A Performance Model of Fast 2D-DCT Parallel JPEG Encoding Using CUDA GPU and SMP-Architecture

Mohammed K. Shatnawi
Hussein A. Shatnawi

University of Ottawa
SAU University

Presentation Outline

- Introduction
- Parallel JPEG Implementation
- Performance Evaluation
- Experimental Results
- Conclusion

Introduction

- Multi-core Processor
- NVIDIA video card with GPU
- CUDA
- Main Features of GPU CUDA
- SESC

Introduction

- Comparing JPEG algorithm implementation on SESC with GPU

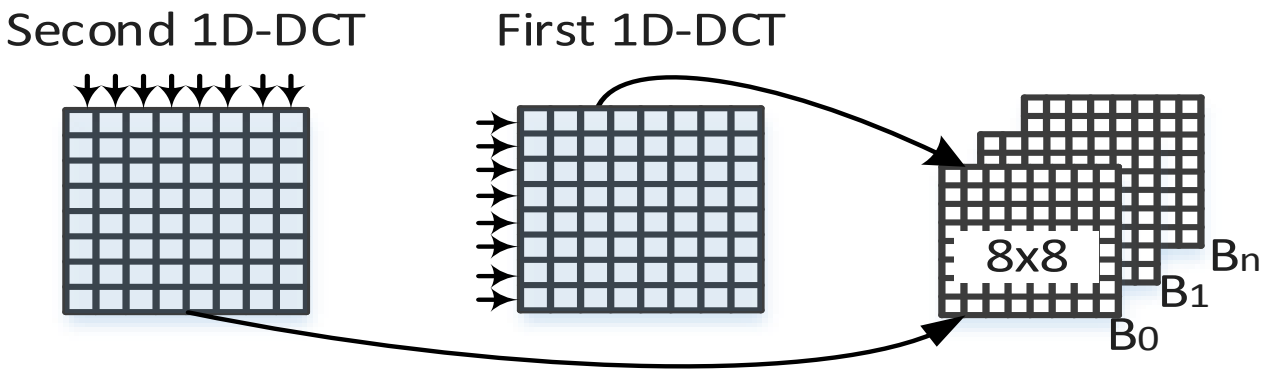
Parallel JPEG Implementation

- JPEG steps:
  - Convert from RGB to YCBCR
  - Discrete cosine transform (DCT)
  - Quantization
  - Encoding

Parallel JPEG Implementation

- Running JPEG on CUDA GPU and SESC
  - Small image
  - Big image

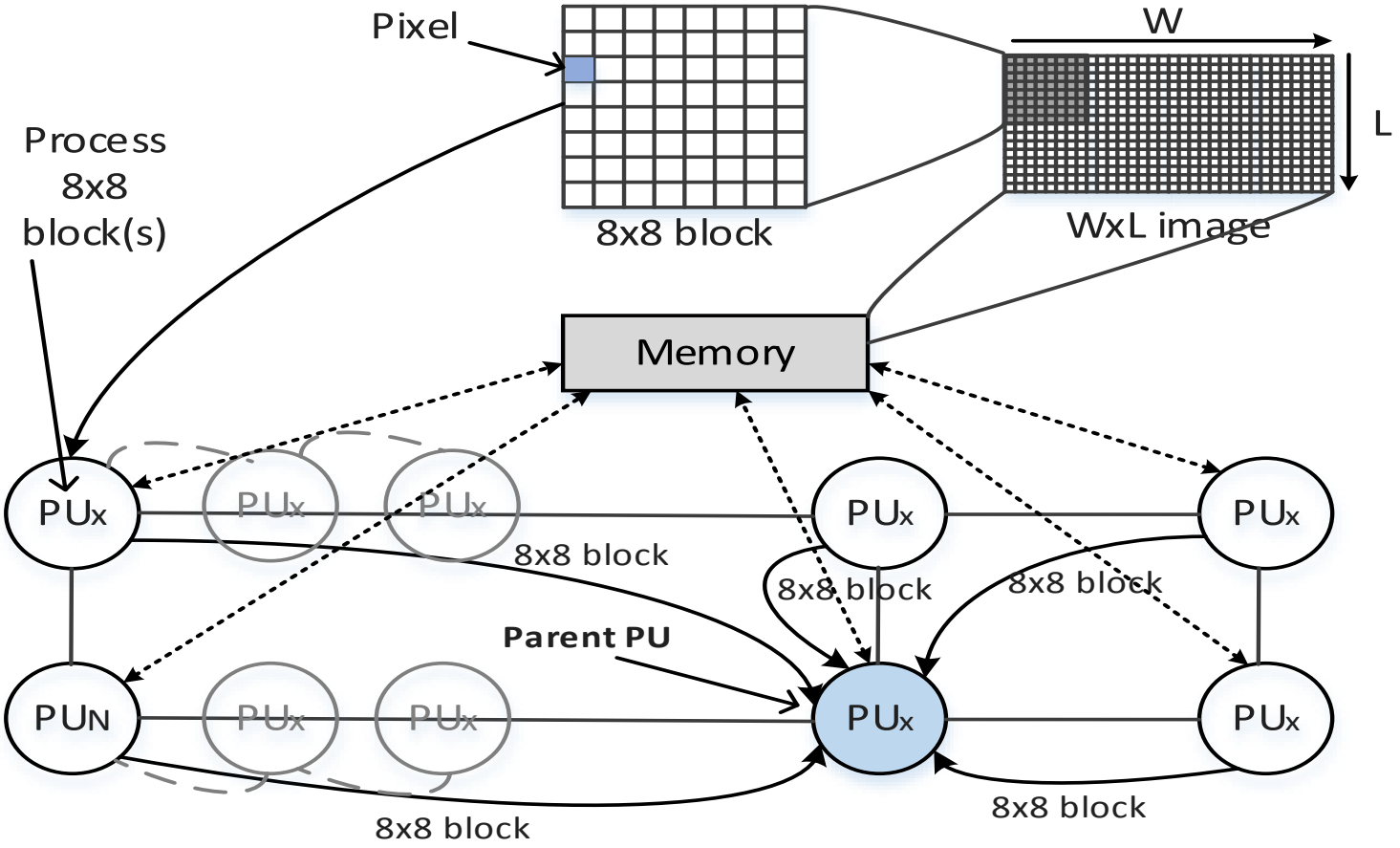
Fast 2D-DCT



Parallel block transposes in the FCT

Parallel Model Analysis

- Processing JPEG among available PUs and the main memory

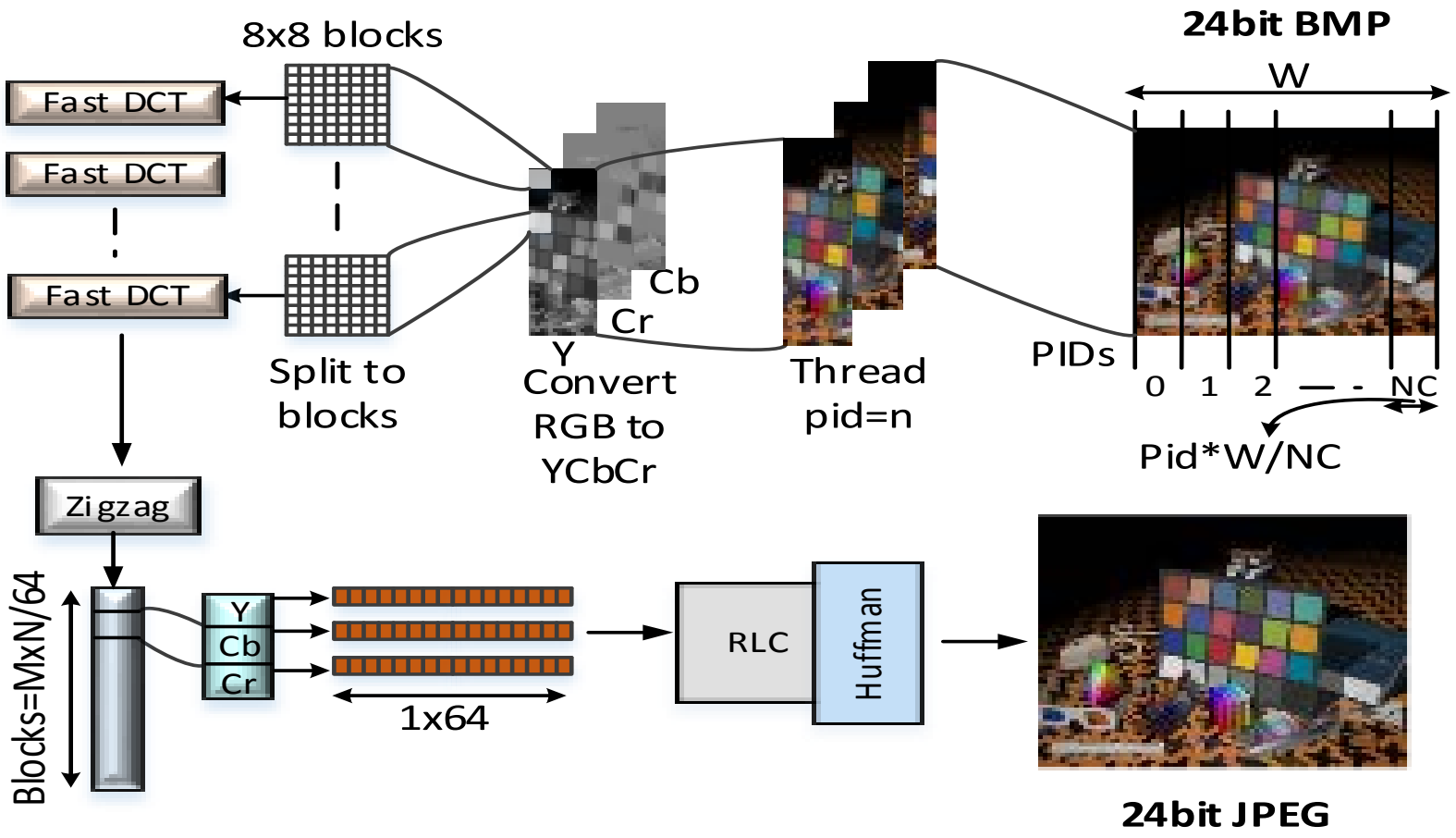


Cross-Architectural Design

- Guarantee of cross-compatibility on GPU and SMD systems.
- Provide cross portability across different hardware

Cross-Architectural Design

- Proposed Algorithm stages for SMP and GPU



Performance Evaluation

- Optimized GPU and out-of-order optimized SMP architecture
  - System Specifications
  - Evaluation Metrics

System Specifications

	System	
	SMD	***GPU***
Architecture	Symmetric	Symmetric grid
Technology	70nm	40nm
Memory	256-bit shared	128-bit global
Processor clock	5GHz	1.4GHz
int, fp registers	128-bit	64-bit
NP, N	[1-32]	96

Evaluation Metrics

- Speedup: the improvement of parallel code in the range of [0, N] or [0, NP]
- Efficiency: the average speedup of each PU in range of [0, 1]
- Efficiency cutoff point: the highest number of (N or NP) at which $\eta > 50\%$.
- Utilization: the percentage of available utilized resources $U = m_i/NP \; or \; U = m_i/NP$

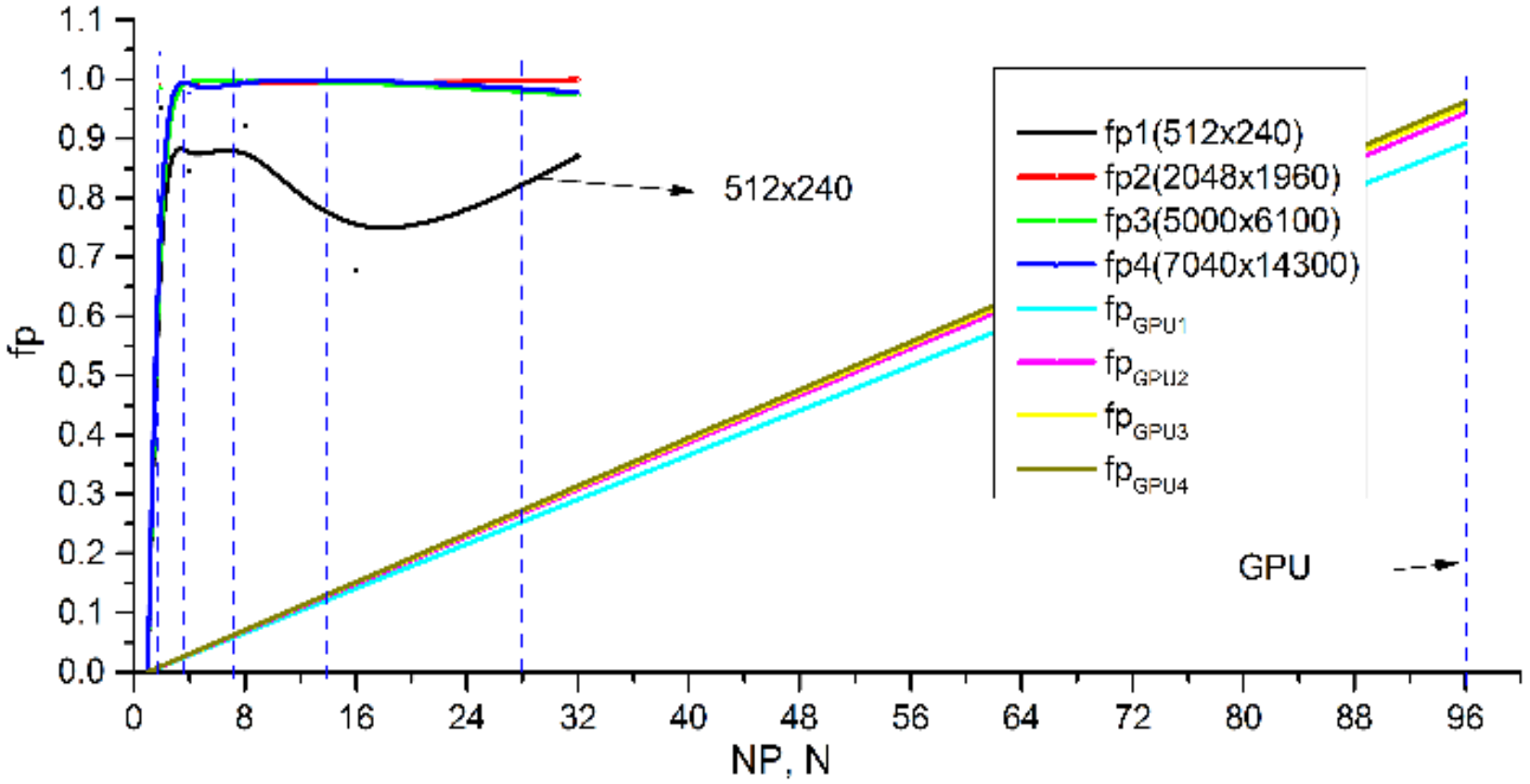
Evaluation Metrics

- Scalability: how η maintains constant when input size and (N or NP) increase.
- *Amdahl's law (S_{max}):* $S_{max} = \frac{1}{f_s + \frac{fp}{N}}$
- Karp-Flatt: define the sequential fraction f_s . The less the value f_s the better the performance. Let $P = N\ or\ NP$, then:

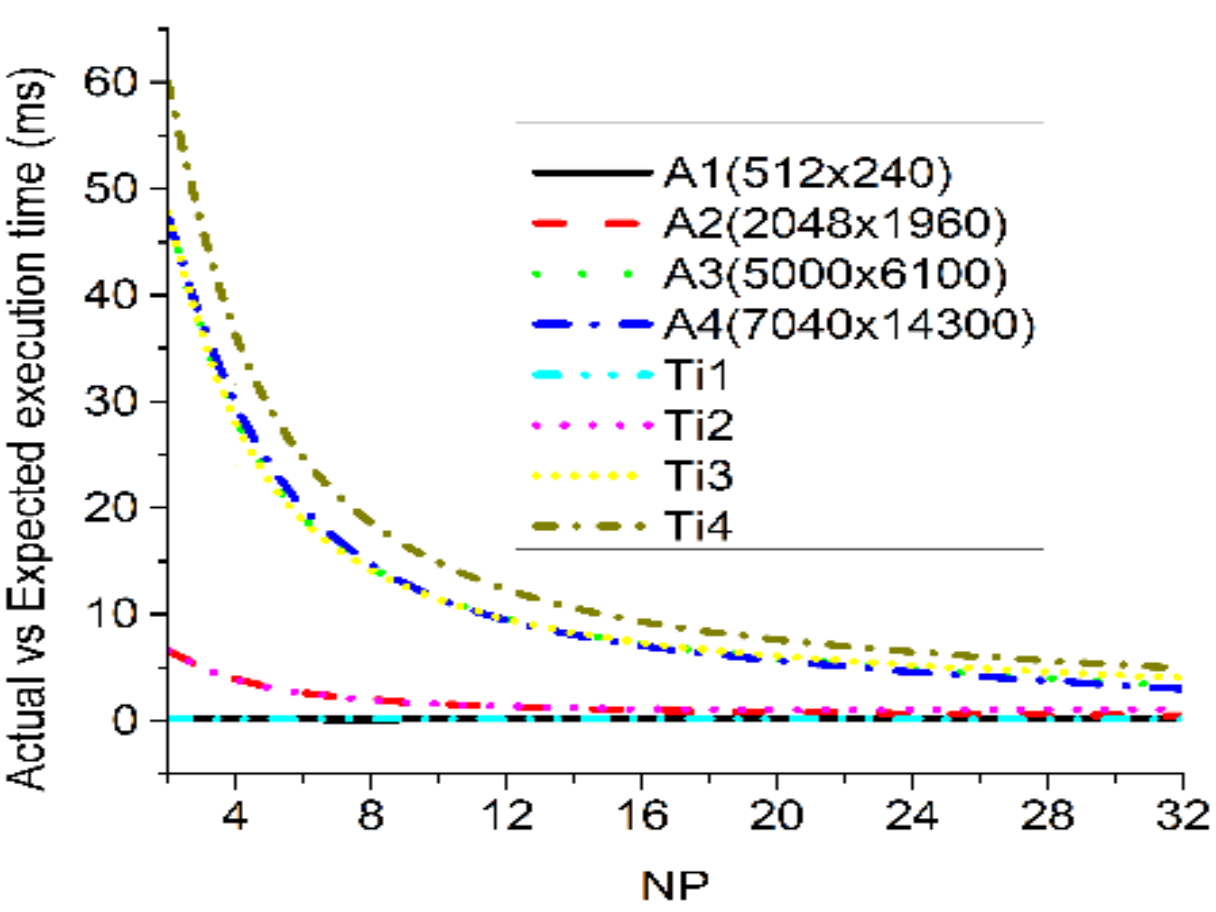
$$f_s = \frac{P - S}{S(P - 1)}$$

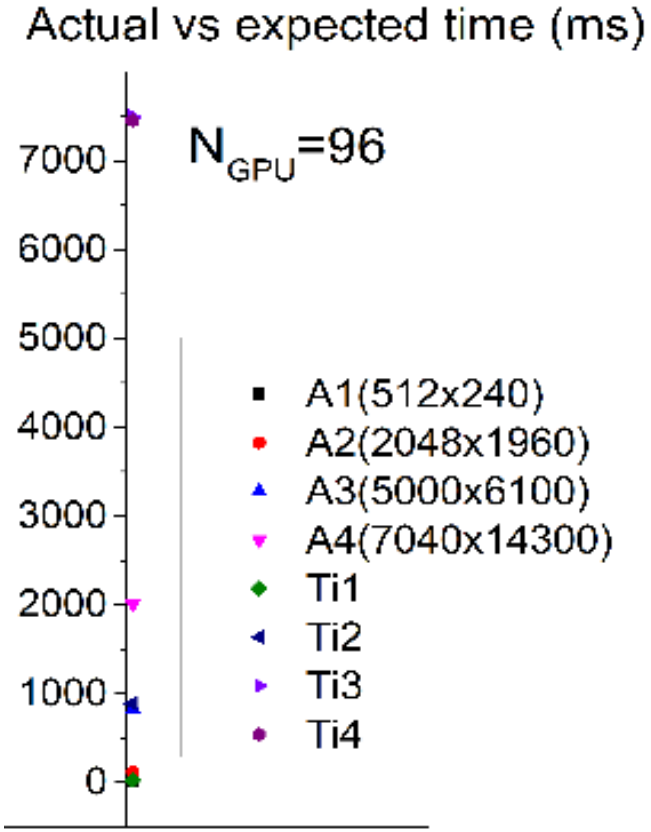
Experimental Results

The theoretical f_p values for 4 BMP images

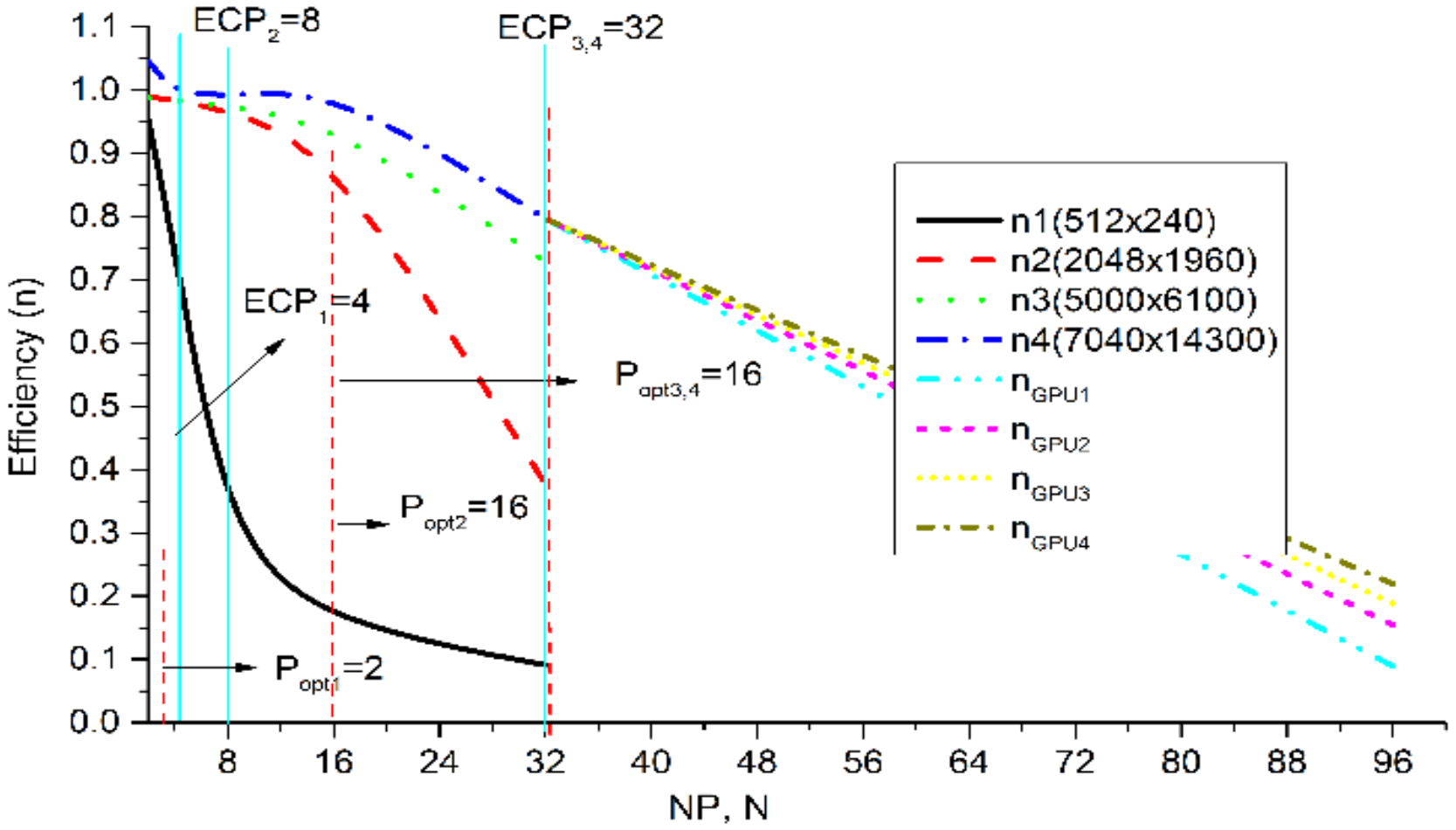


Encoding times and T_{i+1} values in SMP and GPU simulation

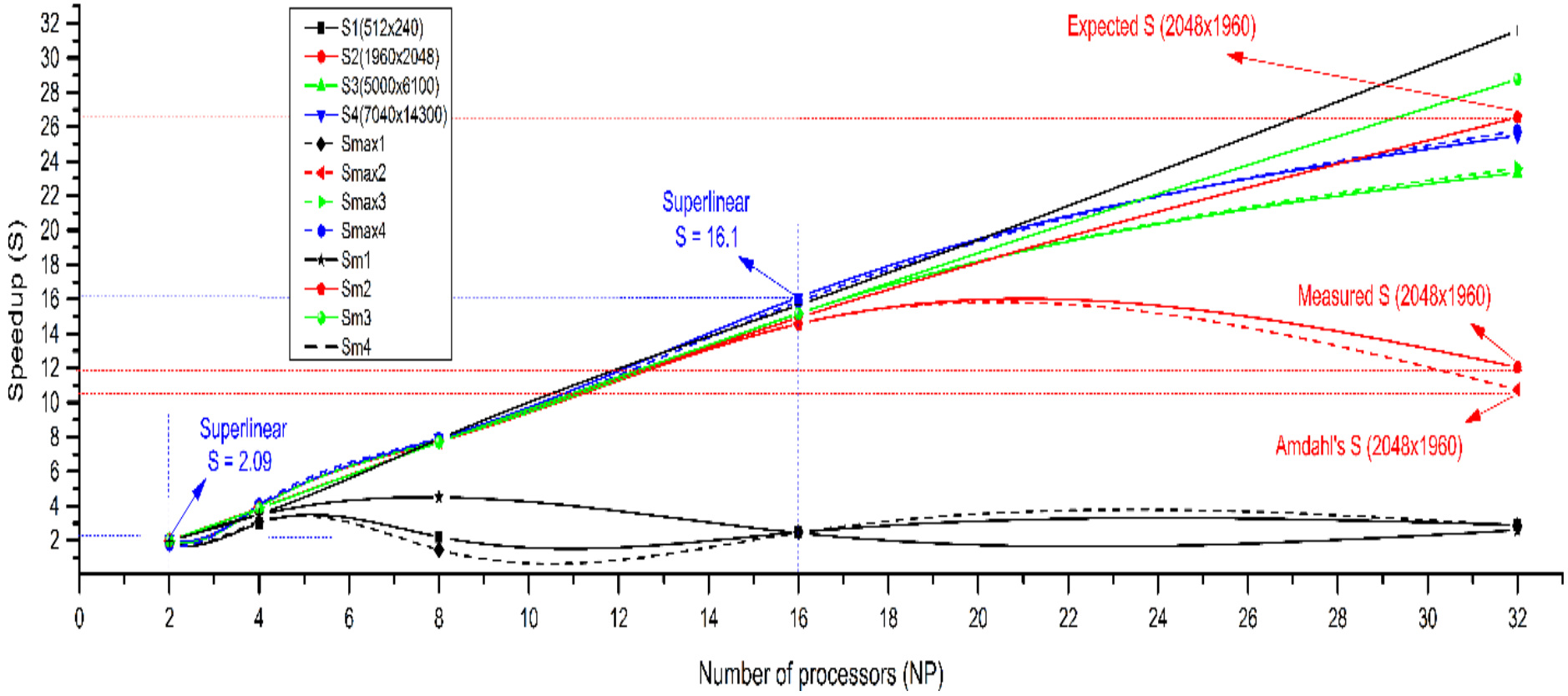




The efficiency, ECP and the optimal number of processors



Runtime speedup for 4 24-bit BMP images on multiple NP in SMP architecture



GPU SIMULATION RESULT

	24-bit BMP images			
	~369KB 512x240	**~12MB 1960x2048**	**~92MB 5000x6100**	**~302MB 7040x14300**
T_S	64	1764	14832	42460
T_P	7.500934	118.55	820.15	2020.82
T_{i+1}	32	891	7495	7435
S	8.53	14.88	18.08	21.01
η	8.89%	15.50%	18.84%	21.89%
ECP	1	1	1	1
f_p	89.21%	94.26%	95.46%	96.24%

Conclusion

- Two efficient design for JPEG encoding for 24-bit BMP images were presented
- Significant speedup in both parallel CPU and GPU execution
- CPU performed the proposed JPEG algorithm much better