 doi:10.20944/preprints202103.0558.v1

Article

Parallel simulation of a multi-span DWDM system limited by FWM

Rafael Sanchez-Lara [1], Jose Luis Lopez-Martinez [2], Joel Antonio Trejo-Sanchez [3],Herman Leonard Offerhaus [4] and Jose Alfredo Alvarez-Chavez [4]*

[1] Universidad Autonoma del Carmen (UNACAR), Facultad de Ingenieria, C.56, No. 4, C.P. 24180, Cd. Del Carmen, Campeche, Mexico; rsanchez@pampano.unacar.mx
[2] Facultad de Matematicas, Universidad Autonoma de Yucatan,Merida, Mexico; jose.lopez@correo.uady.mx
[3] CONACYT-Centro de Investigación en Matemáticas; joel.trejo@cimat.mx
[4] Optical Sciences Group, University of Twente, Enschede, The Netherlands;h.l.offerhaus@utwente.nl; j.a.alvarezchavez@utwente.nl
* Correspondence: j.a.alvarezchavez@utwente.nl

Abstract: One of the non-linear phenomena affecting high bandwidth and long reach communication systems is Four-Wave Mixing (FWM). Unfortunately, the simulation of systems for parameter optimization requires more time as the number of channels increases. In this paper, we propose a new high-performance computational model to obtain the optimal design parameters in a multi-span Dense Wavelength Division Multiplexing (DWDM) system, limited by FWM and the intrinsic Amplified Spontaneous Emission (ASE) noise of optical amplifiers employed in each segment. The simulation in this work provides a complete optical design characterization and compares the efficiency and speed improvement of the proposed parallelization model versus an earlier sequential model. Additionally, an analysis of the computational complexity of parallel models is presented, in which two parallel implementations are used; firstly, Open MultiProcessing (OpenMP) which is based on the use of a central multi-core processing unit and secondly, the Compute Unified Device Architecture (CUDA), which is based on the use of a Graphics Processing Unit (GPU). Results show that parallelism improves by up to 40 times the performance of a simulation when nested parallelization with CUDA is used over a sequential method and up to 6 times compared with the implementation with OpenMP using 12 processors. Within our parallel implementation, it is possible to simulate with an increased number of channels that which was impractical in the sequential simulation.

Keywords: Computer Simulation; DWDM; FWM; GPU programming; Parallel Architectures; Parallel Programming

1. Introduction

Non-linear phenomena occur in optical fibre systems especially when they are used at their maximum . It is a limitation for multi-segment, high-bandwidth and long-range Dense Wavelength Division Multiplexing (DWDM) systems. DWDM is used to maximize the bandwidth capabilities of optical fibres where their segments are connected by optical repeaters. Such communication systems are found in land or transoceanic communication systems. Five non-linear effects that play a role in such systems are known so far: Four-Wave Mixing (FWM), Stimulated Raman Scattering (SRS), Stimulated Brillouin Scattering (SBS), Self-Phase Modulation (SPM) and Cross-Phase Modulation (XPM); each appears under certain design circumstances. In any design analysis, basic linear effects will also be present such as fibre attenuation and dispersion, defined by the parameter a and D, respectively. Erbium Doped Fiber Amplifiers (EDFAs) have been used to compensate for fibre losses and to increase the transmission distance, causing at the same time an increase on amplified spontaneous emission (ASE) noise and nonlinearities [1,2].Different kinds of optical fibres have been designed to reduce the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2021 doi:10.20944/preprints202103.0558.v1

effect of dispersion and nonlinear effects [3–6]. There are numerous studies on non-linear impairments showing the importance and complexity of such phenomena for DWDM [7–12].

The sequential model presented in [13] can be used on a multi-span DWDM system limited by FWM/ASE to obtain the optimal power (P_{opt}) and maximum link length (L_{max}) with respect to the dispersion D. The model is highly parallelizable so that different parallelization models can be used to reduce the computational complexity. Parallelization can significantly reduce the processing or execution time of a sequential algorithm [14],including the simulation of Non-Linear phenomena [15]. To implement parallelization, several tools exist, such as MPI (Message Passing Interface) [16], OpenMP[17], and recently GPU cards [18]. The latter are programmable using the libraries provided by NVIDIA, in particular the CUDA-C language. Other examples using GPU cards are mentioned in [19,20], where the use of this tool significantly improves the execution of algorithms.

In this work we use two schemes to significantly reduce the time complexity of DWDM and simulate more channels than obtained in [13].First, we show a multiprocessing parallel paradigm using OpenMP since it only uses the processors located in the Central Processing Unit (CPU) and does not require a distributed system like MPI. The efficiency has been proven [21], but it is limited by the number the cores in the processor. Secondly,we use of Graphic Processing Units for the cost-benefit ratio [22–24].For CUDA, we use Dynamic Parallelism (DM) [22]. Dynamic parallelism consists of hierarchical kernels in a tree structure, where a thread is defined as a father kernel. This father kernel can generate new child kernels to execute tasks in the GPU context. The height of the tree depends on the characteristics of the GPU card. Finally, an analysis of both implementations using metrics such as speedup, efficiency and performance ratio is presented.

The paper is divided into the following sections: Section 2 describes the theory of a multi-span DWDM system limited by FWM; Section 3 describes the proposed sequential model of computation; in Section 4 architectural models are explained, in Section 5, experimental, and simulated results are presented in terms of execution time, efficiency, performance ratio, and speedup. Finally concluding remarks are presented in Section 6.

2. Theoretical Analysis

The power dependence of the refractive index is denoted by χ^3 and FWM, is described using χ^3. If three optical fields with carrier frequencies f_1,f_2, and f_3 copropagate inside the fiber simultaneously, FWM generates a fourth field whose frequency f_4 is related to other frequencies by a relation $f_4 = f_1 \pm f_2 \pm f_3$. All frequencies corresponding to different plus and minus sign combinations are possible. In practice, most do not grow to significance because of a phase-mis-matching. Frequency combinations of the form $f_4 = f_1 + f_2 - f_3$ are troublesome for multichannel communication systems since they can become nearly phase-matched when channel wavelengths lie close to the zero-dispersion wavelength [1].

A multi-span DWDM system refers to the multiplexing of multiple information frequency channels $(f_1, f_2, \ldots, f_N)$ in a single fibre with the aim of transmitting the largest possible bandwidth. To achieve long ranges, the optical fibre is segmented and connected by optical amplifiers as shown in Figure 1. The operating bandwidth of the amplifiers must match the spectral bandwidth of the combined channels to be transmitted. The C band is the most commonly used for its low lossesand the EDFA fits this system [2].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2021 doi:10.20944/preprints202103.0558.v1

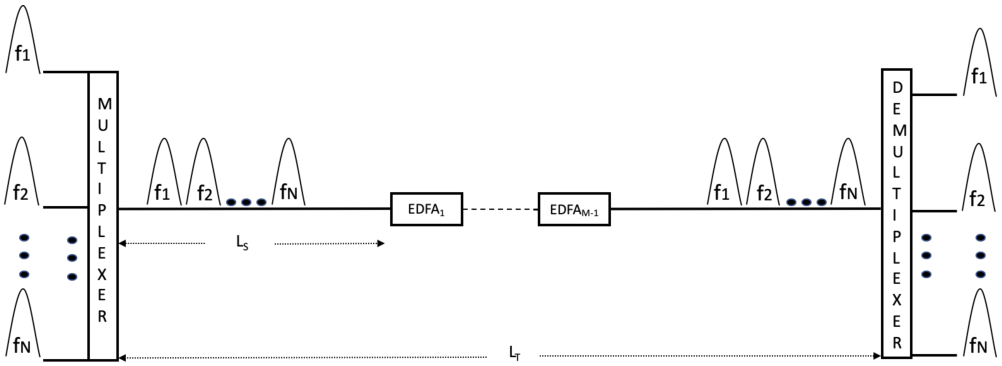

Figure 1. Multi-span DWDM System setup

Figure 1 represents the multi-span DWDM system setup. f_i represents the $i-th$ frequency channel, L_S is the length of a segment or the space between amplifiers (typical values between 50 km and 100 km), N is the total number of channels, L_T is the total length of the system and M is the total number of segments.

The total spectral width in band C is $W = f_N - f_1 = 3.75$ THz $(30nm)$. The multi-channel density Δf is a function of the total number of N channels and can be expressed as:

$$\Delta f = f_i - f_{i-1} = \frac{W}{N-1}. \tag{1}$$

The central frequency can be expressed as:

$$f = \frac{f_1 + f_N}{2} = 193THz[1550nm]. \tag{2}$$

An FWM signal appears by mixing three frequency channels f_i, f_j and f_k, so that is, $f_n = f_i + f_j - f_k (i, j \neq k)$. The power at f_n is given by [11,12,25]:

$$P_{i,j,k} = k^2 P_i P_j P_k \eta_{i,j,k} d^2_{i,j,k} (\frac{ML_{eff}}{A_{eff}})^2 e^{-aL_S}, \tag{3}$$

for $k = \frac{32\pi^3\chi^3}{\eta^2 c\lambda}$, where λ is wavelength, c is the vacuum light speed, η is the core refractive index, a is the linear loss coefficient,P_i, P_j, P_k are launched input power levels per channel, $L_{eff} = (1 - exp(-aL_S)/a$ is the fibre effective length, A_{eff} is the effective area of the guided mode, d is the degeneracy factor ($d = 3$ for $i = j, d = 6$ in case $i \neq j$), χ^3 is the third-order nonlinear electric susceptibility $(6 \times 10^{-14} m^3 W^{-1})$, and $\eta_{i,j,k}$ is the FWM efficiency, which is given by:

$$\eta_{i,j,k} = (\frac{a^2}{a^2 + \Delta\beta^2_{i,j,k}})(1 + \frac{4e^{-aL_S}}{(1 - e^{-aL_S})^2}) \sin^2(\frac{\Delta\beta_{i,j,k} L_S}{2}), \tag{4}$$

where $\Delta\beta_{i,j,k}$ is the phase mismatch that is expressed as:

$$\begin{aligned} \Delta\beta_{i,j,k} = (\frac{2\pi\lambda^2}{c})(|f_i - f_k||f_j - f_k|) \\ (D + \frac{dD}{d\lambda}((f_i - f_k) + (f_j - f_k))). \end{aligned} \tag{5}$$

D is the fibre chromatic dispersion coefficient. In a DWDM system with N number of channels and with equal channel spacing, the total FWM power at the frequency f_n or in other words, the total FWM power at the aforementioned frequency can be expressed as follows:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2021 doi:10.20944/preprints202103.0558.v1

$$P_{..} = \sum_{f_j - f_k} P_{i,j,k} \cdot {}^{aL_a}\left(\frac{ML_{eff}}{A_{eff}}\right)^2 \sum_{f_n = f_i + f_j - f_k} \eta_{i,j,k} d^2_{i,j,k}. \tag{6}$$

This include the sum of all cases that satisfy $f_n = f_i + f_j - f_k$, $(i, j \neq k)$. For simplicity, we consider equal transmission power in each channel ($P_S = P_i = P_j = P_k$), therefore, the most affected channel with the highest FWM, ($n = w$) power can be expressed as:

$$P_w = k^2 P_S^3 \left(\frac{ML_{eff}}{A_{eff}}\right)^2 Y e^{-aL_S}, \tag{7}$$

where Y indicates the maximum value of the summation $\sum \eta_{i,j,k} d^2_{i,j,k}$ for average channel as follows:

$$Y = \max\left(\sum \eta_{i,j,k} d^2_{i,j,k}\right), \quad i, j, k = 1, 2, \ldots, N. \tag{8}$$

FWM can be considered as an interfering signal, so if we consider a typical maximum value of 20 dB of optical signal to interference ratio in the receiver [1], the maximum power for the most interfered channel, is:

$$P_m = \frac{A_{eff}}{10kML_{eff}Y^{1/2}}. \tag{9}$$

This should be compared to the total ASE noise power in the receiver can be estimated, which increases with the number of amplifiers and is as [26] :

$$P_{sp} = 2\eta_{sp}(G - 1)hfB_0M, \tag{10}$$

where h is Plank's constant $(6.634 \times 10^{-34} J.s)$, f is the centre frequency, G is the gain, B_0 is the bandwidth of the optical filter that can be approximated by $2B$, $2\eta_{sp}$ is the amplifier noise factor whose minimum value is 2 for an ideal amplifier and η_{sp} is the population inversion parameter [1]. For equal amplifier spacing $(M = L/L_a)$ and an amplifier gain G that compensates for fibre loss as:

$$G = e^{aL_S}, \tag{11}$$

The minimum power per channel to ensure the required SNR $(20dB)$ can be obtained as:

$$P = 200\eta_{sp}(G - 1)hfB_0L/La. \tag{12}$$

The FWM effect imposes an upper limit on the power per channel and in the ASE case, it imposes a lower limit when the transmission distance is increased. Therefore, by intersecting P and P_m the maximum value of the transmission distance $L = L_{max\_fwm}$ is obtained and the optimal transmission power per channel $P_{o\_fwm} = P = P_m$ can be expressed in the following ways L_{max_fwm} :

$$= \left\{\frac{A_{eff}L_a^2}{200hf\eta_{sp}B_0(Ee^{aL_a} - 1)L_{eff}10kY^{1/2}}\right\}^{1/2}, \tag{13}$$

and P_{o_fwm}

$$= \left\{\frac{200hf\eta_{sp}B_0(Ee^{aL_a} - 1)}{L_{eff}10kY^{1/2}}\right\}^{1/2}. \tag{14}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2021 doi:10.20944/preprints202103.0558.v1

Equations (13) and (14) only represent a system limited by FWM-ASE noise. The values of equation (14) must be lower than the power thresholds of the non-linear phenomena of Raman [27] and Brillouin[28–31] stimulated scattering.

By analyzing equations (13), (15) and (16), where equation (15) represents the maximum transmission distance due to Dispersion and equation (16) is the maximum transmission distance due to the SRS-ASE effect [32,33], we can obtain the equation (17) for D as a function of N, to limit the system for only FWM-ASE effect, without the need to analyze both non-linearities [34] so:

$$L_{max\_D} = \frac{C}{2B^2\lambda^2|D|}, \tag{15}$$

$$L_{max\_srs} = \frac{210^5 L_S}{\sqrt{hfn_{sp}B_0N(N-1)(Ee^{aL_S}-1)\Delta f L_{eff}}}, \tag{16}$$

$$D < 8.9 \times 10^{-5}N^2 + 0.011N + 1.1 \tag{17}$$

Obtaining the graphs for $L_{max\_fwm}(D)$ and $P_{o\_fwm}(D)$, by using equations (13), (14) and (17) is time consuming. In Section 3 we describe the main difficulties in obtaining these values using the sequential model. We show that the execution time in the sequential model is $O(N^5)$. In Section 4 we present two parallel architectures, and we present the improvement in time by using such architectures by computing the values for $L_{max\_fwm}(D)$ and $P_{o\_fwm}(D)$ concurrently.

3. Sequential model

Pseudocode 1 presents the sequential algorithm to compute $[L_{max\_fwm}]$ and $[P_{o\_fwm}]$. The algorithm searches for the optimal values among all intervals and channels satisfying the relation $f_n = f_i + f_j - f_k$, given the wavelength channels N and the value μ that defines the number of equally spaced intervals.Line 2 of the pseudocode computes the value of the D_{upper_limit} using Eq (17) followed by the set of intervals $D = \{d_i | d_1 = 0.0 \wedge d_\mu \leq D_{upper\_limit} \wedge |d_i - d_{i+1}| = \epsilon\}$ according to the number of required intervals μ. $|D|$ is the cardinality of set D.

To compare the performance and efficiency of our algorithm with respect to other existing methods [35] we compute the complexity of this method. The computational cost is dominated by the nested cycles between lines 4 through 15. Line 4 performs $|D|$ steps, and Line 5 performs N steps. Line 6 performs $O(N^3)$ steps due to the combination that is generated at the time of the formation of the set. The maximum cardinality of the set in Lines 7-11 is N^3. Therefore, the computational complexity in lines 6 - 12 will be $O(2N^3)$. However, due to the nesting of the cycles on Lines 4 and 5 that affects Lines 6-11, the complexity is given by $O(|D|N^4)$. Line 13 obtains the maximum value of a set of elements of $[P]$ that, in the worst case, will be of length N^3. Finally, the complexity of the Line 14 is constant. In total, the computational complexity of the proposed algorithm is given by $O(|D|N^4)$. Note that N and μ take the value of 240 and 144, respectively. Since $|D| \approx N/2$, the time complexity is $O(N^5)$.

4. Parallel algorithm ParDWDM

We use two algorithms to reduce the execution time. Firstly, we use the OpenMP API for Symmetric Multi-processors (SMP) architecture [36] and distribute the load of work among the cores of the processor. Secondly, we use the more sophisticated parallel approach based on the CUDA-C for GPU processing and distributed a more homogeneous workload between graphics processors.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2021 doi:10.20944/preprints202103.0558.v1

Pseudocode 1: Pseudocode for the sequential algorithm

Input : N,μ ,ϵ
Output: L_{max_fwm}, P_{o_fwm}

1 **begin**
2 Let Eq(24) to obtain D_{upper_limit} define the set of intervals of dispersion as
3 $D = \{d_i | d_1 = 0.0 \wedge d_\mu \leq D_{upper\_limit} \wedge |d_i - d_{i+1}| = \epsilon\}$, for $1 \leq i \leq \mu$. For $N = 240$ and $\mu = 144$ intervals in D, there exist $D_{upper\_limit} < 9$ and $\epsilon = 0.0625$;
4 **for** *each* d_i *in* D **do**
5 **for** *each* $0 < n \leq N$ **do**
6 Let $S = \{s_w = (f_i, f_j, f_k) \mid \forall i, j, k \epsilon \{1..N\}\}$
7 **for** *each* s_w *of* S **do**
8 **if** $(f_n == (f_i + f_j - f_k))$ **then**
9 Compute Eq.(6) and add to $[P]_n$;
10 **end**
11 **end**
12 **end**
13 Compute Y Eq.(8);
14 Let Eq.(13) and Eq.(14) to obtain $[L_{max\_fwm}]_i$ and $[P_{o\_fwm}]_i$ for d_i ;
15 **end**
16 **end**

4.1. OpenMP

OpenMP uses a master-slave architecture [16]. In a master-slave architecture, the master executes itself sequentially, and creates a set of threads (slaves) to execute the current assigned computation. The master processor ensures that the workload is distributed equitably among all the slave processors. Pseudocode 2 describes the main steps of the algorithm ParDWDM using OpenMP.

The master thread computes the set D from input ϵ into the main memory in the form of an array, and distributes the n channels to the individual threads for each d_i , as shown in Figure 2. The complexity of this algorithm is $O(N^4)$. Notice that Line 6 performs a combination that is not parallelizable due to the dependency on the generation of such sets and the existence of a loop-based iteration dependency[37] in Line 7. Therefore, the proposed parallel algorithm has a parallelism level where only a set of polynomial number of processing units perform a large amount of work. This type of parallelism is geared toward coarse-grained parallelism[37].

Pseudocode 2: Pseudocode for OpenMP

Input : N,μ,ϵ
Output: L_{max_fwm}, P_{o_fwm}

1 **begin**
2 Let Eq(24) to obtain D_{upper_limit}
3 **for** *each* d_i *in* D **do**
4 **#pragma omp parallel for**
5 **for** *each* $0 < n \leq N$ **do**
6 Let $S = \{s_w = (f_i, f_j, f_k) \mid \forall i, j, k \epsilon \{1..N\}\}$
7 **for** *each* s_w *of* S **do**
8 **if** $(f_n == (f_i + f_j - f_k))$ **then**
9 Compute Eq.(6) and add to $[P]_n$;
10 **end**
11 **end**
12 **end**
13 Compute Y Eq.(8);
14 Let Eq.(13) and Eq.(14) to obtain $[L_{max\_fwm}]_i$ and $[P_{o\_fwm}]_i$ for d_i ;
15 **end**
16 **end**

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2021 doi:10.20944/preprints202103.0558.v1

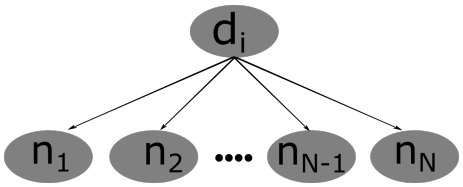

Figure 2. A parallel architecture for ParDWDM Algorithm with one parallelism level

To improve the perfomance of the simulation we use dynamic parallelism using GPUs.

4.2. Dynamic paralellism using CUDA

CUDA was created by the NVIDIA company and uses a SIMT (Single Instruction, Multiple Threads) parallel systems model based on the use of threads, warps, blocks, grids, and kernels [38].In CUDA, the GPU is named the device, and the CPU is referred to as the host. To execute a parallel process in the device, first the process must be initiated from the host via the definition and call of a *kernel*. However, CUDA dynamic parallelism enables a CUDA kernel (GPU function) to launch new kernels and avoid communication overload with the host[22].

Figure 3 shows the parallel architecture for ParDWDM using CUDA. In dynamic parallelism[22,24], the threads in the first level, create new threads at a lower level to be executed concurrently. The number of levels depends on the features of the hosted GPU card. Pseudocodes 3,4, and 5, show respectively the implementation. Firstly, in pseudocode 3 in line 3, the GPU function named $kernel_D$ will be called D times at the same time and each with a different block. Thus, at this point, the order of temporal complexity is $O(1)$. Next, $kernel_D$ (pseudocode 4) calls the GPU function named $kernel_N$ N times, each with a different thread using nested parallelism, this step of the proposed parallel algorithm has a temporal complexity of $O(1)$. Finally, GPU function $kernel_N$ (pseudocode 5) calculates the set S in line 2 and compute Eq(6) inside the loop that goes from line 3 to 7. These lines have a complexity of $O(N^3)$, as mentioned previously.

Therefore, the computation execution time is reduced from $O(N^5)$ to $O(N^3)$.

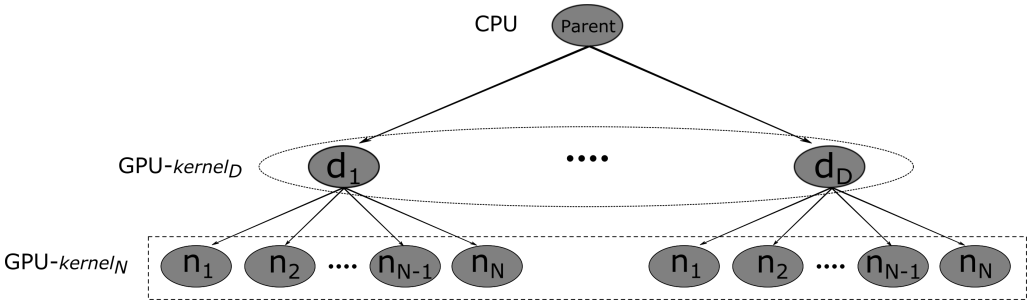

Figure 3. A Proposed parallel architecture for ParDWDM Algorithm using CUDA dynamic parallelism

Pseudocode 3: Pseudocode for the proposed parallel algorithm using CUDA

Input : N,μ,ϵ
Output: L_{max_fwm}, P_{o_fwm}
1 **begin**
2 Let Eq(24) to obtain D_{upper_limit}
3 $[L_{max\_fwm}], [P_{o\_fwm}] \leftarrow kernel_D <<< D, 1 >>> (parameters \dots)$
4 **end**

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2021 doi:10.20944/preprints202103.0558.v1

Pseudocode 4: Pseudocode for the $kernel_D$

```
begin
    [P] ← kernel_N <<< 1, N >>> (parameters ...)
    syncthreads()
    Compute Y Eq.(8) ;
    Let Eq.(13) and Eq.(14) to obtain [L_max_fwm]_i and [P_o_fwm]_i for D_i ;
end
```

Pseudocode 5: Pseudocode for the $kernel_N$ which is called by $kernel_D$

```
begin
    Let S = {s_w = (f_i, f_j, f_k) | ∀i, j, kϵ{1..N}}
    for each s_w of S do
        if (f_n == (f_i + f_j − f_k)) then
            Compute Eq.(6) and add to [P]_n;
        end
    end
end
```

5. Computation times and results

The results obtained from experiments using $N = 240$ channels and $\mu = 144$ are presented. These experiments were conducted on an Intel Xeon computer with 12 cores, each with 1.6 GHz of clock speed in the case of OpenMP, and with an NVIDIA Quadro k6000 card with 2880 CUDA cores and 12 GB of memory in the case of CUDA-C.

The parallel programming platform on a CUDA GPU version 10 was used with Visual Studio 2017 with a plug-in for CUDA. OpenMP 2.0 with Visual Studio 2017 was used for the CPU parallel model.

The experiments consisted of applying the parallel methods and the sequential method to parameter above shown ($N = 240$ and $\mu = 144$).

We use the Speedup to measure the outperformance of the parallel algorithm by comparing its execution time versus the execution time of the best sequential algorithm [39]. The Speedup is defined as:

$$S_p(n) = \frac{T^*(n)}{T_p(n)}, \tag{18}$$

where $T^*(n)$ is the time taken for the best sequential algorithm to resolve a problem of size n and $T_p(n)$ is the time taken for the parallel algorithm with p processors to resolve the same size n problem. Optimal Speedup is defined by $S_p(n) \approx p$ and Efficiency is defined by

$$E_p(n) = \frac{T_1(n)}{pT_p(n)}. \tag{19}$$

Efficiency is a measure of the effective utilization of the p processors [39] in the parallel algorithm.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2021 doi:10.20944/preprints202103.0558.v1

Table 1. Sequential single core vs parallel multicore OpenMP execution

Number of CPU cores	Execution time in s	Speedup	Efficiency
1	898.783	1	1
2	472.591	1.90	0.95
4	254.039	3.53	0.88
6	174.577	5.14	0.85
8	141.948	6.33	0.79
10	133.123	6.75	0.67
12	136.021	6.60	0.55

Table 1 shows the values of Speedup and Efficiency for ParDWDM using OpenMP.The parallel method was up to six times faster than the sequential method using ≥ 8 processors. The Efficiency is almost optimal when using ≤ 6 processors. The Speedup decreases as the number of cores increases. This behavior is explained by Amdahl's law that for a large number of processors the efficiency of the algorithm does not increase in parallel algorithms after the maximum efficiency is obtained.

The performance using GPUs with nested CUDA-C is 20.58 seconds, which is 6.46 longer than its counterpart using the best time executed in the OpenMP implementation and more than 40 times better than the sequential equivalent. It is possible to decrease the time by 2.6 times using the CUDA SFU (Special Function Units) for *Sin* and *Sqrt* evaluation [40,41], however, the accuracy of the results is will be significantly affected.

Note that the number of CPU processors in our implementation is much lower than the number of GPU processors. Here we use the Performance Ratio (PR), which is a measure to quantify CPU speed for a specific quantity of input data (computational task) of size n using a constant number of processors concerning the total time taken by GPU performing the same input [19] . The Performance Ratio (PR) is defined as follows

$$PR(n) = \frac{T(n,\nu)_{CPU}}{T(n,\zeta)_{GPU}}, \tag{20}$$

where $T(n,\nu)_{CPU}$ is the time taken for an algorithm using ν CPU processors to resolve a problem of size n, $T(n,\zeta)_{GPU}$ is the time taken for an algorithm using ζ GPU cores to resolve a problem of size n.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2021 doi:10.20944/preprints202103.0558.v1

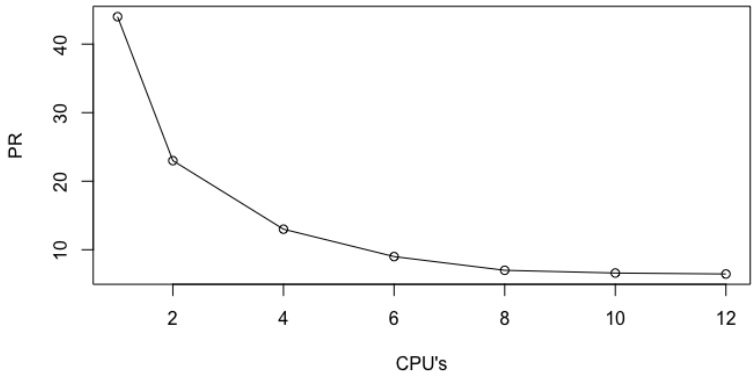

Figure 4. *PR* of the GPU implementation vs the multicore with 1,2,4,6,8,10,12 number of cores respectively for the algorithm ParDWDM

Figure 4 shows the PR in relation to the parallel implementation methods that use CPUs and GPUs when they receive the same data as the input n. Note that the parallel versions on GPU achieve speeds up to 6 times faster than those seen for the parallel implementations with 12 CPUs.

5.0.1. Simulation results

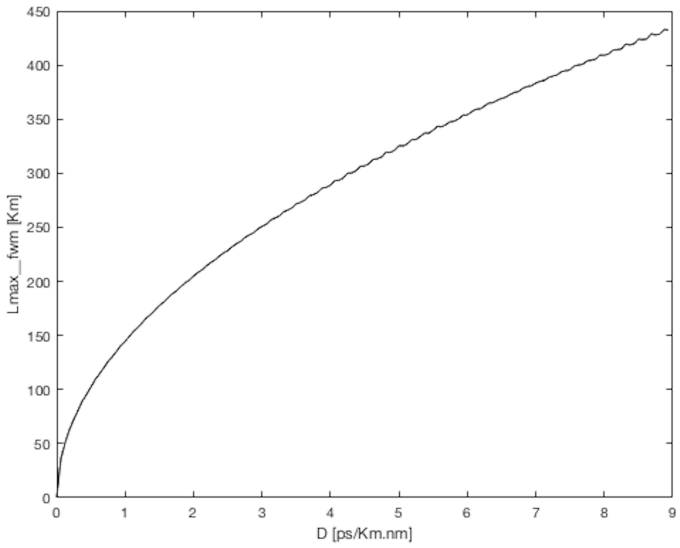

Figure 5. Maximum length evolution Vs Dispersion, due to FWM and ASE noise (240 channels)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 23 March 2021 doi:10.20944/preprints202103.0558.v1

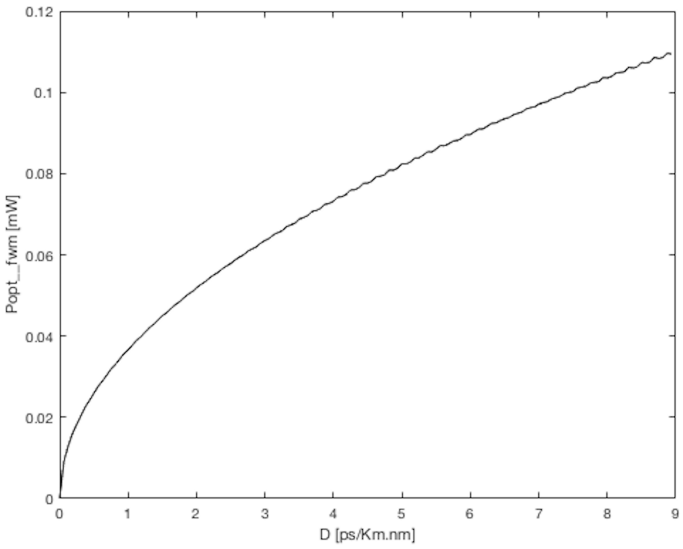

Figure 6. Optimum power Vs Dispersion, due to FWM and ASE noise (240 channels)

Figure 5 presents the maximum link length limit for a multi-stage system with optical repeaters while considering different dispersion values. The dispersion values are calculated from equation (17). It can be seen that at a dispersion of 9 ps/km-nm, the maximum permitted distance for the system is approximately 430 km. Figure 6 shows the optimum transmission power. Where it is observed that as the dispersion increases, the optimum power value also increases; up to 0.11 mW for 9 ps/km-nm dispersion.

Finally, note that due to restrictions of the algorithm, we could not report results for 240 channels in [13]. Consequently, the greater the number of channels the longer execution time.

6. Conclusions

The study of the limitations imposed by non-linear phenomena in a DWDM system is increasingly complex when many channels are transmitted. This work has presented the optimization of a multi-stage DWDM system limited by three factors (FWM, dispersion and amplifier noise). The maximum value of the optimum transmission power at a $D = 9ps/Km - nm$ was 0.11 mW. These results were calculated for 240 information channels. A parallel architecture for these simulations was used which reduces the execution time from $O(N^5)$ in one single processor to $O(N^3)$ using t number of GPU threads, where t is in $O(DN)$. Notice that D is proportional to $N/2$, therefore there exists $O(N^2)$ GPU threads, each one executing $O(N^3)$ operations. Thus, executing $O(N^5)$ in our sequential model is equivalent to executing $O(N^3)$ operations concurrently using $O(N^2)$ GPU threads under the parallel model. GPU outperforms by about six times the execution time compared to the best multi-core algorithm. Note that GPU also outperforms the best sequential model in terms of execution time by about 40 times. One can conclude that GPU is suitable for performing experiments that require many sections of fine granularity.

Author Contributions: All authors contributed to the research work.

Funding: This research received no external funding

Acknowledgments: The authors would like to express their gratitude to UNACAR, UADY, CONACYT and CIMAT from Mexico, and the Optical Sciences group at University of Twente in The Netherlands.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FWM	Four-Wave Mixing
DWDM	Dense Wavelength Division Multiplexing
ASE	Amplified Spontaneous Emission
SRS	Stimulated Raman Scattering
SBS	Stimulated Brillouin Scattering
SPM	Self-Phase Modulation
XPM	Cross-Phase Modulation
EDFA	Erbium Doped Fiber Amplifier
OpenMP	Open MultiProcessing
CUDA	Compute Unified Device Architecture
GPU	Graphics Processing Unit
CPU	Central Processing Unit
MPI	Message Passing Interface
DM	Dynamic Parallelism
SMP	Symmetric Multi-processors
API	Application Programming Interface
SIMT	Single Instruction, Multiple Threads
SFU	Special Function Units
PR	Performance Ratio

References

1. Agrawal, GP. *Fibre-optic Communication systems*, 4rd ed.; John Wiley&Sons, USA, 2010; pp. 233–294.
2. Rodas, PE.; Coronel, EJ. Simulation and analysis of a gain flat filter GFF for the correction of gain fluctuations produced by an EDFA amplifier for a WDM system. Proceedings of IEEE Chilecon, 28 october 2015, Universidad Central de ChileSantiago, pp. 319-322.
3. Rostami, A.; Rahbari, J.; Andalib, A. Investigation of power penalty in WDM systems for dispersion managed fibres.*Optik* **2013**,*124* 2072-2075.
4. Karfaa, YM.;Ismail, M.; Abbou, FM.; Shaari, S.; Majumder, SP. Effects of four-wave mixing crosstalk in WDM networks on the transmitted optical frequencies and wavelengths of channels for various fiber types, Asia-Pacific Conference on Applied Electromagnetics, 4 december 2007, Asian Malaysia, 1-5.
5. Djordjevic, IB. Transmission limitations of WDM transmission systems with dispersion compensated links in the presence of fiber nonlinearities, Proceedings of Papers TELSIKS, 19 september 2001, Nis Yugoslavia, 496-499.
6. Rana, M.; Khan, M. Dense Wavelength Division Multiplexing Optical Network System. *IJECE***2012**,*2*, 203–206.
7. Kowalewski, M.; Marciniak, M.; Sedlin, A. Nonlinear interactions in wavelength-multiplexed optical fibre telecommunication systems. *J Optic Pure Appl Optic***2000**,*2*,319-326.
8. Kaur, G.;Singh, MI.;Patterh, MS. Theoretical investigations of the combined effect of fibre nonlinearities, amplifier and receiver noise in a DWDM transmission system.*J. Opt. Fibre Commun.* **2009**, *6*, 1-10.
9. Wu, M.;Way, WI. Fibre Nonlinearity Limitations in Ultra-Dense WDM Systems. *IEEE J. Lightw. Technol.* **2004**,*22*,1483–1497.
10. Hadrien, L,;Hodzic, A.;Petermann, K.; Robinson, A.;Epworthet, R. Analytical Model for the Design of Multispan DWDM Transmission Systems. *IEEE Photon. Technol. Lett.***2005**, *17*,247-249.
11. Maeda, MW.; Sessa, WB.; Way, WI.; Yi-Yan, A. ;Custis, L.; Spicer, R. Laming RI . The effect of four-wave mixing in fibers on optical frequency-division multiplexed systems. J. Lightw. Technol **1990**,*8*, 1402–1408.
12. Ellis, AD.; McCarthy, ME.; Alkhateeb, MAZ.; Sorokina, M.; Doran, NJ. Performance limits in optical communications due to fibre nonlinearity. *Adv. Opt.* **2017**, *9*,429–503.
13. Alvarez-Chavez, JA; Sanchez-Lara, R.; Ek-Ek, JR.; Offerhaus, HL.; May-Alarcon, M.; Golikov, V. Optimum Power in a Multi-Span DWDM System Limited by Non-Linear Effects. *OPJ* **2018**,*8*, 337–347.
14. JáJá, J. *An introduction to parallel algorithms*; Addison-Wesley Publishing Company, Inc.,USA,1992.
15. Kumar, V.; Karypis, G.; Gupta, A.; Grama, A. *Introduction to parallel computing*,2nd ed.; Addison-Wesley Profesional, USA,2003.
16. Sanchez-Lara, R.; Trejo-Sanchez, J.; Lopez-Martinez, J.; Alvarez-Chavez, J. Simulation of an inelastic dispersive phenomenon: stimulated Brillouin scattering in a single-mode fiber segment through parallelism. *J Supercomput.* **2018**,*7*, 3264–3277.
17. Chandra, R.; Menon, R.; Dagum, L.; Kohr, D.; Maydan, D.; McDonald, J. *Parallel programming in OpenMP*; Morgan kaufmann, USA, 2001.
18. Pacheco, P. *Parallel programming with MPI*; Morgan Kaufmann,USA, 1997.
19. Yam-Uicab, R.; Lopez-Martinez, J.; Trejo-Sanchez, J.; Hidalgo-Silva, H.; Gonzalez-Segura, S. A fast Hough Transform algorithm for straight lines detection in an image using GPU parallel computing with CUDA-C. *J Supercomput.* **2017**,*73*,4823–4842.

20. Stojanovic, N,; Stojanovic, D. High Performance Processing and Analysis of Geospatial Data Using CUDA on GPU. *Adv Electr Comput Eng*,**2014**, *14*,109-114.
21. Chapman, B.; Jost, G.; Van Der Pas, R. *Using OpenMP: portable shared memory parallel programming*;MIT press, USA,2008.
22. Kirk, DB.; Hwu, WW.*Programming Massively Parallel Processors: A hands-on approach*;Morgan Kaufmann Publishers,USA,2016.
23. Yam-Uicab, R.;Lopez-Martinez, J.; Llanes-Castro, E.; Narvaez-Diaz, L.; Trejo-Sanchez, J. A Parallel Algorithm for the Counting of Ellipses Present in Conglomerates Using GPU.*J.Math Probl Eng* **2018**,*2018*,1–17.
24. Sharma, B.; Han, J.*Learn CUDA Programming*;Packt Publishing,USA,2019.
25. Inoue, K.; Four Wave Mixing in an Optical Fibre in the Zero-Dispersion Wavelength Region. *IEEE J. Lightw. Technol.***1992**,*10*,1533–1561.
26. Marcuse, D. Derivation of Analytical Expressions for the Bit-Error Probability in Lightwave Systems with Optical Amplifiers. *IEEE J. Lightw. Technol.***1990**,*8*,1816–1823.
27. Sanchez-Lara, R.; Alvarez-Chavez, JA.;de la Cruz-May, L.;Perez-Sanchez, GG.; Jaramillo-Vigueras, D. Cascaded Raman generation of the N'th Stokes in single mode-fibres.*Opt. Eng.* **2014** ,*53*,1–5.
28. Sanchez-Lara, R.;Alvarez-Chavez, JA.; Mendez-Martinez, F.; de la Cruz-May, L.;Perez-Sanchez, GG. Threshold and maximum power evolution of stimulated Brillouin scattering and Rayleigh backscattering in a SM fibre-segment. *Laser Phys.***2015**,*25*,1–5.
29. Hekmat, MJ.; Dashtabi, MM.; Manavi, SR.;Hassanpour, E.;Massudi, R. Study of the stimulated Brillouin scattering power threshold in high power double-clad fiber lasers.*Laser Phys.* **2012**,*23*,1–5.
30. Kovalev, VI. Peculiarities of the single and repetitive pulse threshold pump power for stimulated Brillouin scattering in long optical fibres. *Laser Phys.* **2014**,*24*,1–5.
31. Tithi, FH.; Islam, MS.; Majumder, SP. Compensation of SBS using cross-phase modulation in WDM transmission system. 7th ICECE, Dhaka, Bangladesh, 20 Dec. 2012, pp. 39-42.
32. Henry, P. Lightwave Primer.*IEEE Journal Quantum Elect.* **1985**,*21*, 1862–1879.
33. Chraplyvy, AR.; Tkach, RW. What is the Actual Capacity of Single-Mode Fibres in Amplified Lightwave Systems. *IEEE Photon. Technol. Lett.* **1993** ,*5*, 666–668.
34. Vanholsbeeck, F.; Sthepane, C.; Emplit, P.; Sylvestre, T. Coupled-mode analysis of stimulated Raman scattering and four-wave mixing in wavelength-division multiplexed systems.*Opt. Commun.* **2005**,*250*,191–201.
35. Cormen, T.; Leiserson, C.;Rivest, R.;Stein,C.*Introduction to Algorithms*,3rd ed.;The MIT Press,Cambridge,USA, 2009.
36. Gebali, F. *Algorithms and Parallel Computing* ;John Wiley & Sons Inc.,USA, 2011.
37. Cook, S. *CUDA programming: a developer's guide to parallel computing with GPUs*; Morgan Kaufmann Publishers, USA,2012.
38. Aubanel, E.*Elements of Parallel Computing*;CRC Press,USA, 2016.
39. Blazewicz, J.; Ecker, K.; Plateau, B.; Trystram, D. *Handbook on parallel and distributed processing*;Springer,USA, 2000.
40. Januszewski, M.; Kostur, M. Accelerating numerical solution of stochastic differential equations with CUDA. *Comput. Phys. Commun*,**2010** ,*181* 183–188.
41. Spiechowicz, J.; Kostur, M. ;Machura, L. GPU accelerated Monte Carlo simulation of Brownian motors dynamics with CUDA, *Comput. Phys. Commun*,**2015** ,*191* 140–149