

# HPEC 2003



## **Gedae Runtime Kernel Performance Characterization**

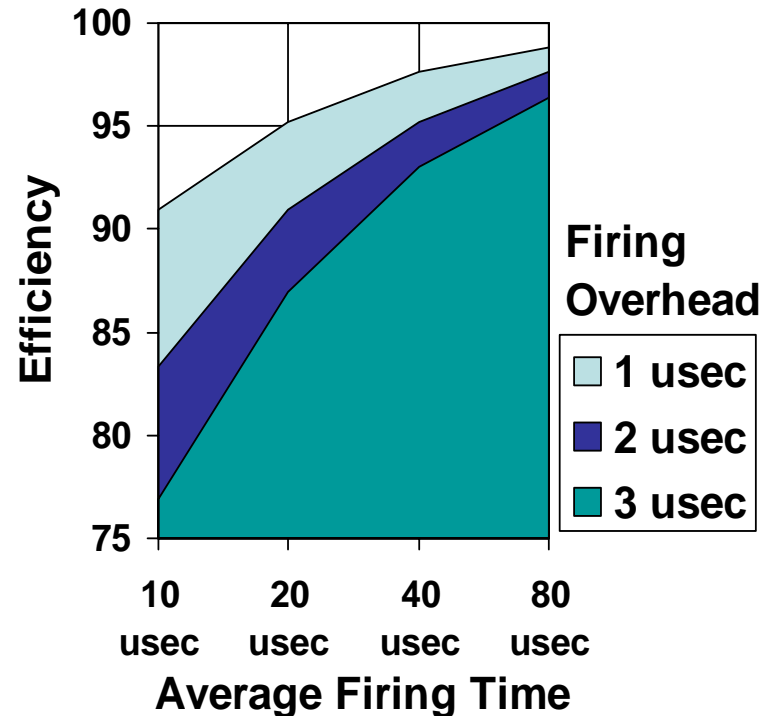
**Kerry Barnes, William Lundgren,  
James Steed  
Gedae, Inc.**

# Efficiency of Gedae Applications



- How efficient are Gedae applications compared to hand coding?
- What is the overhead of the Gedae Runtime Kernel?
- What is the cost of executing a primitive?
- What is the cost of dynamic vs. static scheduling?

Efficiency vs Firing Overhead and Average Firing Time



# Instrumenting Kernel



- **Gedae Kernel instrumented with timer functions**
- **44 sections of code were timed**
- **Benchmarks**
  - **rt\_stay\_easy: static scheduling**
  - **e\_comm: dynamic scheduling**
  - **noise\_removal: dynamic scheduling with segmentation**
- **Timings gathered on AltiVec targets from 3 vendors**
- **1-3 usec overhead for all benchmarks**

Graph:	rt_stap_easy		e_comm		noise_removal	
Proc	Efficiency %	Overhead (usec/fire)	Efficiency %	Overhead (usec/fire)	Efficiency %	Overhead (usec/fire)
AltiVec1	97	1.7	90	2.7	95	1.8

# Categorize Overhead



- Four categories accounted for most of the overhead time
- Suggested enhancements:
  - Code generation of static schedule
  - Eliminate copies by directly adjusting pointers
  - Improve schedule state change functions

Function	rt_stap_easy	e_comm	noise_removal
run-sched-pass	90%	20%	18%
copy functions		18%	9%
schedule state		36%	26%
queue management		10%	15%
Total % of Overhead	90%	84%	68%

**We can comprehensively measure the overhead and systematically optimize the most costly components**