

# Energy-Efficiency and Performance Comparison of Aerosol Optical Depth Retrieval on Distributed Embedded SoC Architectures

Dustin Feld, Jochen Garcke, Jia Liu, Eric Schricker, Thomas Soddemann, and Yong Xue

**Abstract** The *Aerosol Optical Depth* (AOD) is a significant optical property of aerosols and is applied to the atmospheric correction of remotely sensed surface features as well as for monitoring volcanic eruptions, forest fires, and air quality in general, as well as gathering data for climate predictions on the basis of observations from satellites. We have developed an AOD retrieval workflow for processing satellite data not only with ordinary CPUs but also with parallel processors and GPU accelerators in a distributed hardware environment. This workflow includes pre-processing procedures which are followed by the runtime dominating main retrieval method. In this paper, we investigate if and how the main retrieval method can accommodate recent upcoming embedded hardware architectures in the field of high performance computing. We analyze and confirm the achieved performance as well as energy efficiency with real-world data from the moderate-resolution imaging spectroradiometer (MODIS) and even compare the potential of those new architectures to today's commonly available HPC hardware. Due to the very low energy intake, such embedded hardware architectures provide a great chance for situations with strong energy constraints like the pre-processing of recorded data on board of satellites.

---

Dustin Feld · Eric Schricker · Thomas Soddemann  
Fraunhofer Institute for Algorithms and Scientific Computing SCAI, Schloss Birlinghoven, 53757  
Sankt Augustin, Germany  
e-mail: [dustin.feld@scai.fraunhofer.de](mailto:dustin.feld@scai.fraunhofer.de)

Jochen Garcke  
Fraunhofer Institute for Algorithms and Scientific Computing SCAI, Schloss Birlinghoven, 53757  
Sankt Augustin, Germany  
Institute for Numerical Simulation, University of Bonn, Wegelerstr. 6, 53115 Bonn, Germany

Jia Liu · Yong Xue  
Institute of Remote Sensing and Digital Earth, Beijing, China

## 1 Introduction

Atmospheric aerosols are liquid or solid particles suspended in the air from natural and anthropogenic origin. They scatter and absorb solar radiation, and, to a lesser extent, scatter, absorb and emit terrestrial radiation (*direct effects*). Additionally, aerosols acting as cloud condensation nuclei and ice nuclei are referred to as *indirect effects*. A consequence of the direct effect (caused by absorbing aerosols) which changes cloud properties is called the semi-direct effect [5]. Besides the fact that aerosols affect the air quality and human health [8], such aerosol effects are currently considered one of the largest uncertainties in global radiative forcing [11]. These aspects have made the characterization of atmospheric aerosols a great concern in recent years.

The Aerosol Optical Depth (AOD), a measure of light extinction by aerosols in the atmospheric column above the earth's surface, is a comprehensive variable to remotely assess the aerosol burden in the atmosphere [14]. AOD data can be used by applications like the atmospheric correction of remotely sensed surface features, monitoring of sources and sinks of aerosols, radiative transfer models etc.

Compared with ground measurements, satellite remote sensing provides an effective method for monitoring spatial distribution and temporal variation of aerosols. Many approaches have been developed for the retrieval of AOD using satellite remote sensing observations, including the use of advanced very-high-resolution radiometer (AVHRR), medium-resolution imaging spectrometer (MERIS), moderate-resolution imaging spectroradiometer (MODIS), multi-angle imaging spectroradiometer (MISR) and others [6].

A wide range of datasets have been published since the operation of MODIS sensor on TERRA and AQUA satellites. The broad swath of 2330 km enables MODIS to provide global coverage with near daily frequency [14]. AOD datasets derived from MODIS observations have been used to estimate surface particulate matter [13], construct global climatology [10] etc. One commonly used AOD dataset can be obtained from the *National Aeronautics and Space Administration* (NASA) based on the DarkTarget and DeepBlue method [2]. The work in this paper adopts the synergetic retrieval of aerosol properties (SRAP) algorithm developed by in [17] to generate AOD datasets over China.

## 2 Method

Although many approaches have been developed to retrieve AOD, it is still a difficult task to retrieve AOD over land because it is not easy to separate aerosols' signals from the land surface contributions. This paper takes the SRAP-MODIS algorithm as the study case. It has been developed to solve the aerosol retrieval problem over bright land surfaces. The algorithm utilizes the high-frequency multi-temporal and multi-spectral information from MODIS data aboard both TERRA and AQUA satellites to produce the AOD results.

**Table 1** Symbols and applied values used in the SRAP algorithm.

	Symbol	Implication
input	$A'_{i,j}$	apparent reflectance (reflectance on the top of atmosphere)
	$a$	$a = 2$ (cf. [12] and [16])
	$b_i$	$b_i = \sec\theta_i$ , with $\theta_i$ the solar zenith angle for two satellite observations
	$\varepsilon$	$\varepsilon = 0.1$
	$\lambda_i$	wavelengths for three visible bands
	$b'_i$	$b'_i = \sec\theta'_i$ , with $\theta'_i$ the sensor zenith angle for two satellite observations
output	$A_{i,j}$	ground surface reflectance
	$\beta_i$	Ångstrom's turbidity coefficient
	$\alpha$	wavelength exponent

The SRAP-MODIS algorithm is a simple but practical algorithm introduced in [16] on an operational bi-angle approach model for retrieving AOD and the earth surface reflectance [12]. More details can be found in [17].

In the algorithm, the ground surface reflectance  $A_{i,j}$  is expressed by

$$A_{i,j} = \frac{(aA'_{i,j} - b_i) + b_i(1 - A'_{i,j})e^{\varepsilon(b_i-a)(0.00879\lambda_j^{-4.09} + \beta_i\lambda_j^{-\alpha})b'_i}}{(aA'_{i,j} - b_i) + a(1 - A'_{i,j})e^{\varepsilon(b_i-a)(0.00879\lambda_j^{-4.09} + \beta_i\lambda_j^{-\alpha})b'_i}}, \quad (1)$$

where  $i = 1, 2$  represent the observations of TERRA MODIS and AQUA MODIS respectively, and  $j = 1, 2, 3$  stand for three visible spectral bands at central wavelengths of 470, 550, 660 nm.

The symbols in Eq. (1) include both, *known* variables extracted from MODIS hierarchical data format (HDF) information such as  $A'_{i,j}$  (the reflectance on the *Top Of Atmosphere* (TOA)) and *unknown* variables to be solved for, the symbols in Eq. (1) are listed and explained in Table 1.

By assumption, the ground surface bidirectional reflectance properties and aerosol types and properties do not change for two MODIS observations within short time intervals between TERRA and AQUA overpass. Thus, the wavelength exponent  $\alpha$  is invariant for two observations and three visible bands. Ångstrom's turbidity coefficient, which represents the concentration of aerosol particles, may change for two overpass TERRA and AQUA observations; hence we have  $\beta_1$  and  $\beta_2$  for two overpass times.

The ground surface reflectance  $A_{i,j}$  can be approximated by the variation in the wavelength and the variation in the geometry [1]. Under this assumption, the ratio of two views' ground surface reflectance  $K_j$  for the wavelength  $j$  can be formulated as

$$K_j = \frac{A_{1,j}}{A_{2,j}}, \quad (2)$$

where  $A_{1,j}$  and  $A_{2,j}$  are the surface reflectances for the TERRA MODIS and AQUA MODIS.

Since aerosol extinction decreases rapidly with the wavelength, the AOD at  $2.12 \mu\text{m}$  is very small compared to that at the visible spectra bands. The atmospheric contribution at  $2.12 \mu\text{m}$  is relatively small, hence  $K_{\lambda=2.12\mu\text{m}}$  can be approximated as the ratio between the reflectance on the TOA. Besides, since  $K_j$  is assumed independent of the wavelength, the value at  $2.12 \mu\text{m}$  can be used for the visible bands. Thus, Eq. (3) is serving as the constraint between two ground surface reflectances for each visible band:

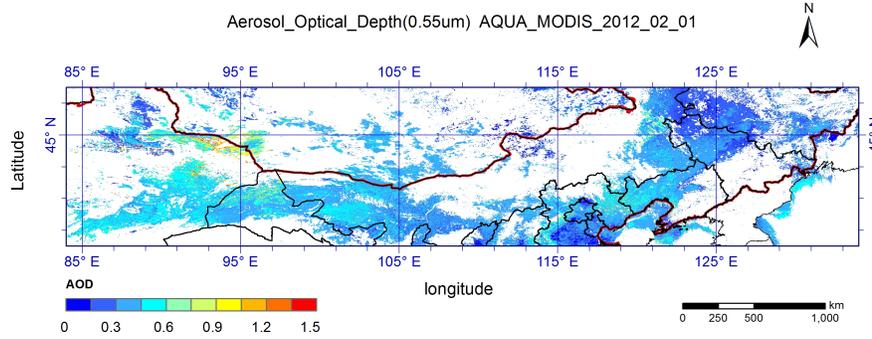
$$\frac{A_{1,j}}{A_{2,j}} = K_j = K_{\lambda=2.12\mu\text{m}} = \frac{A'_{1,\lambda=2.12\mu\text{m}}}{A'_{2,\lambda=2.12\mu\text{m}}}. \quad (3)$$

As a result, the unknown variables reduce to three, i.e. Ångström's turbidity coefficient  $\beta_i$  ( $i = 1, 2$ ) for two overpass observations and the wavelength exponent  $\alpha$ . Three equations containing three unknown variables to be solved are formed.

In our implementation, the three variables  $\beta_1$ ,  $\beta_2$  and  $\alpha$  from the obtained nonlinear equations are solved using the Broyden quasi-Newton method in the implementation from the 'Numerical Recipes in C' [9]. The known variables in Table 1 are taken as the input data for the method while it outputs the results for  $\beta_i$  and  $\alpha$ . These values are then used to calculate the resulting ground surface reflectances  $A_{i,j}$  and to further calculate the AOD  $\tau_A$  according to Ångström's turbidity formula

$$\tau_A = \beta_i \lambda_j^{-\alpha}. \quad (4)$$

Figure 1 shows an example of an AOD result map for the observed benchmark region.



**Fig. 1** AOD result map for AQUA MODIS at 550 nm band.

### 3 Implementation

For an appropriate AOD retrieval using the SRAP algorithm, pre-processing procedures such as cloud masking, absorption and geometric corrections, interpolations etc. need to be implemented and applied in addition to the final AOD model solving. We profiled the runtime of all serial procedures for the retrieval workflow on a workstation system in an earlier publication [4]. The profiling results showed that the final AOD model solving procedure takes up almost 50% of the total workflow runtime and, thus, is the most time-consuming part. It is therefore in the exclusive focus of the study in this paper. The pseudo code for the AOD model solving part is shown in Algorithm 1. It mainly consists of *data input and output (I/O)* procedures, i.e. the image data reading and writing from or to hard disk steps and the *computation* performing the solving of nonlinear equations and AOD calculation according to Ångström's turbidity formula addressed in Sect. 2. The reading and writing of image data from and to disc were implemented using the Geospatial Data Abstraction Library (GDAL) [15]. These functions are not parallelized and are therefore executed on a single core of the host CPU for all versions of the implementation. The computation procedure has a pixel-based nature in the operations without communication across the pixels. Thus, the solving of nonlinear equations and the AOD calculation for each pixel can be assigned to an individual parallel thread without any explicit or implicit synchronization.

For multi-core processors, the loop in Algorithm 1 was parallelized using an OpenMP directive `#pragma omp parallel for` with the advanced scheduling strategy `schedule (dynamic)`. This choice of a dynamic scheduling strategy is crucial for a good resulting performance as the time needed for the individual pixels' calculation varies depending on the input vector. This is based on varying convergence speeds for different inputs on the one hand and on the fact that different pixels follow different control flows in the calculation kernel on the other hand. E.g. the calculation for a pixel over sea finishes way faster than one over land. If a static scheduling is applied, coherent regions over sea are typically mapped to the same core and this core would therefore become idle long before other 'over-land' cores finish their calculation. As the calculation time for each pixel is relatively long, the overhead for dynamic scheduling pays off. Alternatively, a static scheduling with a chunk-size of 1 behaves similar. This influence was intensively studied and verified in an earlier publication [3].

For the GPU implementation, the Compute Unified Device Architecture C (*CUDA-C*) was used. A respective AOD kernel was designed and implemented so that each thread corresponds to the calculation for one pixel. The pixels are distributed on a *grid-block* of parallel threads. After reading the images from disc, the input images' data are copied from the CPU host's main memory to the GPU device's global memory at once, and, after computation of the AOD result, data are copied back to the CPU host and written to output files on disc. The thread-block size was configured to  $8 \times 8$  parallel threads per thread-block and the dimensions of the grid of thread-blocks were set dynamically corresponding to the image size to cover the whole image with thread-blocks.

---

**Algorithm 1:** Pseudo code for the AOD model solving.

---

**Data:** corrected reflectance, sensor zenith, solar zenith, initial values  
**Result:** AOD results

read-in the scene’s image data (with GDAL)  
 create an image representation of each spectral band

**if calculate DISTRIBUTED then**  
 | split the scene (each representation) in #nodes blocks  
 | distribute the data via message passing (MPI)

**OnEach Node do in parallel**  
 | **if use CUDA GPU then**  
 | | copy image representations to the accelerator’s memory  
 | **for each pixel  $p$  in the scene do in parallel**  
 | | use  $\mathbf{x}_p$  to solve (1) with (3) for  
 | |  $\alpha$  (wavelength exponent) and  
 | |  $\beta$  (Ångstrom’s turbidity coefficient)  
 | | calculate (4)  
 | |  $\tau_A$  (AOD according to Ångstrom’s turbidity formula)  
 | **if use CUDA GPU then**  
 | | copy back results from the accelerator’s memory

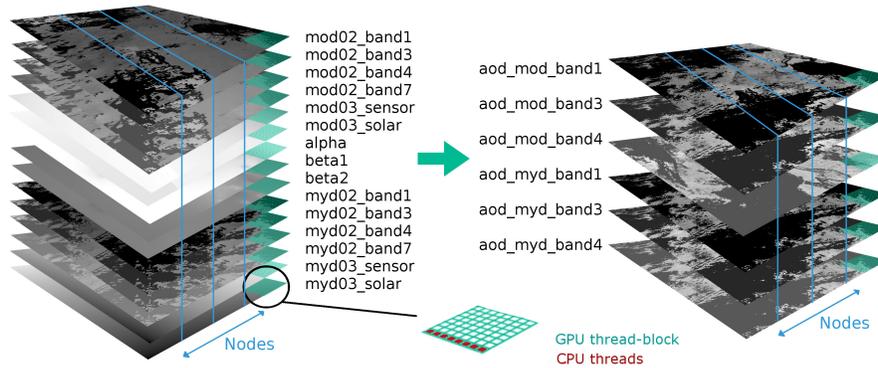
**if calculate DISTRIBUTED then**  
 | collect results from all nodes  
 write resulting images to files

---

For distributed computing, the Message Passing Interface (*MPI*) was used to combine the computing power of multiple nodes. The input images were uniformly split into approximately identically sized pieces according to the number of nodes. Each resulting part of the input image is distributed to its node and AOD is computed there either on the multi-core processors or the GPU. Because of the static splitting, the workload on the different nodes may be unbalanced but due to the relatively slow interconnect with gigabit ethernet, a splitting into smaller portions combined with a dynamic distribution on request is not investigated for this architecture. It could nevertheless improve the multi-node performance furthermore. After the computations on all nodes have finished, the AOD results are finally gathered from all nodes and combined. Algorithm 1 and Fig. 2 depict the overall parallel procedure of the AOD retrieval ( $\mathbf{x}_p$  represents the vector containing observations for the different spectral bands for pixel  $p$ ).

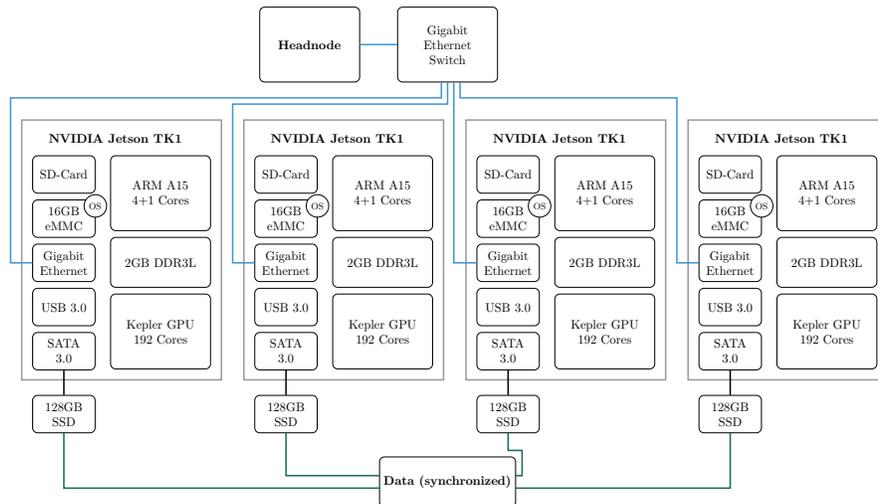
## 4 Embedded Low-Energy System

We set up an embedded system combined of four NVIDIA Tegra K1 (*TK1*) boards as shown in Fig. 3. These boards are designed to provide relatively high computing power especially for GPU applications while consuming only a relatively low amount



**Fig. 2** Multi-level parallelization on the spectral bands' data hypercube.

of energy. They are primarily targeted for mobile applications. The TK1 builds on the same NVIDIA Kepler architecture that drives high performance graphic compute units like the Tesla K20, but with much fewer cores (192 compared to 2496), slower and less memory (2 GB *DDR3* memory shared with the CPU compared to 5 GB of exclusive *GDDR5* memory) and other restrictions. Each board is driven by a 4 + 1 ARM Cortex A15 with four fast cores and one ultra low-energy core. As a consequence, a whole TK1 board has a power intake of not more than 10 watts while a high performance graphics card may take in up to 225 watts. In the middle of such 'extreme' GPUs, there is a wide variety of CUDA capable GPU accelerators, like the NVIDIA GTX 680 we use in our benchmarks. This card contains 1536 CUDA cores, 2 GB exclusive *GDDR5* memory and consumes up to 200 watts.



**Fig. 3** Jetson TK1 cluster configuration.

Undoubtedly, such high performance GPU devices like the K20 provide computational power that is of a higher order of magnitude (3.52 Tflops [peak single-precision]) than the TK1's GPU chip (326 Gflops [peak single-precision])<sup>1</sup>. As well, a high performance Intel Xeon CPU computes way faster than the equipped ARM cores on the TK1.

Certainly, the best performing hardware solution with a fast CPU and a fast GPU composed of the aforementioned hardware would be a system combining Intel Xeon CPUs with NVIDIA Tesla GPUs. If hard energy constraints similar to automotive or on-board satellite situations exist, a low-energy constellation like on the TK1 may be chosen. The question to be answered in this paper is, what to choose if the *energy efficiency* of such different configurations is the criterion of choice. This can be formalized by the  $\frac{\text{pixels}}{\text{joule}} = \frac{\text{pixels}}{\text{wattsecond}}$  ratio. Real-time computations producing new input data every  $M$  minutes combined with the demand to save energy are typical scenarios. If new data arrives every  $M$  minutes, a solution with lower performance and lower energy consumption may be preferable to high performance solutions as long as it processes the data set in less than  $M$  minutes to meet the time constraint.

In the following benchmarks, each hardware constellation is used to compute the AOD for the same earth region so that the number of pixels is constant. Therefore, the overall energy consumption can be used to compare the energy efficiency of different hardware constellations.

## 5 Benchmarks

The benchmarks' input data are extracted and pre-processed from MODIS HDF data, which can be downloaded from the *Level-1 and Atmosphere Archive and Distribution System* (LAADS Web) [7]. The data was selected randomly from February 1, 2012 and it covers 84°E-134°E, 38°N-48°N, which corresponds to 5000 × 1000 pixels. The spatial resolution of each pixel is 1km. In the following benchmarks, calculations are performed with single-precision floating point data.

### 5.1 Benchmark Environment

We used two benchmark systems in our experiments: the aforementioned low-energy system based on four NVIDIA Tegra K1 boards from Sect. 4, each equipped with an ARM 4(+1) quad-core CPU and an NVIDIA Kepler GPU, and a workstation equipped with an Intel Xeon E3-1275 V2 CPU and an NVIDIA GTX 680 GPU. Details are shown in Table 2.

First we compare the performance of the two hardware systems and their different processor types in Sect. 5.2, second we show how they behave in terms of energy

---

<sup>1</sup> <http://www.nvidia.com>

**Table 2** Benchmark system configurations.

		CPU	GPU
WORKSTATION (Scientific Linux 6.6)	<b>Type</b>	Intel® Xeon® E3-1275 V2	NVIDIA® GTX 680
	<b>#Cores</b>	4 (8 with HT)	1536
	<b>Clock-Speed</b>	3,5 GHz	1058 MHz
	<b>Memory</b>	8 GB DDR3	2 GB GDDR5
TEGRA CLUSTER (Gentoo Linux)	<b>Type</b>	ARM Cortex A15	NVIDIA® Kepler™
	<b>#Cores</b>	4(+1) quad-core	192 CUDA cores
	<b>Clock-Speed</b>	up to 2.3 GHz	852 MHz
	<b>Memory</b>	2 GB DDR3 (shared memory space)	

**Table 3** Calculation runtime, HPCo≐High Power Core, LPCo≐Low Power Core, T≐Threads.

Runtime	1xSoC						4xSoC		XeonWS	
	CPU			GPU			MPI	CPU		GPU
	1 LPCo	1 HPCo	4 HPCo	1 LPCo	1 HPCo	1 HPCo+GPU	1T	4T		
Calculation	2914.83	2861.37	717.37	47.14	46.93	18.13	192.15	49.28	3.92	
I/O	4.78	4.08	3.94	4.87	4.24	4.23	1.81	1.44	1.44	
Overall	2919.62	2865.45	721.31	52.01	51.16	22.36	193.97	50.72	5.36	

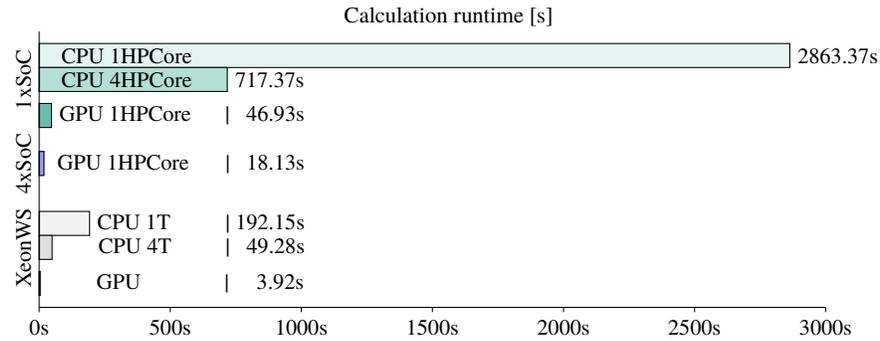
consumption and efficiency in Sect. 5.3. As the TK1 contains not only the quad-core CPU and a GPU but also an additional low power core, we as well analyze the potential of this core in both counts (energy consumption and efficiency). Regarding this, the four powerful cores are named *HPCores*, the additional low power core is named *LPCore*. The TK1 ‘*System on Chip*’ boards are also referred to as *SoC* and the workstation as *XeonWS*.

All runs were performed 10 times, the maximal and minimal runtime and energy values were neglected and the average of the remaining 8 runs is reported. The relative standard deviation of the runtimes of all single node runs was below 1 % in our benchmarks while the one of the multi-node runs with four TK1 boards was still below 2 %. The respective ranges of power intake are shown by the shaded areas in Figures 7-14.

Whenever only one TK1 system is used, the results are reported in green, if more than one TK1 boards are used, it is shown in purple and for the Xeon workstation in gray.

## 5.2 Performance Benchmarks

Table 3 contains the calculation runtimes for all constellations while Fig. 4 only shows the most relevant times as a bar chart. The results for one board (1xSoC) reveal that using one of the faster four cores instead of the low energy core improves the runtime by only about two percent. Using all four fast cores with multi-threading instead of one of those leads to a near-ideal overall speedup of 3.97x. Executing the code on

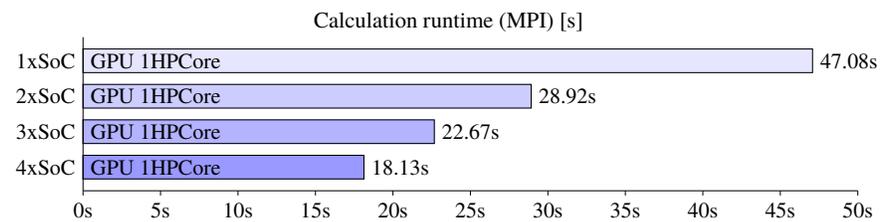


**Fig. 4** Calculation runtime, HPCore $\hat{=}$ High Power Core, T $\hat{=}$ Threads.

one SoC on the GPU, while the serial I/O is performed on one of the faster ARM CPU cores, is 14 $\times$  faster than on the four main ARM cores. Activating the low power core instead of a normal one leads to a 15% performance loss in the CPU based I/O routines while it does not significantly influence the GPU based calculation time. Therefore, the best performing single SoC configuration is to use one of the faster ARM cores for the I/O routines along with the GPU for the calculation part.

Distributing the work on all four boards with this configuration leads to an additional speedup of 2.3 $\times$ . Figure 5 shows how the runtime decreases the more boards are used to process the whole work. It as well shows how the speedup stagnates from 1.63 $\times$  when taking two boards, 2.07 $\times$  for three and, finally, 2.6 $\times$  for all four boards. This confirms that the interconnect via gigabit ethernet, which is used for the MPI communication, can become a de facto bottleneck in such a system configuration.

Comparing those runtimes to the workstation shows that multi-threading on the Xeon CPU as well scales near-optimal with a speedup of 3.82 $\times$  on four cores and that the Xeon CPU performs about 15 $\times$  better than one ARM core. The resulting performance on all Xeon cores is comparable to the performance of one TK1 with the GPU. The GPU in the workstation expectedly reaches the best overall performance.



**Fig. 5** Calculation runtime (MPI), HPCore $\hat{=}$ High Power Core, T $\hat{=}$ Threads.

### 5.3 Energy Benchmarks

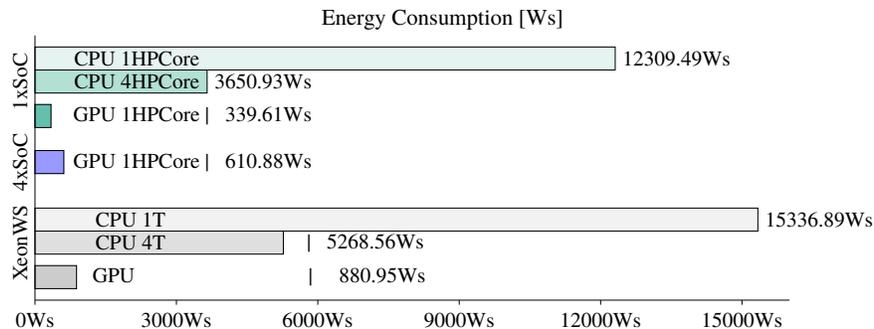
Table 4 contains the respective energy consumptions for all configurations. The power intake was measured with an ISO certified digital multimeter 'Voltcraft VC870' at the maximal clock-rate of one measurement per second. This introduces an inaccuracy for small runtimes that should be equalized by averaging the consumption of multiple repeated runs. As in the previous section, Fig. 6 shows the most relevant results as a bar chart. As the system allows it, we deactivated all cores that are not used in any of the TK1 measurements.

**Table 4** Energy consumption, HPCo≐High Power Core, LPCo≐Low Power Core, T≐Threads.

Power/Energy	1xSoC					4xSoC
	CPU			GPU		MPI
	1 LPCo	1 HPCo	4 HPCo	1 LPCo	1 HPCo	1 HPCo+GPU
∅ Power Intake [W]	4.14	4.30	5.06	6.48	6.64	27.32
Energy consumption [Ws]	12086.11	12309.49	3650.93	337.20	339.61	610.88
<i>pixels/Ws</i>	414	406	1370	14828	14723	8185

Power/Energy	XeonWS		
	CPU		GPU
	1T	4T	
∅ Power Intake [W]	79.07	103.88	164.36
Energy Consumption [Ws]	15336.89	5268.56	880.95
<i>pixels/Ws</i>	326	949	5676



**Fig. 6** Energy consumption, HPCore≐High Power Core, T≐Threads.

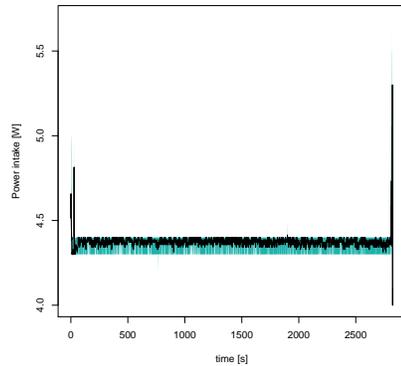
Executing the code on four cores instead of only one on the CPU consumes only about 3650.93 Ws compared to 12309.49 Ws and is therefore much more energy efficient as the four core version is 3.97x faster and only has a 18% higher average power intake (~5.1 W cp. to 4.3 W - see Fig. 7 and 8). This is based on a relatively high power intake for all the periphery in the idle system. The best energy efficiency on one TK1 board is reached when using the GPU.

As shown in Sect. 5.2, exhausting the compute power of all four TK1 boards leads to the best TK1 performance but not with an ideal scaling due to the inter-board communication via ethernet. As the power intake grows linearly when using multiple identical boards simultaneously (plus some extra power for the ethernet switch), using multiple boards does not improve the energy efficiency.

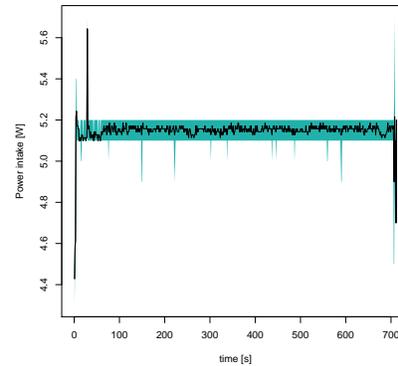
Comparing those results to the ones of the workstation, the TK1 system is for every configuration (single-core CPU, multi-core CPU, GPU) more energy efficient than its pendant on the workstation. Nevertheless, the high power intake of the workstation's components is almost equalized by the likewise faster computing. The biggest advantage in terms of energy efficiency can be seen in the GPU benchmarks. While the workstation's GPU consumes 880.95 Ws for the whole calculation, a single TK1 board settles this task with consuming only 339.61 Ws.

For better comparison, Table 4 additionally contains the  $\frac{\text{pixels}}{W_s}$  ratio which quantifies how many pixels can averagely be calculated with the energy of one wattsecond (resp. one joule).

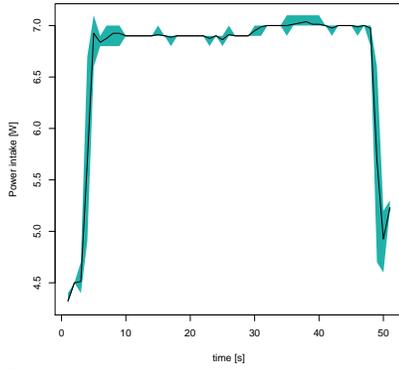
Figures 7-14 illustrate the power intake over time for the different runs. The shaded areas represent the range of variation in the repeated runs. It can be concluded that the power intake on the TK1s is generally very stable over time (cp. especially Fig. 14). On the workstation, there is more deviation in the intake, especially for the GPU runs, but this was probably caused by the very short runtimes with only few measurement points.



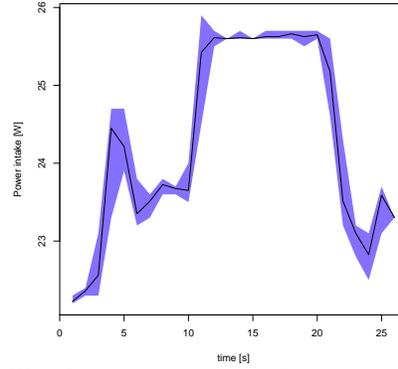
**Fig. 7** Power intake 1xSoC CPU 1HPCore.



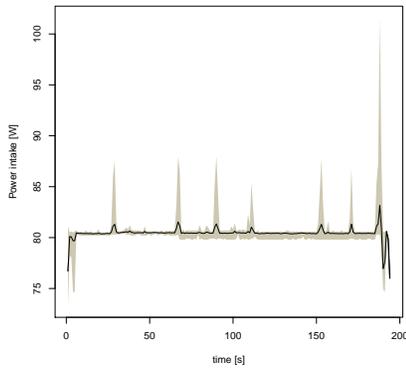
**Fig. 8** Power intake 1xSoC CPU 4HPCore.



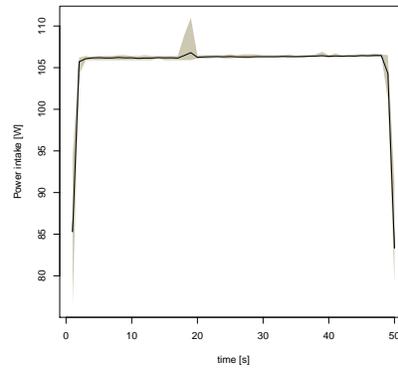
**Fig. 9** Power intake 1xSoC GPU 1HPCore.



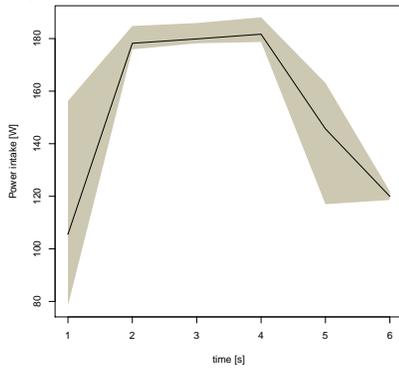
**Fig. 10** Power intake 4xSoC GPU 1HPCore.



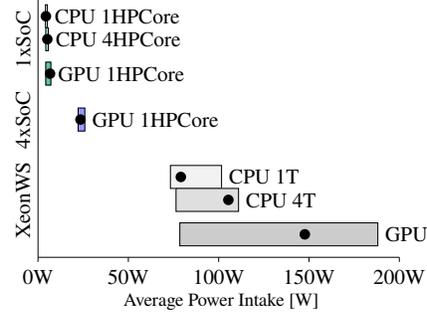
**Fig. 11** Power intake XeonWS CPU 1T.



**Fig. 12** Power intake XeonWS CPU 4T.



**Fig. 13** Power intake XeonWS GPU.



**Fig. 14** Power intake range ( $\bullet \hat{=}$  Average).

*The ranges of power intake among the 10 repeated runs are shown by the shaded areas.*

## 6 Discussion

Concerning the suitability for parallel execution and the applicability for different parallel architectures, the results from Sect. 5 show that the code scales very well in a multi-threaded shared memory environment and is exceptionally well suited for GPU computing. As a result, a GPU on a TK1 board can achieve a performance comparable to the Xeon CPU while consuming only a fraction of the energy. The difference in both, energy consumption and performance, of the low power ARM core on the TK1 to one of the ‘normal’ cores was rather small in our benchmarks. Due to the existent but not ideal multi-node scaling, the usage of multiple boards improves performance on the one hand but diminishes energy-efficiency on the other hand.

In an earlier publication [3], we investigated the performance and energy efficiency of the retrieval code in a server environment equipped with a high performance Kepler K20 GPU. The calculation of the same scene took 2.61 s on the GPU consuming 621.91 Ws (approximately 60% is consumed by the GPU, 40% by the remaining system) while processing 8040 pixels per Ws. Thus, the energy efficiency of this high-end server system is in between the one of the GPU in the benchmarked workstation system and the one on a TK1 board.

Which configuration to choose therefore strongly depends on the environmental constraints. The most energy efficient solution for executing the presented code is to use the GPU of one TK1 board. If the energy restrictions are a bit looser, a compound of several such boards can further improve the performance while keeping the energy intake relatively low. Such a system outperforms our Xeon CPU on four threads by a factor of  $2.7x$  while the Xeon system consumes more than  $8.6x$  times the energy. It has to be noted that the idling GPU power intake is included in such CPU measurements on the workstation.

When looking purely for highest computational performance with no energy constraints, the workstation GPU clearly wins.

Considering all benchmarks and both goals, performance and energy-efficiency, a very interesting workstation constellation for the future will be a low energy CPU along with a high-end GPU.

## 7 Outlook

In this paper, we investigated the potential of an embedded SoC architecture equipped with a multi-core ARM processor and an NVIDIA Kepler GPU in terms of both, computational performance and energy efficiency for the retrieval of AOD. The results show that embedded SoC boards like the NVIDIA Tegra K1 provide a relatively high computational power paired with a low power intake, especially if a code can make efficient use of the GPU.

In the future, we plan to port more codes that are used in real-time scenarios to such architectures to save energy in scenarios where either the energy constraints are

naturally tight or where the provided computational power of such systems is simply sufficient.

**Acknowledgements** This work was partially funded by the German Ministry for Education and Research (BMBF) under project grant 01|S13016A within the ITEA2-Project MACH.

## References

1. R. J. FLOWERDEW AND J. D. HAIGH, *An approximation to improve accuracy in the derivation of surface reflectances from multi-look satellite radiometers*, *Geophysical Research Letters*, 22 (1995), pp. 1693–1696.
2. R. C. LEVY, S. MATTOO, L. A. MUNCHAK, ET AL., *The Collection 6 MODIS aerosol products over land and ocean*, *Atmospheric Measurement Techniques*, 6 (2013), pp. 2989–3034.
3. J. LIU, D. FELD, Y. XUE, J. GARCKE, AND T. SODDEMANN, *Multicore processors and graphics processing unit accelerators for parallel retrieval of aerosol optical depth from satellite data: Implementation, performance, and energy efficiency*, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8 (2015), pp. 2306–2317.
4. J. LIU, D. FELD, Y. XUE, ET AL., *An efficient geosciences workflow on multi-core processors and GPUs: A case study for aerosol optical depth retrieval from MODIS satellite data*, *International Journal of Digital Earth*, 9 (2016), pp. 748–765.
5. U. LOHMANN AND J. FEICHTER, *Global indirect aerosol effects: A review*, *Atmospheric Chemistry and Physics*, 5 (2005), pp. 715–737.
6. L. MEI, Y. XUE, H. XU, ET AL., *Validation and analysis of aerosol optical thickness retrieval over land*, *International Journal of Remote Sensing*, 33 (2012), pp. 781–803.
7. NASA, *Level 1 and Atmosphere Archive and Distribution System (LAADS Web)*. Website, 2015. Online at <http://ladsweb.nascom.nasa.gov/>, visited 2015-10-20.
8. U. PÖSCHL, *Atmospheric aerosols: Composition, transformation, climate and health effects*, *Angewandte Chemie International Edition*, 44 (2005), pp. 7520–7540.
9. W. H. PRESS, S. A. TEUKOLSKY, W. T. VETTERLING, AND B. P. FLANNERY, *Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing*, Cambridge University Press, New York, NY, USA, 1992.
10. L. A. REMER, R. G. KLEIDMAN, R. C. LEVY, ET AL., *Global aerosol climatology from the MODIS satellite sensors*, *Journal of Geophysical Research: Atmospheres*, 113 (2008). D14S07.
11. T. F. STOCKER, D. QIN, G.-K. PLATTNER, ET AL., *Technical summary*, in *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, T. F. Stocker, D. Qin, G.-K. Plattner, et al., eds., Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2013, book section TS, pp. 33–115.
12. J. TANG, Y. XUE, T. YU, AND Y. GUAN, *Aerosol optical thickness determination by exploiting the synergy of TERRA and AQUA MODIS*, *Remote Sensing of Environment*, 94 (2005), pp. 327–334.
13. T.-C. TSAI, Y.-J. JENG, D. A. CHU, J.-P. CHEN, AND S.-C. CHANG, *Analysis of the relationship between MODIS aerosol optical depth and particulate matter from 2006 to 2008*, *Atmospheric Environment*, 45 (2011), pp. 4777–4788.
14. Y. WANG, Y. XUE, Y. LI, ET AL., *Prior knowledge-supported aerosol optical depth retrieval over land surfaces at 500m spatial resolution with MODIS data*, *International Journal of Remote Sensing*, 33 (2012), pp. 674–691.
15. F. WARMERDAM, *GDAL - Geospatial Data Abstraction Library*. Website, 2015. Online at <http://www.gdal.org/>, visited 2015-10-20.
16. Y. XUE AND A. P. CRACKNELL, *Operational bi-angle approach to retrieve the earth surface albedo from AVHRR data in the visible band*, *International Journal of Remote Sensing*, 16 (1995), pp. 417–429.

17. Y. XUE, X. HE, H. XU, ET AL., *China Collection 2.0: The aerosol optical depth dataset from the synergetic retrieval of aerosol properties algorithm*, Atmospheric Environment, 95 (2014), pp. 45–58.