Interstitial Computing:
Utilizing Spare Cycles on Supercomputers

Steve Kleban

and

Scott Clearwater

Sandia National Laboratories

Outline

- What is Interstitial Computing?
- The Performance/Utilization trade-off
- An Interstitial Project
- Continuous Interstitial Computing
- Summary

Interstitial

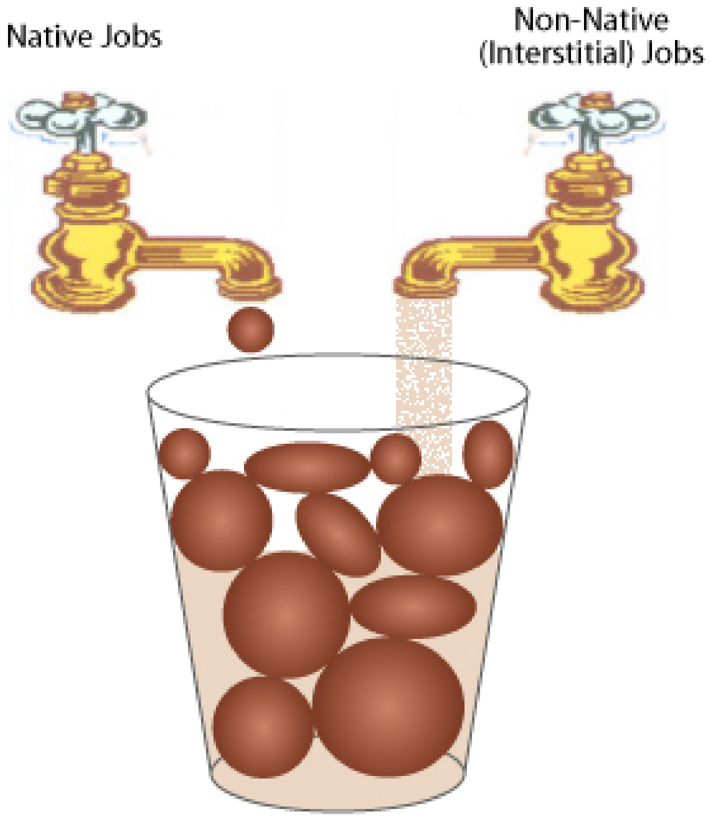
- Definition - a narrow or small space between things or parts

- As seen is physics - space between atoms in a solid

- As seen is biology - space between cells

Interstitial Computing

Interstitial Computing: Utilizing the unused resources in time and space on a cluster with many small jobs.



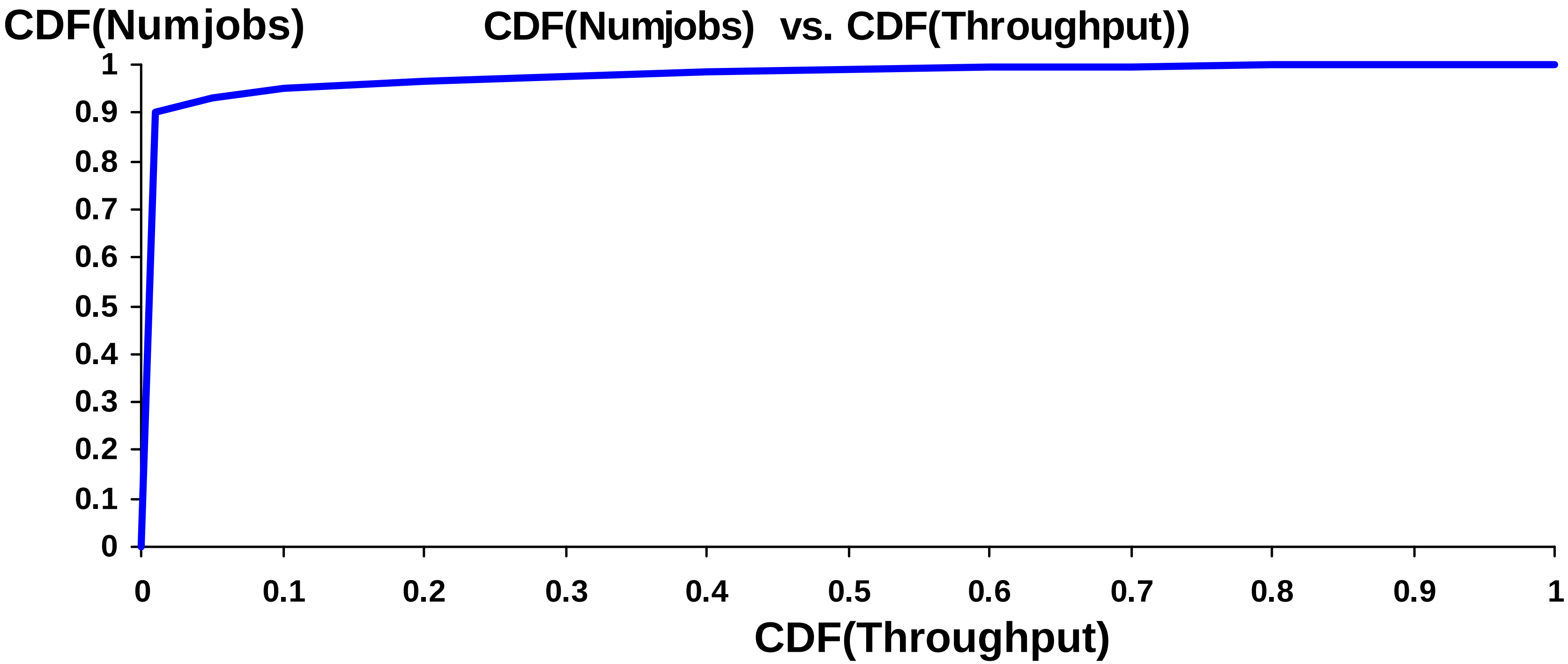
SETI@home, Condor

Why are there Interstices?

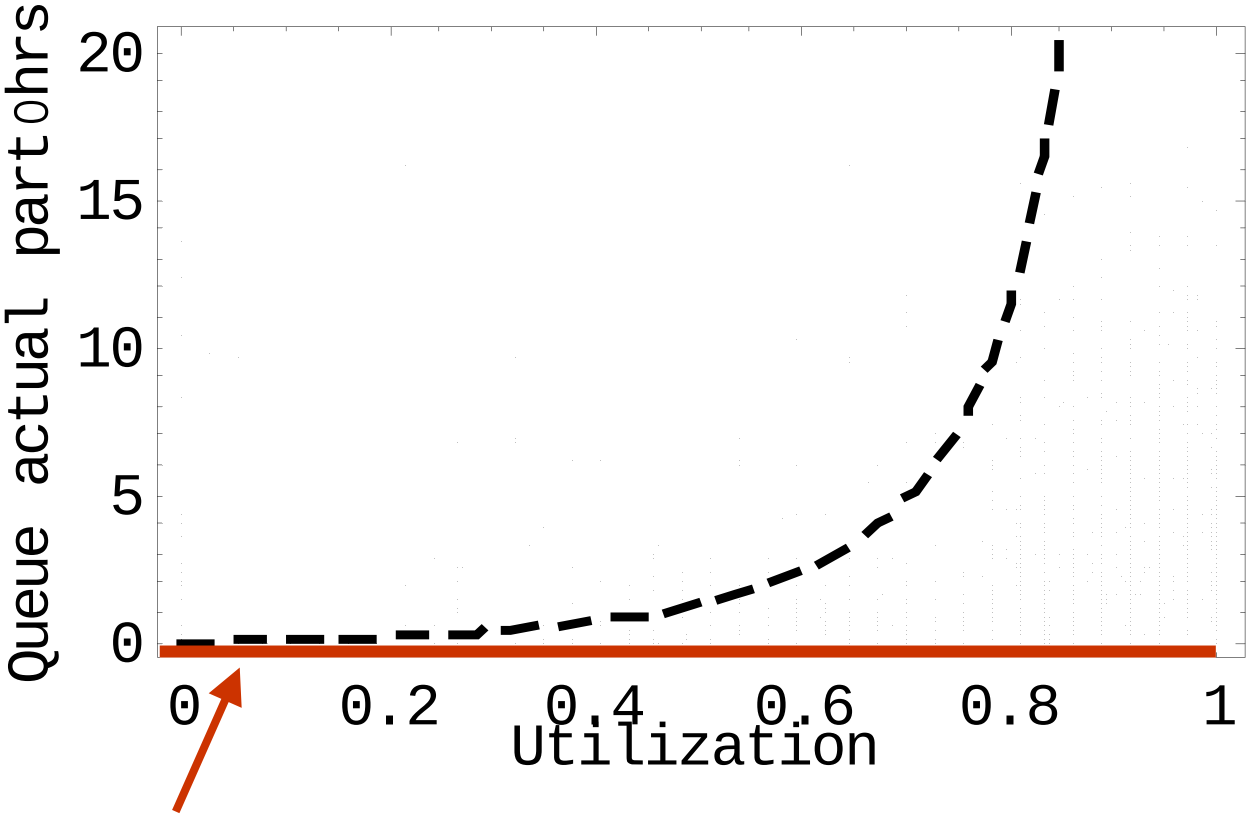
- Created by job mix and submission pattern
- Most of throughput of the machine is due to large jobs--fulfills major objective of the machine
- Not enough small jobs in mix to keep machine at high utilization
- Bursty submissions
  - Long queue may exist but jobs may not fit
  - No queue may exist, resources go unused

Highly Skewed Job Mix Blue Mountain

8,171 jobs in 79 days   Avg. Utilization 79%

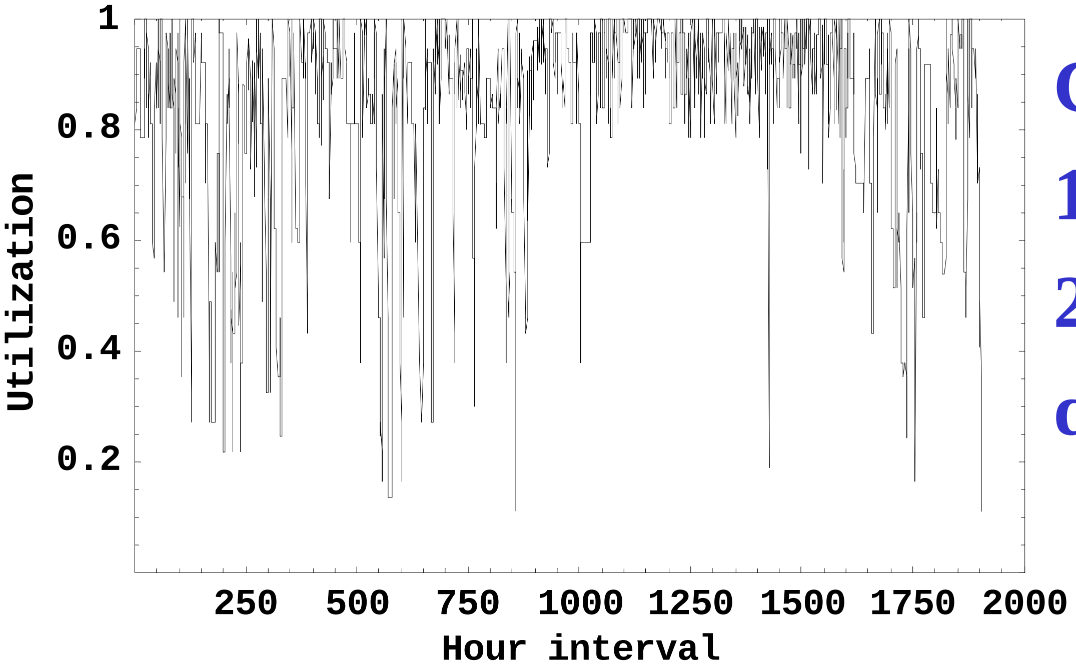


Queue Length and Utilization



At all utilizations, note zero queue lengths which means resources are unused.

Erratic Utilization



Combination of:

1) periods of low demand

2) inability to bin pack during high demand

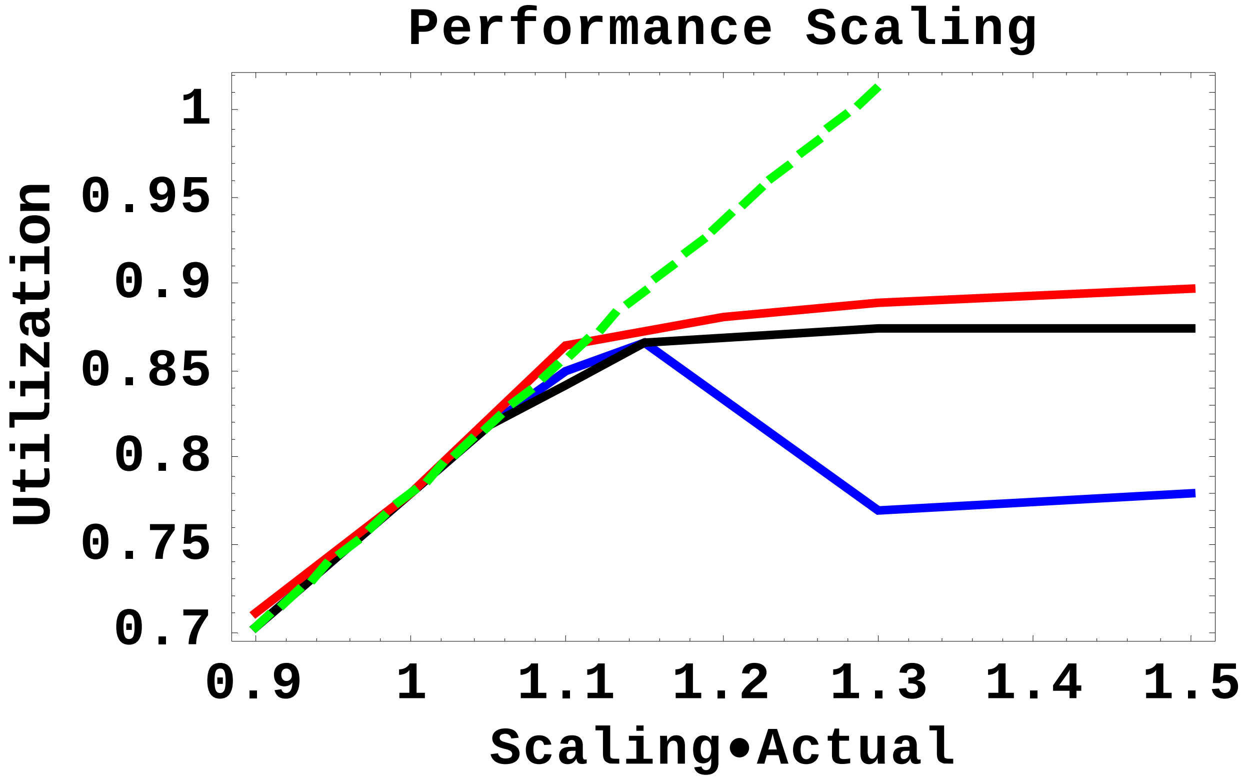
What can we do?

What can we do to increase utilization with minimal impact on makespan of native job?

1.) Scaling - runtime, CPUs, or arrive rate

2.) Interstitial Computing

Why Scaling Won’t Work-I



On BlueMt, scaling works only for ~10%

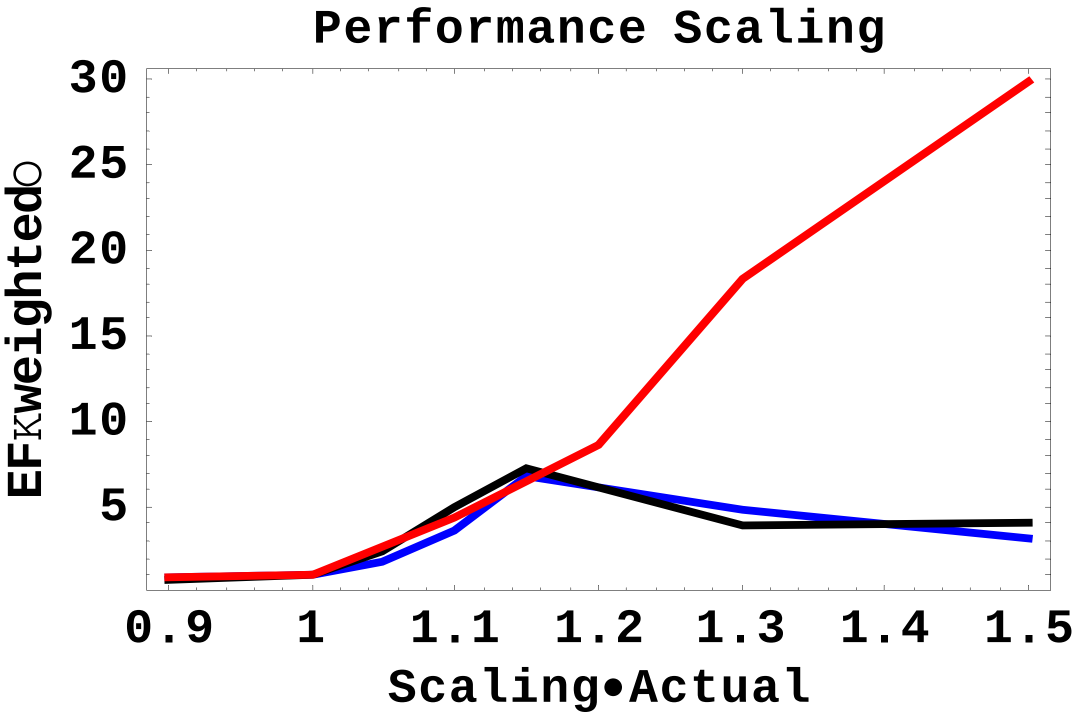
(red)submission rate scaling

(blue)run-time scaling

(black)CPU scaling

(green) perfect scaling

Why Scaling Won’t Work-II



EF = 1+wait/run

Lower EF because utilization also lower.

(red)submission rate scaling
(blue)run-time scaling
(black)CPU scaling

Interstitial Computing – The Algorithm

- The jobs
  - Fixed number (project)
  - Continuous
- Run at lowest priority
  - Only when native jobs cannot run
  - After aggressive backfill of native jobs
- Minimize impact on native jobs

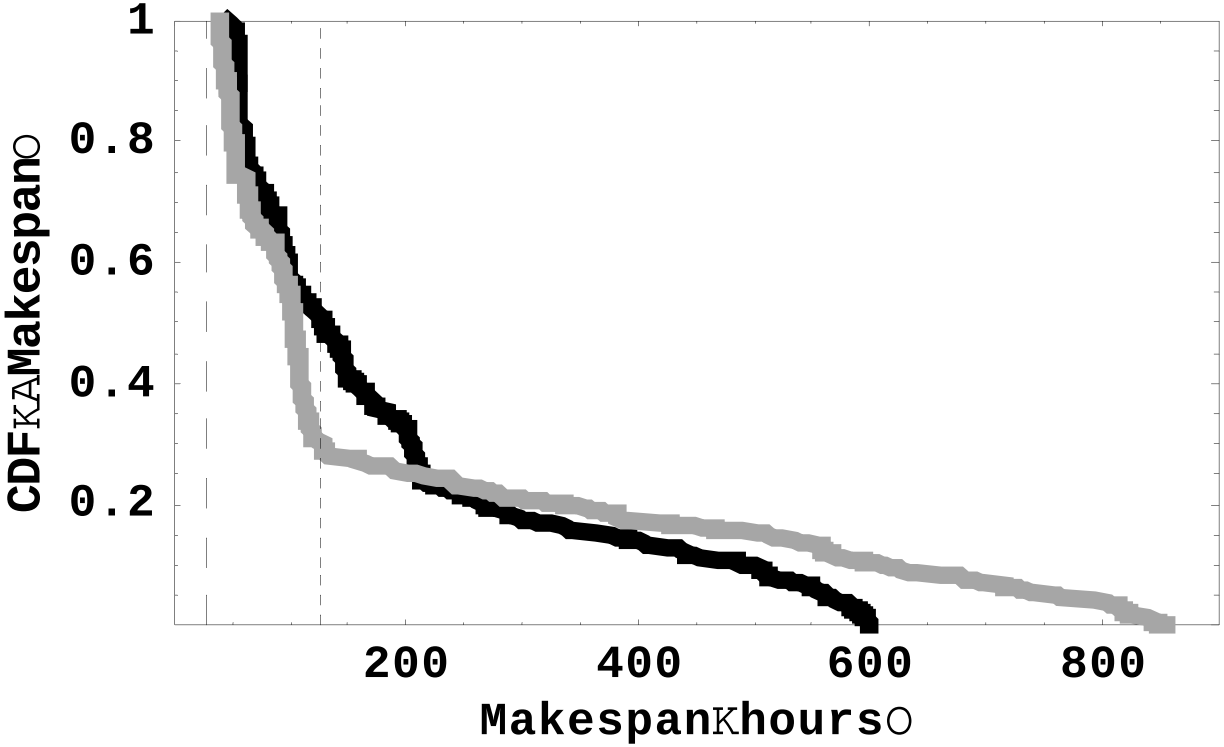
Test Machines

	Ross Sandia	Blue Mtn Los Alamos	Blue Pacific Livermore
CPUs	1436	4662 (large partition)	926 (subset)
clock GHz	0.588	0.262	0.369
TCycles	0.844	1.221	0.342
Utilization	.631	.790	.907
log days	40.7	84.2	63
Jobs	4,423	7,763	12,761
Queue algorithm	Portable Batch System (PBS)	Load Sharing Facility (LSF)	Distributed Production Control System (DPCS)
Log date	2002	2001	2001

An Interstitial Project

- Fixed number of jobs
- Constant small number of CPUs/job (<few percent of machine)
- Constant short run time (<tens minutes)
- Drop projects in at random times during the simulated run

Distribution of Finishing Times



Projects have a wide distribution of finishing times because of highly variable utilization.

(Long dashed)   absolute minimum

(Short dashed) average minimum with this utilization

(Black)32CPU x 120sec x 32kjobs

(Gray) 32CPU x 960sec x 4kjobs

Continual Interstitial Computing

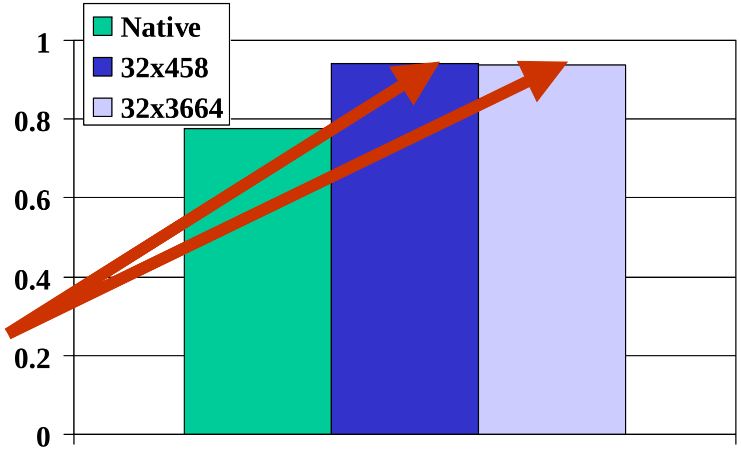
Blue Mountain

	Native Jobs	32CPU × 458sec	32CPU × 3664sec
Interstitial jobs Native jobs	0 8,171	408,685 8,171	49,465 8,171
Overall Util	.776	.942	.939
Native Util	.776	.776	.776
MedianWait sec all / 5% largest	0.0k / 1k	0.2k / 4.4k	0.4k / 5.7k

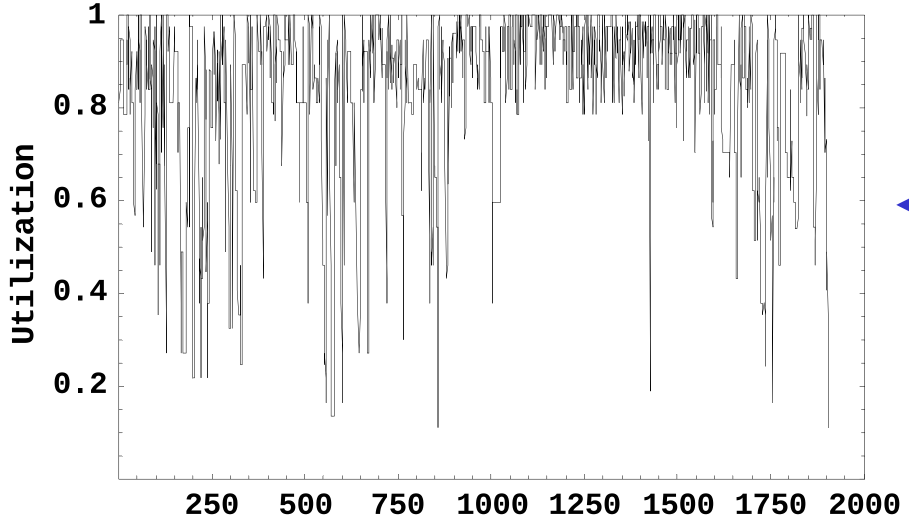
Little effect on utilization of native jobs.

Some effect on wait time.

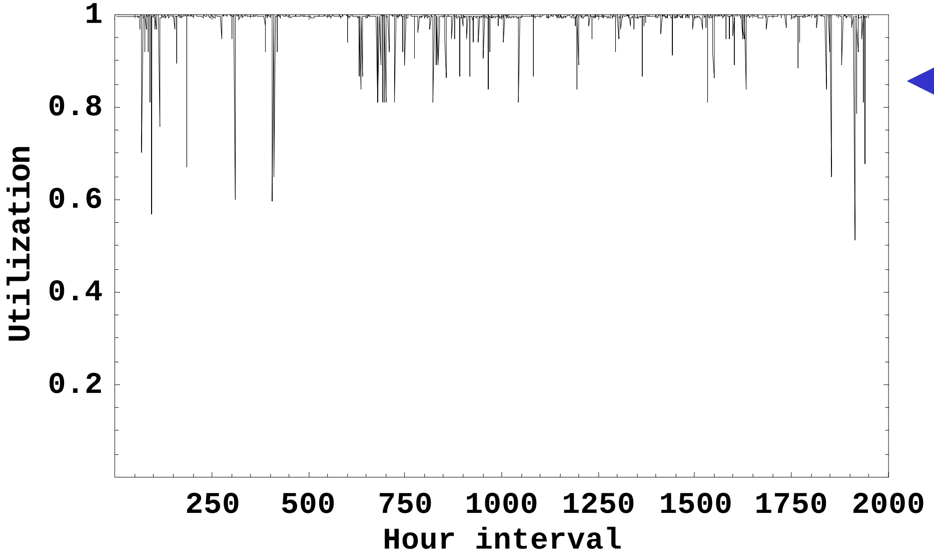
Very high utilization overall.



Utilization: Before and After



←Before



←After

Nearly all utilization has been captured.

Continual Interstitial Computing

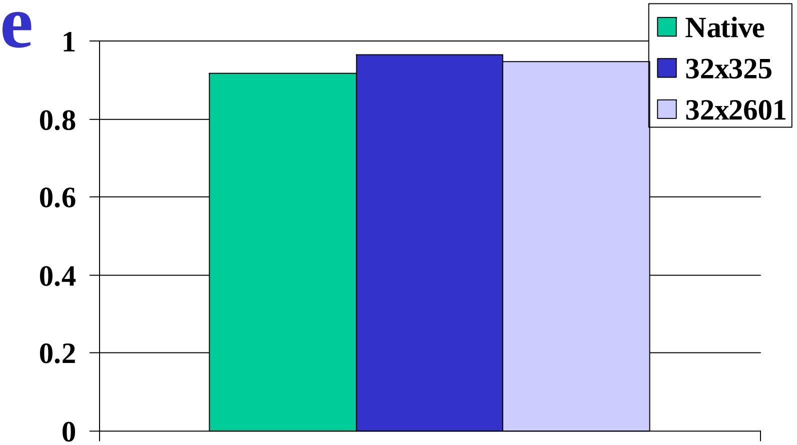
Blue Pacific

	Native Jobs	32CPU × 325sec	32CPU × 2601sec
Interstitial jobs Native jobs	0 10,465	11,392 10,383	1,066 10,346
Overall Util	.916	.964	.946
Native Util	.916	.900	.898
Median Wait sec all / 5% largest	2.1k / 79k	2.0k / 86k	2.5k / 86k

Little effect on utilization of native jobs.

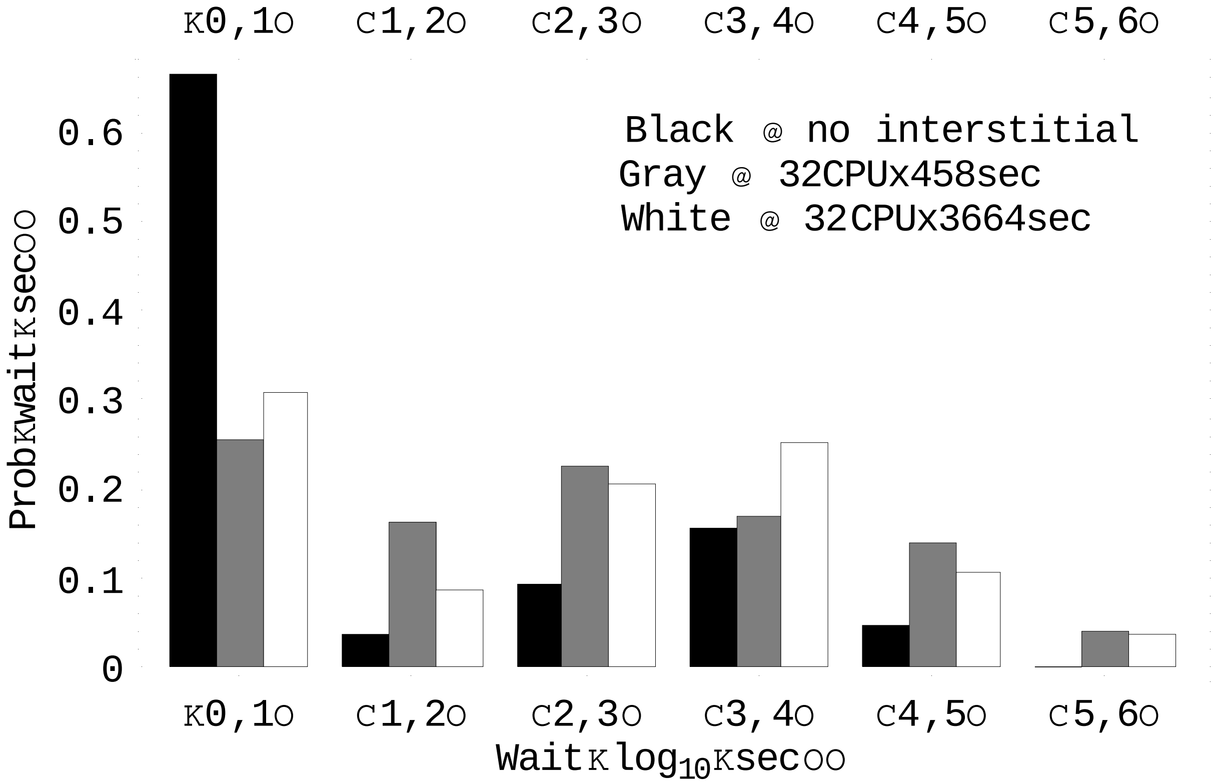
Some effect on wait time.

Small change in utilization because it was already ~90%.



	Native Jobs	32CPU × 204sec	32CPU × 1633sec

Effects on Native Jobs



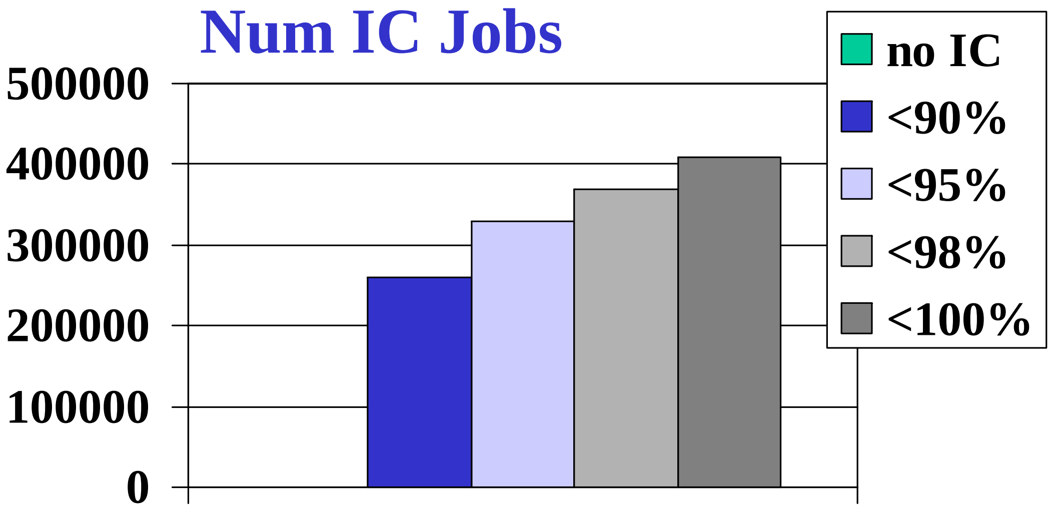
Delay of native jobs by ~IC run-time.

Cascade effect pushes entire distribution of wait times to higher values.

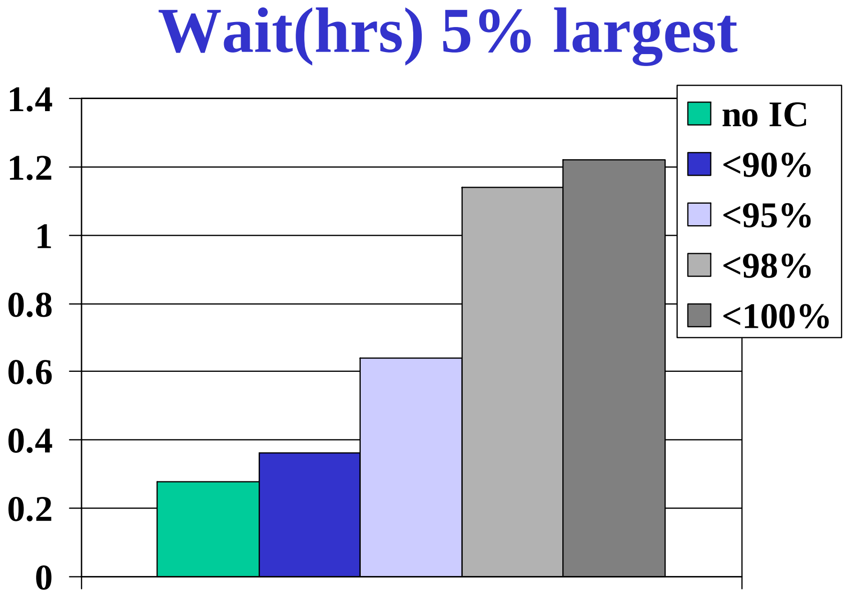
National Nuclear Security Administration

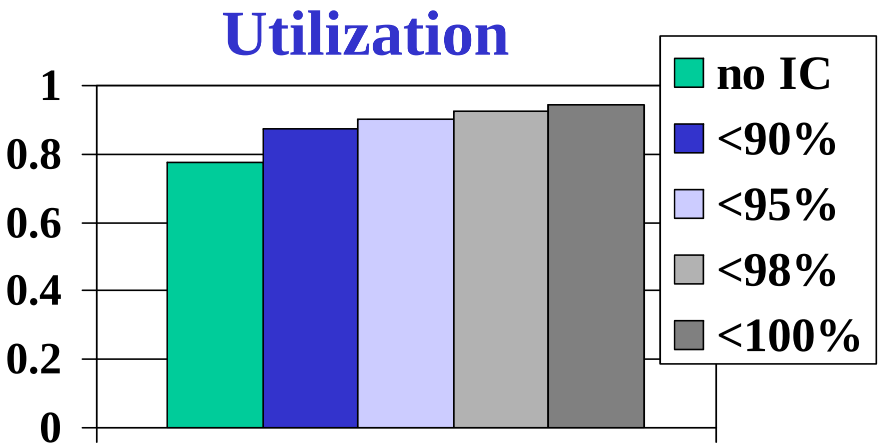
Sandia National Laboratories

Limited Continual IC

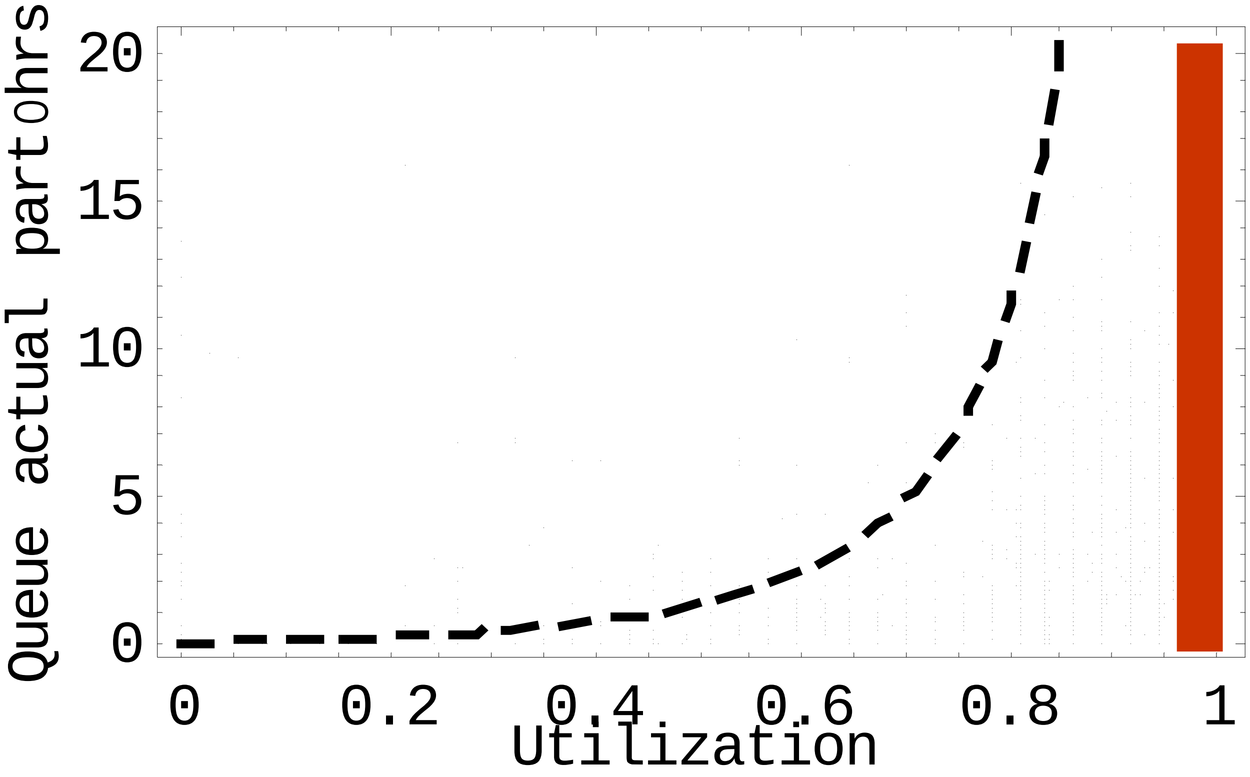


Only submit IC jobs when Util < x%





What have we done?



Summary

- Interstitial computing depends on native job mix and utilization.
- BlueMtn, limited 98% , util goes from 79% to 92% with little impact on native jobs

Summary (cont'd)

Rules of Thumb for Interstitial Computing

- number CPUs/IC job << avg. available CPUs
- run-time/IC job << avg. native run-time
- queue system must be able to handle thousands of jobs
- native utilization < 90%
- IC jobs must be self-contained

Future Work

- Explore different IC job runtime lengths for Interstitial Computing
- Explore in more detail the relationship between CPU/job and utilization for Interstitial Computing

Acknowledgements

The authors gratefully acknowledge the assistance of Stephany Boucher, Charles Hales, Michael Hannah, Steve Humphreys, Moe Jette, Wilbur Johnson, Tom Klingner, Jerry Melendez, Amy Pezzoni, Randall Rheinheimer, Phil Salazar, Bob Wood, and Andy Yoo. We also thank John Noe for his encouragement and support.