USC Viterbi School of Engineering

Interactive, Real-Time, Distributed, Heterogeneous, Agent-based Computing

Information Sciences Institute, Computational Sciences Division

Information Sciences Institute

USC

29 November 2007
Bob Lucas
rflucas@isi.edu

Parallel Universes

Big Science	Military Experimentation
Big SMP Cluster	Dispersed Clusters & PCs
Batch mode	Interactive
Best Effort	Soft Real Time
SPMD	Heterogeneous
MPI	RTI

Urban Resolve Experiment

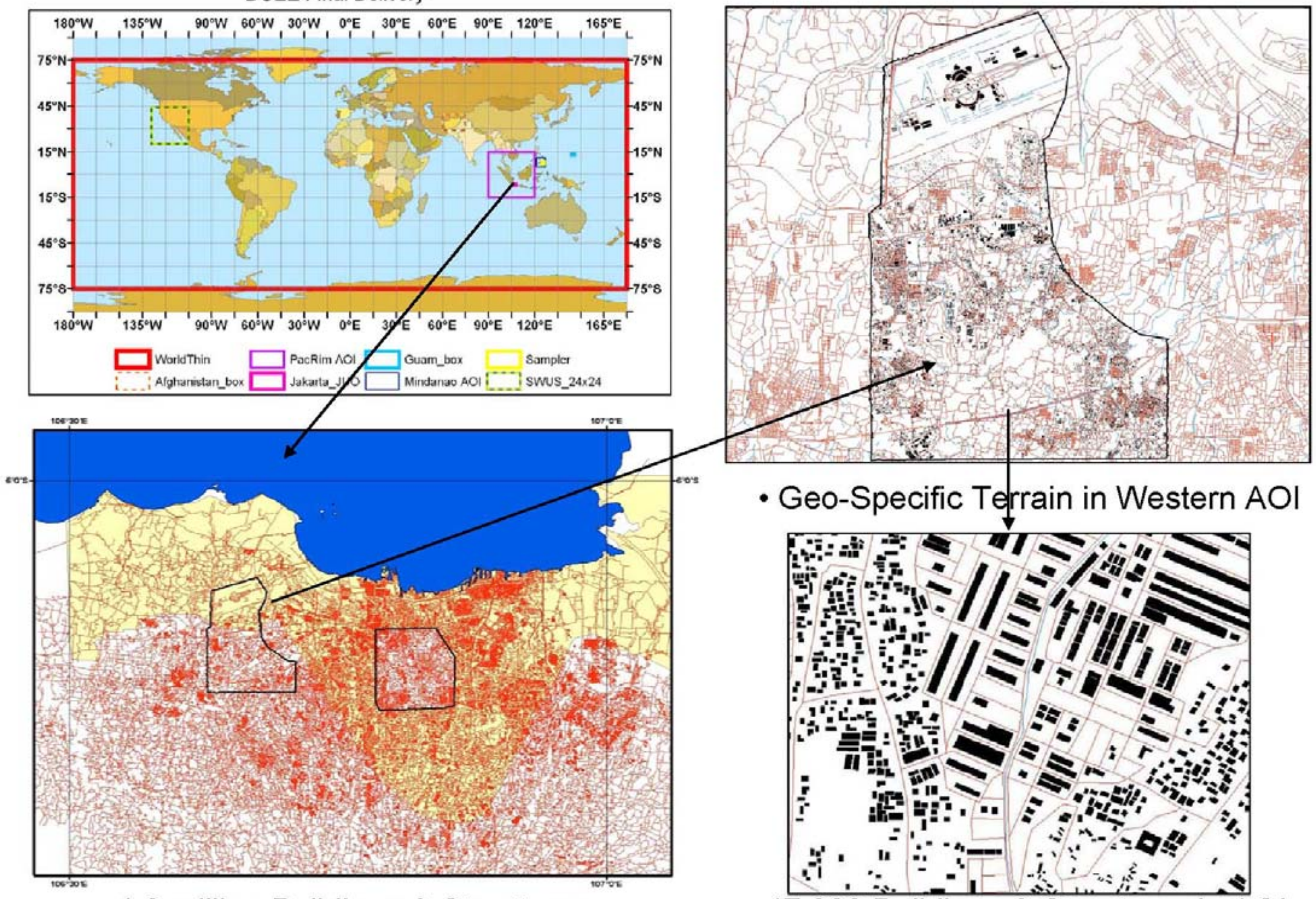
JFCOM Experimentation Directorate

Joint Urban Operations Experiments

Global Terrain

Simulate World-Wide Terrain

DCEE Final Delivery



• Geo-Specific Terrain in Western AOI

•1.8 million Buildings & Structures

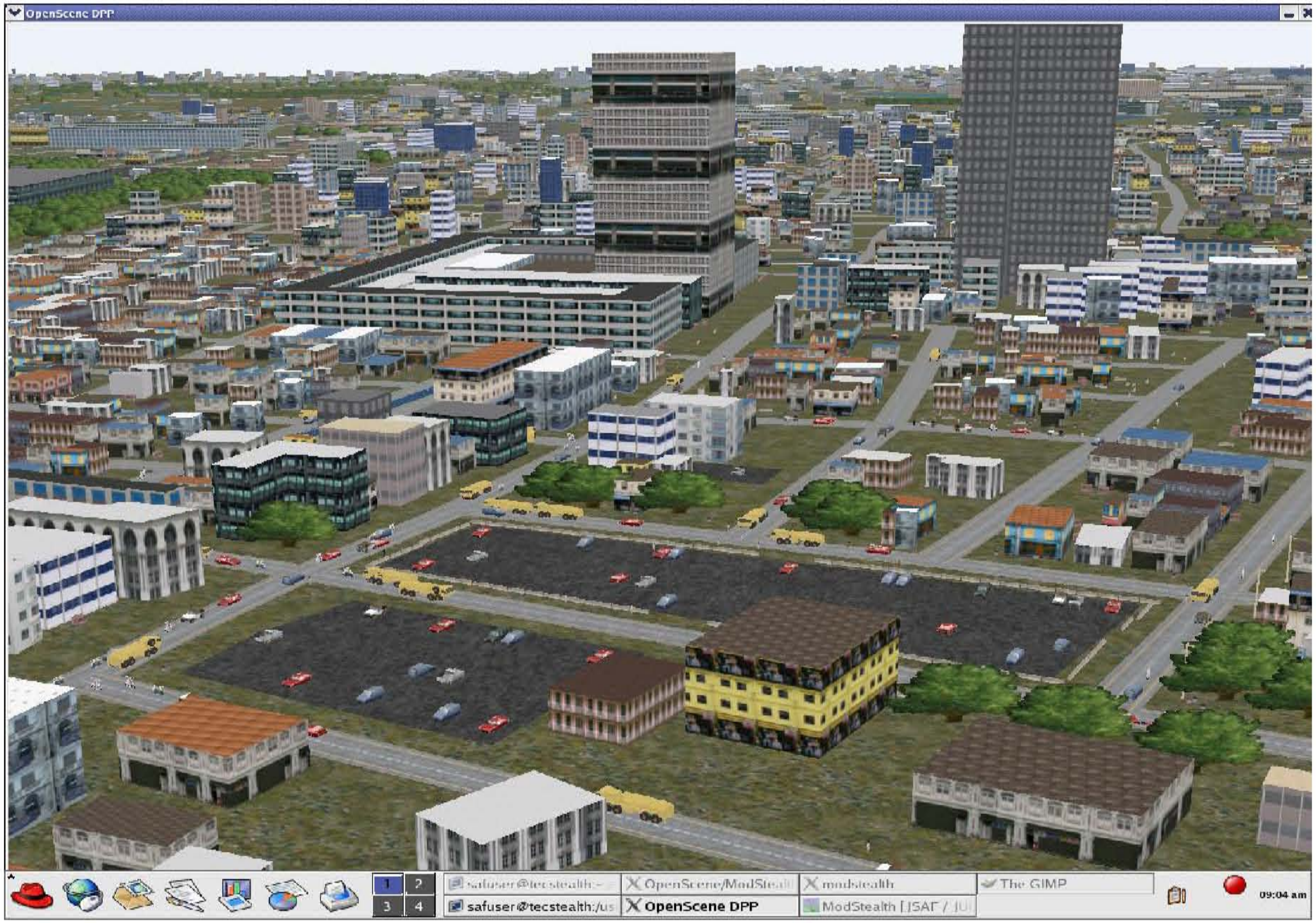
• 47,000 Buildings & Structures in AOI

USC Viterbi
School of Engineering

View of a City

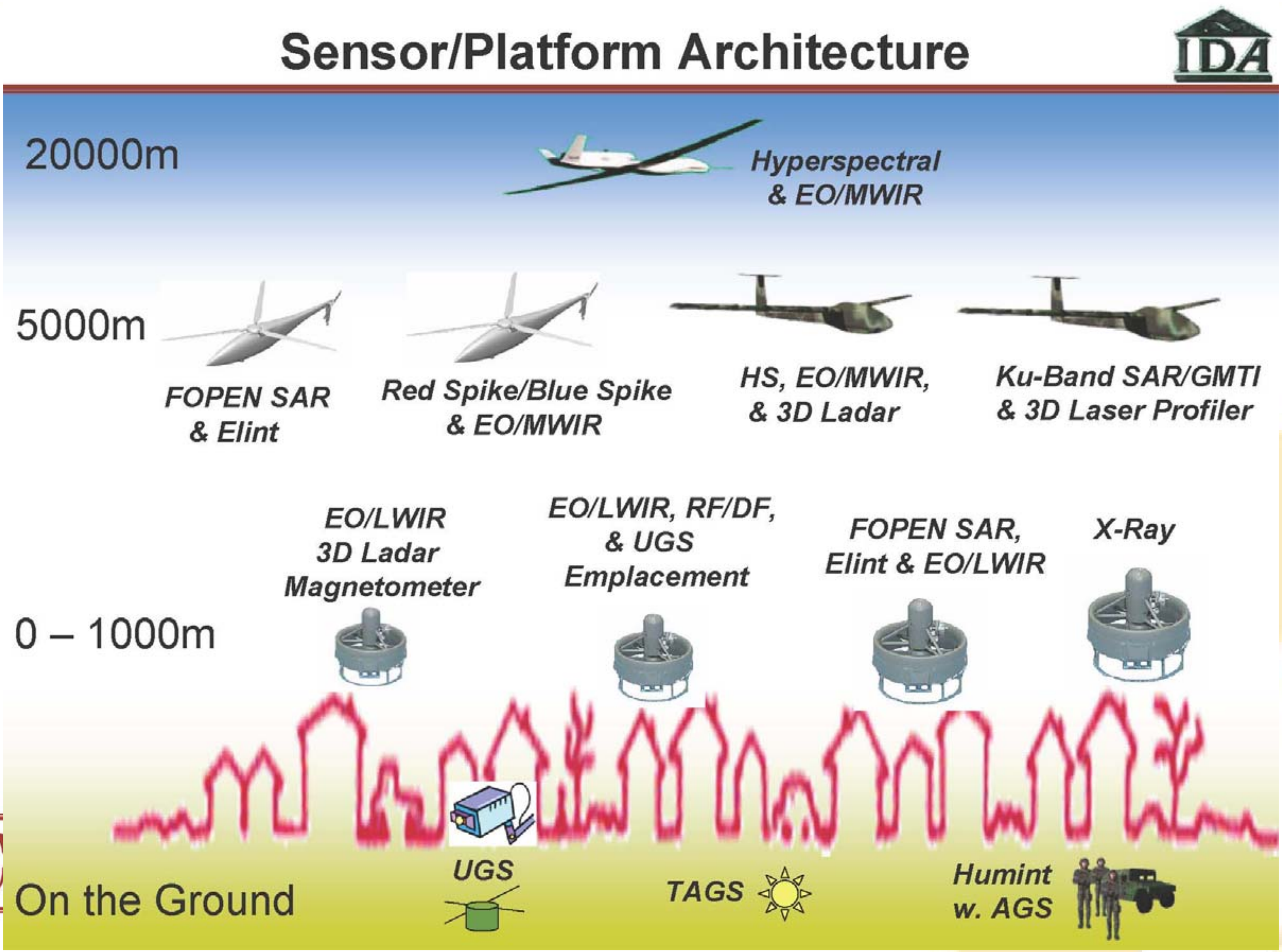
Information Sciences Institute

STEALTH View-Urban Area



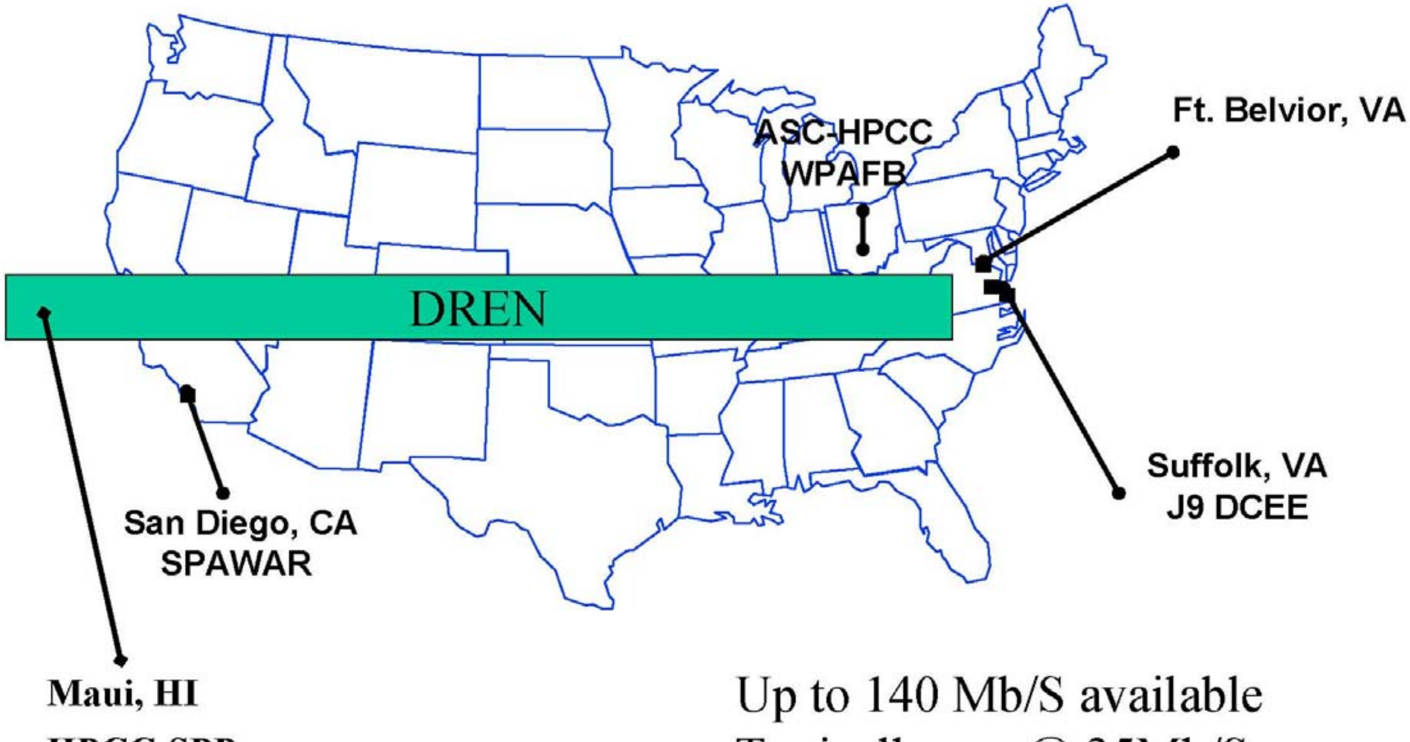
USC

Experimental Sensor Architecture



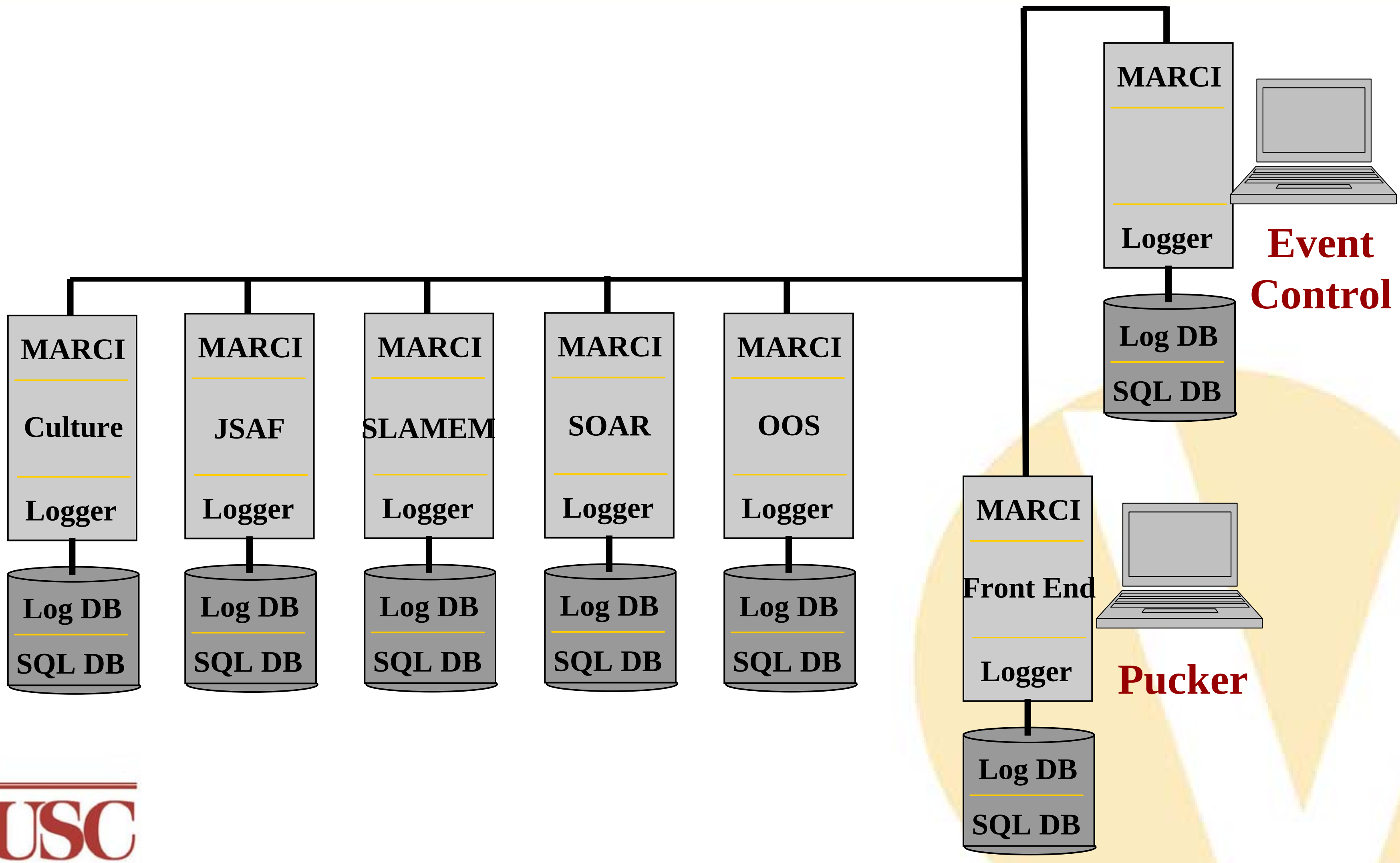
Geographically Distributed

JUO Participants Today



Up to 140 Mb/S available
Typically use @ 35Mb/S

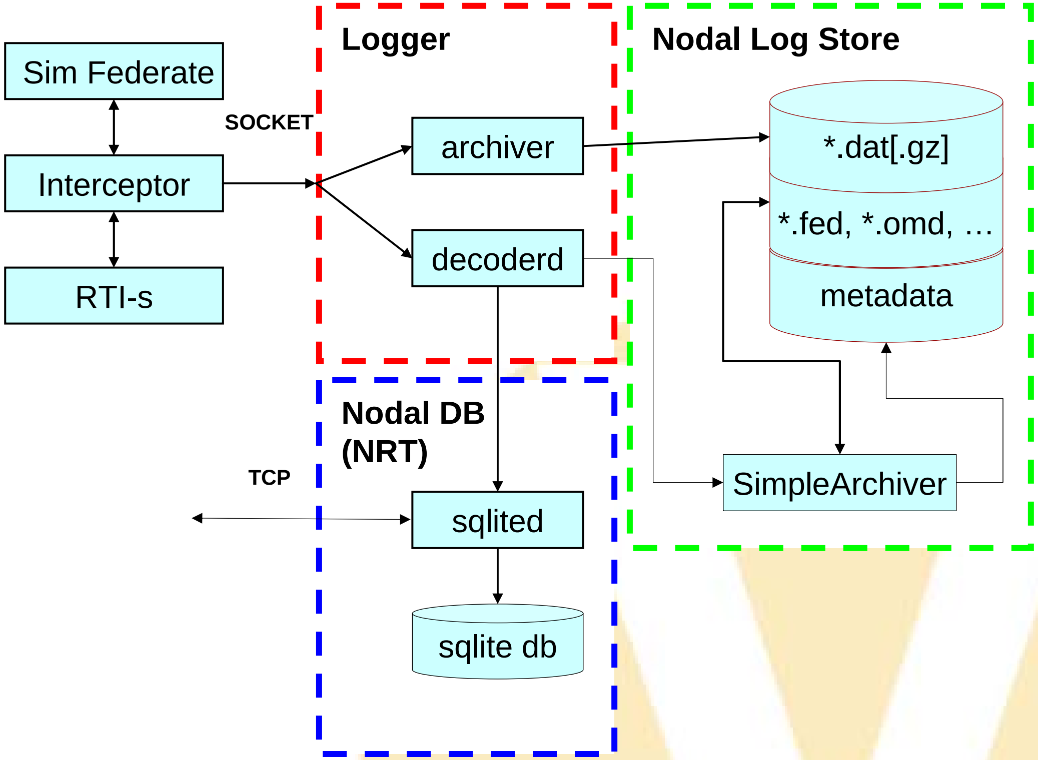
Heterogeneous Computing



Data Intensive Too

Fully distributed logging at point of generation

- **RTI Interceptor** captures simulation events
- **Archiver** stores events to local disk
- **Decoderd** stores events to local relational database
  - Binary DB for R/T queries
  - SQL DB for after action



Issues with Shared Resources

Queuing of interactive jobs

- Code has to run when the people are ready
- Don’t keep the Major General waiting!

Authentication

- Puckers don’t log on. Are they users?
- Our developers all “su – j9”
- They use VNC

Network Security

- MARCI and RTI listen on open ports
- Tolerates failure of any individual node

And of course, the big one …

- Classified data

Dedicated HPC Projects

USC Viterbi School of Engineering

Information Sciences Institute

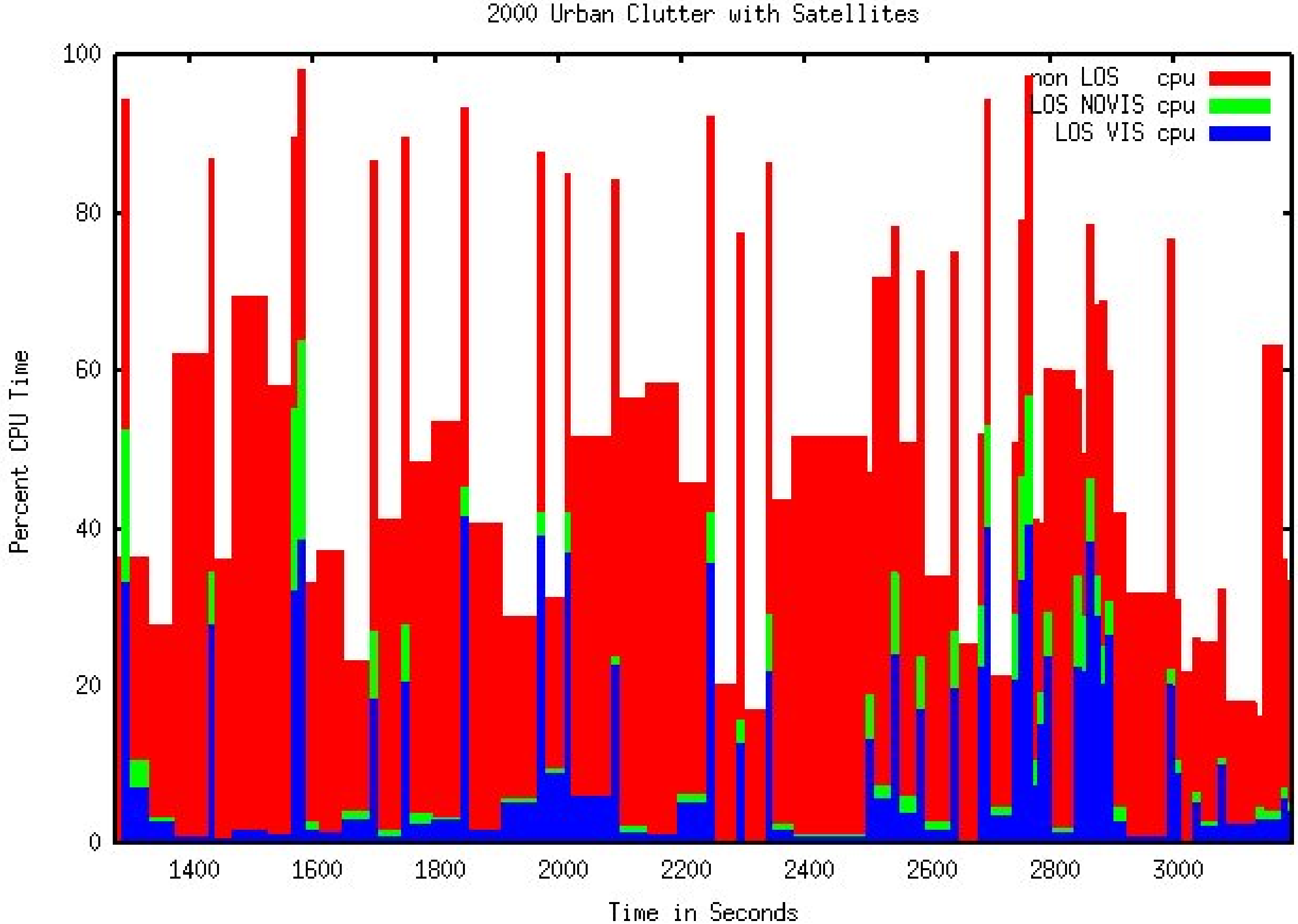
- DOD HPCMPO support for experimentation
- Dedicated systems required due to:
  - Real-time use
  - Security policies
- DHPIs for experimentation:
  - Maui HPCC & ASC in 2004
    - 256 CPUs each
  - JFCOM J7 JATTL in 2007
    - 1024 CPUs
    - 256 GPUs

JFCOM J7 JATTL

USC

Need for Accelerators

Don't port millions of lines of C or Java
We want accelerators to "clip the peaks"



USC Viterbi School of Engineering

GPU Accelerators

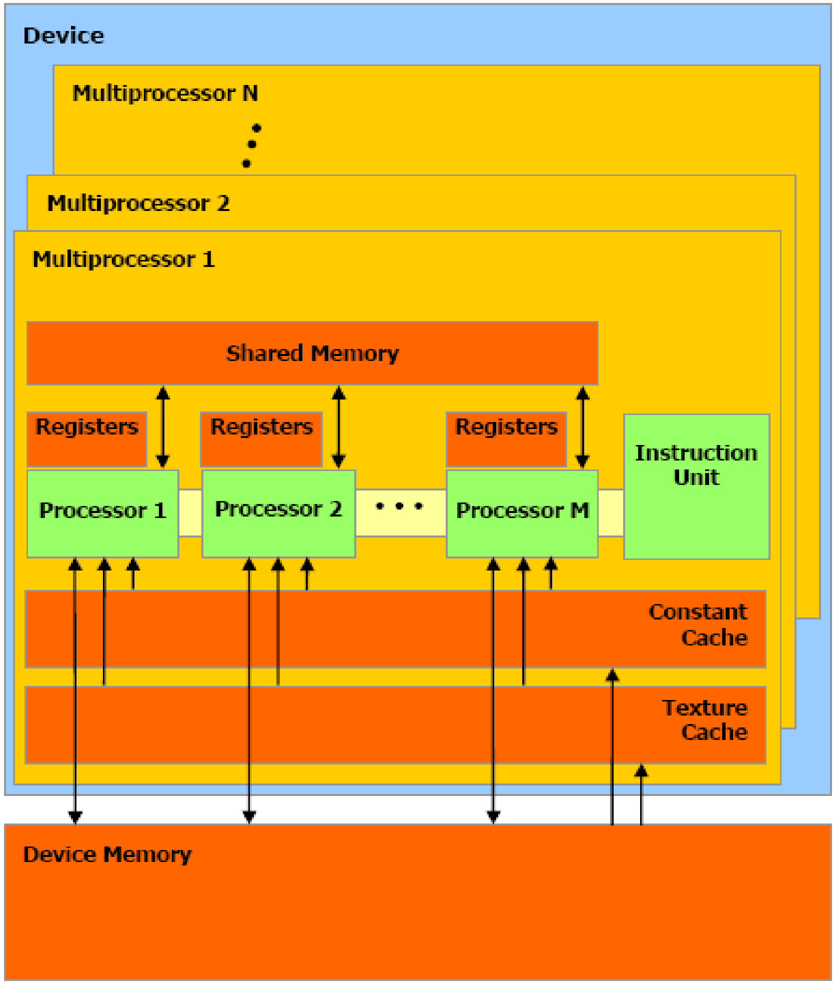
Information Sciences Institute

NVIDIA G80:

- Multi-core processor
- Multi-threaded
- SIMD ALUs
- Explicit hierarchy
- CUDA language
  - MPL descendent

Initial Kernels:

- Line-of-Sight
- Route Planning
- Illumination



A set of SIMD multiprocessors with on-chip shared memory.

Figure 3-1. Hardware Model

Figure courtesy of NVIDIA

USC

CUDA Code Fragment

```
do j = jl, jr
  do i = jr + 1, ld
    x = 0.0
    do k = jl, j - 1
      x = x + s(i, k) * s(k, j)
    end do
    s(i, j) = s(i, j) - x
  end do
end do
```

```
ip=0;
for (j = jl; j <= jr; j++) {
  if(ltid <= (j-1)-jl){
    gpulskj(ip+ltid) = s[IDXS(jl+ltid,j)];
    }
  ip = ip + (j - 1) – jl + 1;
  }

__syncthreads();

for (i  = jr + 1 + tid; i <= ld;
      i += GPUL_THREAD_COUNT) {
  for (j = jl; j <= jr; j++) {
    gpuls(j-jl,ltid) = s[IDXS(i,j)];
    }
  ip=0;
  for (j = jl; j <= jr; j++) {
    x = 0.0f;
    for (k = jl; k <= (j-1); k++) {
      x  = x  + gpuls(k-jl,ltid) * gpulskj(ip);
      ip = ip + 1;
      }
      gpuls(j-jl,ltid) -= x;
    }
  for (j = jl; j <= jr; j++) {
    s[IDXS(i,j)] = gpuls(j-jl,ltid);
    }
  }
```

Catch-22

Ideally one would place entire neighborhoods on one CPU

- Virtue: localize interactions
- Problem: sensor footprints create Amdahl fractions

Solution: "striping" of entities amongst computers

- As the YMP or MTA distributed memory references amongst banks
- Requires each system to hold all the terrain ☹
- Maximizes communication ☹

Ideal solution would be a large shared address space

- Processors pull from work queue, eliminating local bottleneck.
- Eliminate redundant databases (not just terrain)

Shared Memory

- Scalable entities enable increase in fidelity
  - “Artificial Intelligence” rule-driven, behavior models
  - Today fidelity is bounded by the power of one thread
  - Its not going to get better any time soon
  - I can’t imagine parallelizing with MPI

- Enable efficient database queries
  - Today event logs are stored in hundreds of geographically distributed relational databases

Summary
Bruce's Three Points

- Modeling and simulation is moving to HPC
  - Much thanks to HPCMP
  - Big success with ensembles of loosely connected jobs
  - Users work around known problems
- What's holding us back?
  - No scalable entities: entity fidelity limited by power of one thread
  - Distributed databases: limits analytic capability
  - Inertia: 10^7 lines of code & lots of users
- How can new technology help?
  - Needs to be evolutionary (can't abandon TRADOC validated S/W)
  - Heterogeneity OK: accelerators for specific kernels
  - Shared memory for scaling rule-driven behavior models
  - Shared address space:
    - Eliminate redundant storage
    - Enable sophisticated real-time analysis

Acknowledgements

Work supported by AFRL and JFCOM

AFRL Rome Labs Contract F30602-02-C-0213

DHPI systems provided to JFCOM J9 by HPCMPO

2004 DHPI (Koa & Glenn) hosted by MHPCC and ASC

2007 DHPI (Joshua) hosted by JFCOM J7 JATTL

Trick Question

What has?

- **One front end CPU**
- **Eight back end CPUs**
- **SIMD extensions**
- **Private local memories**
- **Shared main memory**

Trick Question

What has?
- One front end CPU
- Eight back end CPUs
- SIMD extensions
- Private local memories
- Shared main memory

Late-model Cray 2 and STI Cell both!

Not quite the same of course
- Only one had CF90: transparent vectorization and autotasking

If STI Cell had CF90, it would be an ideal "crash" machine
- 75% of GM's cycles

Bonus Slides

Summary

Modeling and simulation is a O($1B) business.

- Mainly military training
- Civilian use growing

Accelerators appear to be low-hanging fruit.

- GPUs are cheap and ubiquitous
- Augment standard PCs

Share memory critical to our future

- Scalable entities
- Real-time analysis

Lots of inertia to overcome ☹