreach p' currently in l_s do
                    Compute disease transmission probability q
                      between p' and p;
                    if q > threshold then
                        Send infection message to the infected
                          person (p or p');
                    end
                end
            end
        end
    end
    d++;
end
```

Performance optimizations made challenging by heavy-tailed distributions

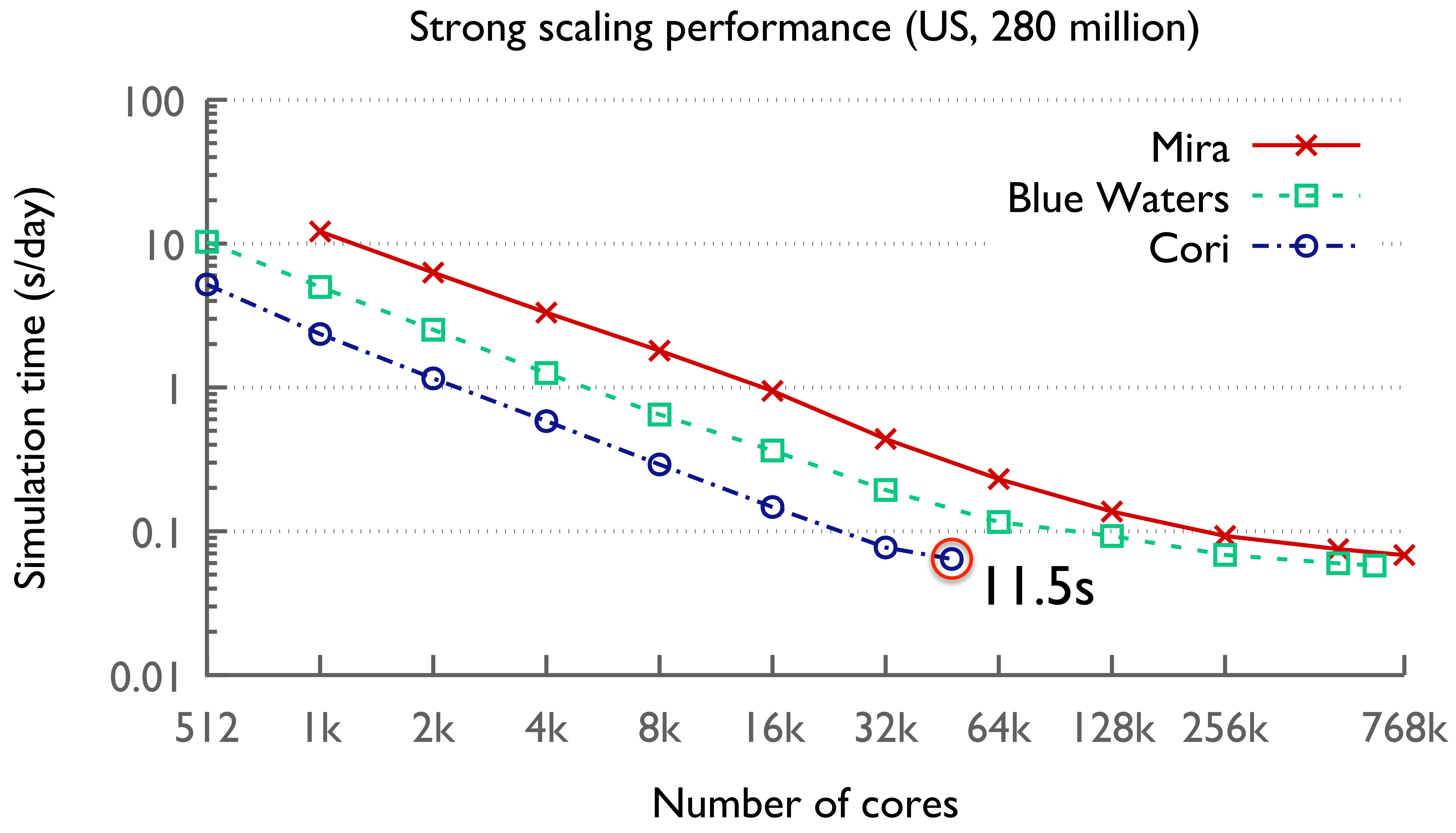


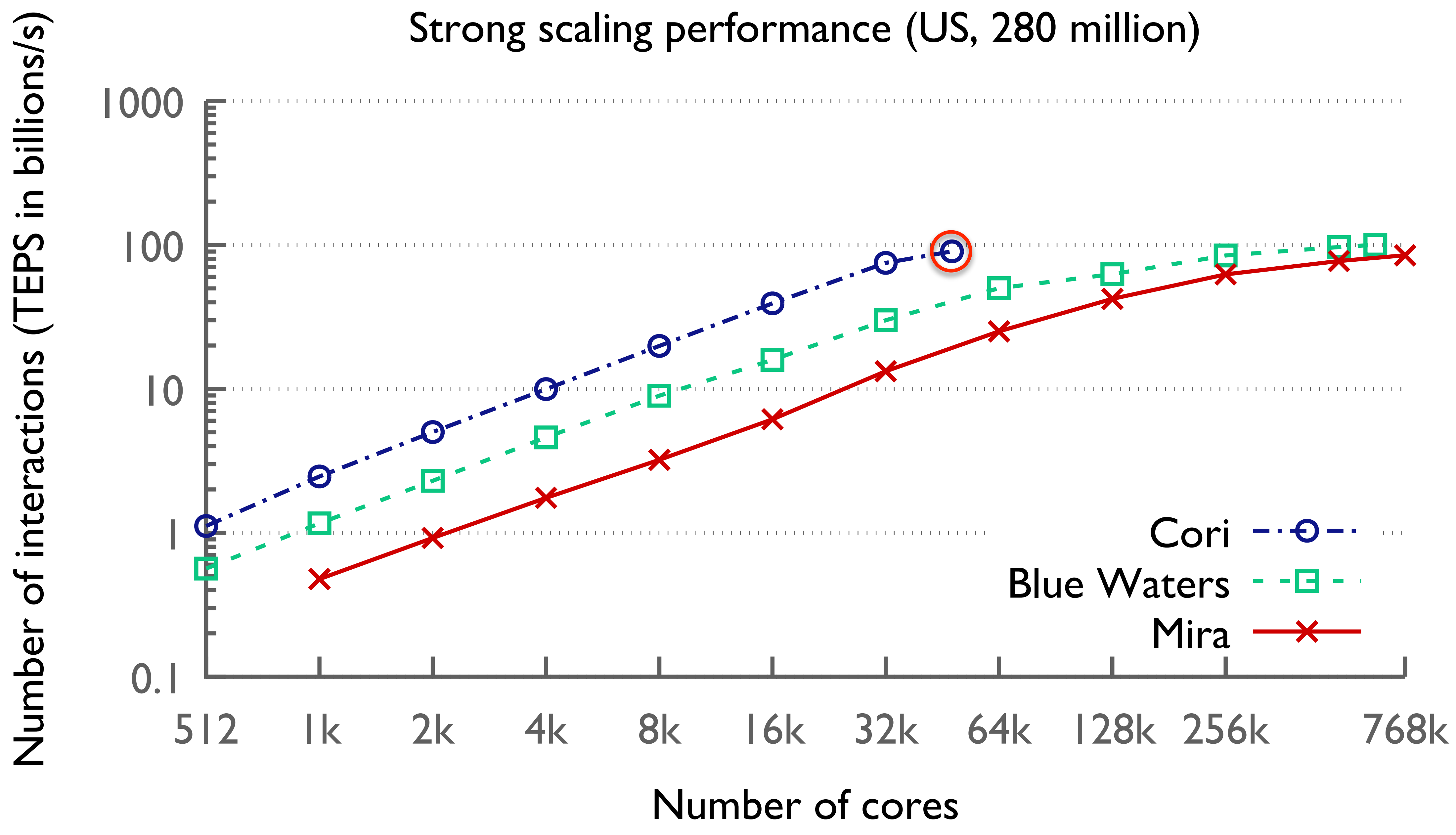


Optimizing communication

- The first phase of the simulation sends many small “visit” messages
- Use Charm++’s TRAM library for message aggregation
- Also alternate the creation and sending of batches of visit messages

Strong scaling performance: US dataset





Impact of the solution

- During an epidemic outbreak, planning and government response require simulation turn around times of <24 hours
- The efficiency and extreme scaling of Episimdemics makes this possible