High Performance Computing (3-0-0-6)

Lecture Notes: Pre Mid Semester Part

Mid Semester Examination (20th Sep 2016), Mid Sem Model Solution

---

Mid Semester Examination (22nd Nov 2016), EndSem Model Solution

---

Lecture Notes: Post Mid Semester Part (All questions of End Sem Exam will be from Post Mid Sem Part)

1. **26 Sep 2016: Large Scale Computer:** Interconnection and Programming, architecture of Param-Ishan, array-tree-star-mesh-hypercube-fat-tree and complete graph properties in term of Diameter, link and Bisection Bandwidth. Graph embedding: tree to mesh example, parameters to optimize at the time of embedding (dilation, load factor and congestion). [Optional reading Embedding tree to mesh, not for Exam]

2. **27 Sep 2016: Scheduling Algorithm** $\alpha|\beta|\gamma$, $1|\sum w_jC_j$, $P/Q|ptmn|C_{max}$, $P|C_{max}$, Machine environment (identical (P), uniform/speed scaled (Q), unrelated/heterogeneous(R)), task environment (execution time, release time, deadline, preemption, dependency (independent, chain, in/out-tree, fork-join, DAG)), Optimization criteria (C_{max},$\sum C_j$,$\sum w_jC_j$, $\sum w_jT_j$, $\sum w_jU_j$)
[[Chapter 1 of Scheduling Algorithm By Peter Brucker]][[Survey of Scheduling Algorithm]]

3. **28 Sep 2016: Scheduling Problems:** Exponential Problems ($P|C_{max}$, $P|prec,pi=1|C_{max}$, $P|prec,pi=1,ptmn|C_{max}$, $P|tree|C_{max}$), List scheduling-2 Approximation for $P|C_{max}$, LPT schduling 4/3 Approximation for ($P|C_{max}$), Critical Path or Highest Level First Greedy algorithm 2-Approximation for $P|prec,pi=1|C_{max}$ and $P|prec|C_{max}$. Polynomial time solution for $P|ptmn|C_{max}$, $Q|ptmn|C_{max}$, P|tree, $pi=1|C_{max}$ using the HLF algorithm, P|pi=1|Cmax using simple executing in ceil(n/m) phases. **$P|prec,ptmn|C_{max}$ is in NPC [[Ref page 70 of Survey of Scheduling Algorithm]]**, Summery of Scheduling Problems From Brucker Book Brief Summary of Complexity of Parallel Machine Scheduling

4. **03 Oct 2016: HPC, Data Center and Cloud Computing** : IITG HPC System (Param-Ishan user guide) Utilization of HPC system, Solution to low utilization of HPC system: private (reduce operational cost), made it public (pay and use for public users, need to ensure security). Virtualization allow to run virtual OS on host OS with upto maximum of 15% performance degradation, Virtualization allow HPC/Data center to go for public mode. In simple HPC, a user can see what other user are doing but in virtualize HPC user can simply submit job and not allowed to see others activities. When ever user submit a JOB to data center, the resource manager at data center create a VM (Virtual Machine) for the JOB and execute JOB on top of VM.
Basic architecture of Cloud System: Users, Broker, Cloud Resources and Cloud Resource Manager. Broker do the negotiation of cost and job envt with users.
Cost of setting up and running a Cloud: Initial Cost and Operation Cost. Operation Cost: Power Consumption, Cooling Cost and Management/Broker cost. HPC: contain many containers, a container contain many racks, a rack contain many rack server/chasis, a chasis contain many server/host/sockets/processor and a host/socket contain many cores. Power model of a server/host: simple model ($P=P_{idle}+(P_{max}-P_{min})*U$, where U is utilization), simple model 2 ($P=1/2\ CV^2f$, where C is capacitance, V is voltage and f frequency of operation), power model of multi-threaded processor ($P=B+i\delta$ and $\delta=S+k.f^3$, where B is based power consumption, i is number of active hardware thread, k is constant and f is frequency of operation)

5. **04 Oct 2016: Data Center and Cloud Computing:** SaSS (google office tool, on line compiler, on line tex maker, etc. user don't have control over OS, machine and performance) Pass (rent machine with OS, software (licensed), kernel of own and use, no control over BW, CPU and Memory usage and other configuration) Iass (user get machines with configurations (cpu, ram, hdd, n/w and gpu share) for some time, he can install any OS, kernel, software...);

6. **14 Oct 2016: Data Center Management Model**

   Private cloud vs public cloud; Cloud Economic Model: Initial cost, revenue model, service cost, user base, Operation cost (broker cost, management cost, power consumption cost and cooling cost);
   Processor power consumption (power model of multi-threaded processor ($P=Bp+i\delta t$ and $\delta t=S+k.f^3$)), chasis power consumption ($Pc=Bc+i\delta p$), rack power consumption ($Pr=Br+i\delta r$) and container power consumption ($Pco=Bco+i\delta co$)).
   Energy consumption and power consumption: $E = P.t = \alpha f^3. t/f = \alpha f^2$; Increase frequency speed of processor power consumption increase (f^3) and energy consumption increase (f^2) but execution time also increase. Energy and Performance are opposite to each other.
   Server Problem: Given a task set and system, to achieve a performance P what will be the energy consumption? Given a task with execution time L and deadline D, what will be energy consumption of the system to execute the task before deadline?
   Laptop Problem: Given a task set and system, what will be of the system with energy budget B? Given a task with execution time L and energy budget of system, what will be earliest finishing time of the task?
   Work of Resource manager and brokers is give maximum profit to cloud provider by managing resource nicely/efficiently and attracting user with a competitive price to user and security.
   Efficient Resource Manager : Try to reduce the operation cost of cloud by scheduling the user request (VMs) onto resources to utilize the resources efficiently with less power consumption and cooling cost. Time, Energy and Cooling are three important parameters in scheduling the VMs onto the cloud resources.

7. **05 Oct 2016: Project Problem Definition** Project Problem Statement
   Scheduling a set of independent real time tasks on system with virtually infinite processor and processor power consumption model is $P=\alpha f^3$. Every task have three parameter arrival time, length and deadline. We need to schedule in a such a way that total energy consumption is minimize and no deadline is missed.
   Approach: choose tasks one after another in any order. For each task, calculate critical frequency for the task $f_c=l_i/(a_i-d_i)$, get a processor from infinite pool of processor and run the task on the choose processor at f_c, after finishing the tasks, return the processor to the pool.
   This approach consume minimum energy for the task system and without missing deadline of any task.

8. **17 Oct 2016: Warehouse Scale-Computing (WSC) and Technique for Improving Energy Consumption of WSC**
   [[Ref: Chapter 6 of Computer Architecture Book, Hennessey-Paterson]] Criteria for WSC: (a) Cost-Performance, (b) Energy Efficiency, (c) Dependability, (d) Network I/O and (e) Executing interactive and batch mode workloads. CRAC (Computer AIR conditioning Unit), Chiller 30%-40% of power, CRAC 10-20% of power of IT infrastructure. IT equipment power consumption: 33% processors, 30% RAM, 10% disks, 5% network, 22% others. Power Utilization Effectiveness (PUE) = (Total Facility Power) / (IT Equipment Power). PUE is greater than equal to 1. PUE 1.55 have almost 0.55 of Cooling. PUE depends on workload and external air temperature. Cost of WSC : CAPEX (Capital cost expenditure) and OPEX (Operational cost Expenditure). OPEX is 2% of 1st year of CAPEX+OPEX in year 2006. 10 year cost OPEX is 30% of total cost. CAPEX is reducing year by year as cost of hardware is reducing. OPEX is increasing as Power cost is increasing. Cloud: Amazon EC2: 8 core, 8GB, 2TB HDD for 1 hour cost $0.68.

   [[Ref: Energy Efficiency of Large System]] Energy Efficiency for Nodes:(a) Techniques for measuring and modeling power consumption, (b) Node level energy optimization (c) grid/data center power management and (e) cloud Virtualization; Node level energy optimization depends on (a) h/w capability (SSD, Si0 state,...), (b) Sleep state for supporting Dynamic power Management (DPM), (c) DVFS and (d) s/W or kernel/OS support to use the H/W support, DPM and DVFS; Grid/Data Center level Power management: (b) Load and Efficient Electric Component management (PSU, PowerGrid in side DC, Inteligent AC vent controller, CRAC,), use of Solar/Wind energy, place the DC at cool place ("Google have put their entire cloud in side the sea to reduce cooling cost"), place DC near to power sources (b) Thermal management (scheduling to reduce temperature/AC cooling requirement), (c) Workload consolidation (reduce Physical number of active/switch on component/machine), (d) Energy aware Task Scheduling using DPM, DVFS, ..; PowerTOP : measuring EC of computing components, use of ACPI

(Advanced Configuration for Power Interface), PUE, Data Center Infrastructure Efficiency DCiE=1/PUE, Node Power=Pcpu(37%)+Pmem(17%)+Pdisk(5%)+Pslot(23%)+Pmotherboard(12%)+Pfan(5%). Virtualization allows (provide on more level of abstraction to system, so) the HPC system manager to consolidate, migration and load balance in efficient way.

9. **18 Oct 2016: Power/Energy Consumption Models for Data Centers : Part I**
[[Ref:Data Center Energy Model]]
Given the data, modeling and fitting data is very difficult:- fitting to some curve or line with minimizing mean square error. So given the data, there may be many models for the same. Measurement always give correct data (provided the measurement is ideal).
Google Data Center (DC): Energy send 50% in Cooling , 11% in Power Conversion, 26% in server/storage, 10 in H/W and N/W and 3% in lighting. System Energy Optimization Cycle: In every epoch (say hour), try to reduce the energy consumption by predicting the load and utize to reconfigure the system state by using the performance-power model of components of the system. Real system power consumption modeling is extremely difficult, also no one can monitor/measure power consumption for the components. So it is nice to use, already known power model of the components of the DC. P_t=f(S_t,A_t,E_t) and P_{t+1}=f(S_t,A_t,E_t), where g(.) is load prediction model. St is internal system state (of OS, host, HDD, SDD, NIC, ...), At is input of application hosted on DC or JOB need to be executed and Et is execution and scheduling strategy (workload assignment, DVFS, ..) at time t.
Digital circuit level power and energy model: E= P.T, P_{total}=$P_{dynamic}$+P_{static}, P_{static} α I_{static}.V, $P_{dynamic}$=$P_{switching}$+$P_{short\text{-}circuit}$+$P_{leakage}$. $P_{switching}$ is primary/major part of $P_{dynamic}$. $P_{switching}$ =$P_{capacitance}$ = A.C.V^2.f.
Power model of a server/host: simple model (P=P_{idle}+(P_{max}-P_{min})*U, where U is utilization), simple model 2 (P=1/2 CV^2f, where C is capacitance, V is voltage and f frequency of operation). Improved Non-Linear Power Model P_u=P_{idle}+(P_{max}-P_{idle})*(2*U-U^r), where r is calibration parameter depends on server configuration (r=1.4). Normalized power: $P_{norm,u}$=($P_{syst,u}$-$P_{idle,u=0}$)/($P_{busy,u=1}$-$P_{idle,u=0}$), another normalized power model $P_{norm,u}$=1-h(u)$^{-1}$=1-(c1.U^{c2}+c3.U^{c4}+...).

10. **19 Oct 2016: Power/Energy Consumption Models for Data Centers : Part II** [[Ref: Energy Efficiency of Large System]] Additive Power Models, System Utilization Based Power Model, Processor Power Models: Single core, Multi-core and GPU enabled Processor, Memory and Storage Power Models, Data Center Level EC: Group of Server, Network and Power/Cooling Power Models.

11. **24 Oct 2016: Cloud Scheduling Basic:**
Task Scheduling on Multi-core (T(t1, t2,...tn) → M(p1,p2,..pm)), Cloud Task Scheduling: Task, Virtual Machine and Hosts (TT(t1, t2,...tn) → V(vm1,vm22,..vmm) → H(h1,h2,...hh)); Task to VM scheduling is user/application level scheduling; Resource Management in Cloud (VMs to Hosts). Each VM characterized by core utilizations. Each task is characterized by VM utilization. User can hire M VM to run N tasks, but all the time all the VMs may not be running. Current utilization of VMs can be predicted (upto some accuracy) from past utilization of user/task. Number of VMs required by user is usually less as compared to hired VM by the users. Colud Resource Manager takes this fact and utilize to reduce power consumption and increase profit. (Similar to flight/bus over booking by Broker assuming many will not turned up).

VM Consolidation: Reducing number of active hosts reduces Power Consumption and hence the total Energy consumption. Simply reducing number of host may increase number of VM migration, and power density of consolidated area, which in turn require more cooling for that area. Load balancing among host will increase VM Migration cost and many host are active and distance between hosts will be higher, but require less cooling power as energy density of host is balanced.
[[Thermal Model of CMP (Page 3, Task Assignment)]]Temperature model of a host can be modeled as summation of (a) previous temp of host, (b) current workload on the host, (c) effect of adjacent hosts based on their temp, and (d) cooling rate of the host. Number of migration effect the performance to the user, so it is not advisable to migrate more than 10% of VMs at any time. VmM migration have performance penalty. Predicted utilization of VM is not accurate, so this can be modeled as u ± 10%. So it is advisable to not to map 100% utilization to a host. We should map us to 90% utilization of the host, so that the host can serve the VM nicely and QoS to user is ensured otherwise cloud provider need to pay penalty to user. If CPU utilization goes above 80% power consumption increase at higher rate with

utilization hence require more power for computing and cooling the chip [[ref PowerConsumption Vs CPU Utilization (Figure 2)]].

Task (Ti, represented by index i), Number of VM (Vmj, represented by index j) and number of host (Hk, represented by index k). Task i mapped to VMj of host k.

12. **25 Oct 2016: VM Scheduling on Cloud Data Center**: [[Resoucelloc-IEEE-TPDS-2013]] Dynamic Resource Allocation using VM for Cloud: Over load avoidance, Green Computing, Skeewness(p)=sqrt($\text{sum}_{i \text{ in hosts}}(r_i/r_{avg}) -1)^2$), Hot spot and cold spot, Hot spot is the utilization of any of it resource is above threshold, temp(p)=$\text{sum}_{r \text{ in } R}$ $(r-r_t)^2$, where R is set of overloaded resources of server and rt is ho threshold for the resource r. Typical value of hot threshold 0.9, cold threshold 0.25, green threhold 0.4 and 0.05 in consolidation limit (only a maximum of 5% of VM are allowed to migrate at a time). When system utilization bellow green threshold the system call for consolidate and reconfigure. When there are many hot spot and higher skewness then system reconfigure to a new state.

    Load prediction used Exponentially Weighted Moving Average EWMA model E(t) = αE(t-1)+(1-α)O(t), 0 ≤ α ≤ 1, where E(t) estimated/predicted and O(t) is observed load at time t and α is trade off between stability and responsiveness. Improved model uses E(t) = O(t) + |α|(O(t)-E(t-1) with -1 ≤ α ≤ 0; Based on another Fast Up and Slow Down (FUSD) uses two alphas: one for up (α1=-0.2) and other for down (α2 =0.7). For up slope (or when O(t) > E(t-1)) to reflect the acceleation of resource demand, we can use α1=-0.2, so E(t)=O(t)+0.2(O(t)-E(t-1)) instead of E(t)=0.2*E(t-1)+0.8O*(t) and for down slope (when O(t) < E(t-1)), it is better to be conservative and use normal value of α, so E(t)=0.7*E(t-1)+0.3*O(t).

    [[Kejiang-IEEE-TPDS-2015]] Type of workload to DC: DC hosting Data bases, File server, Java Server and Web Server; File and DB server are I/O intensive where Java and Web Server are compute intensive; Complementary co-location for Server is better (one I/O intensive and one CPU intensive). Performance loss for migration (Java Server 18%, File server 53%, Data Base Server 34% and File server 24%). Based on SLA performance is Migration is allowed. Migration constraints is less 18% all are not allowed to migrate, if above 18% then JS is allowed to migrate.

13. **26 Oct 2016: Work flow and Work flow scheduling on Cloud:**
    [[WMS]] Work flow example Montage (Creation of mosaic picture of sky using data from many camera), LIGO/Inspiral (Gravitational wave detection work bench), CycbeShake (Simulation of earthquake) and EPIGenomic (Genome sequence analysis). All these work flows are represented using DAG. Scientific work flow management systems (many examples: Pagasus, Apache Airabata, HPC online, Fireworks, TimeStudio). SWMS help to create work flow of scientific application, visualize, simulate on virtual hardware and finally allow to attach process/job on each node and execute on real system. Automated workflow generation from Large DAG of scientific program by coarsening the DAG (by graph coarsening methods: metis, Minesota university), these coarsen DAG is used as scientific work flow. [[Pegasus documentation and work flow Galary]]

    Scheduling DAG on multiprocessor is in NPC ==> heuristics. [[End to End Delay minimization of Work flows in Cloud under Budget Constraints;]] Assume there are five type of VMs (Tiny, Small, Medium, Large, XL), cost of VM depends on it size, larger the size of VM, costlier and speeder. Take all tasks of WF and run on Tiny VM and estimate cost Cmin. Ii Cmin is greater than the budget then work flow cannot be executed. Take all the tasks and run on large machine and cost is less then the budget, allocate XL VM to all the task of WF. If Cmin < B < Cmax, find a global budget level for all the task and assign a type of VM. After that for each time, [Do the local adjustment] take a node from current critical path and try to map to higher VM with meeting deadline and highest time benefit.

14. **31 Oct 2016: Memory mapping and access latency, processor mapping and Reliable data storage RAID:**
    [[Dual Memory Mapping, Memory Bank Assignment and CS523 lecture slide [18-21 slides]]] Variable Mapping to Dual Memories (conflict graph generation, Max Spanning tree/(MinCut) and assign array variable memory), Example of memory mapping
    RAID (for reliability, capacity, availability and performance) RAID level 0 (only stripping to improve performance and capacity), 1 (miring: 1/2 of the disk are used for mirror, writing at /12 the speed but read at full speed of RAID-0 and capacity of 1/2), 2 (bit level striping with ECC), 3 (byte level striping

with Parity) , 4 (Block level striping with parity) and 5 (striping with distributed parity).
Thread to processor mapping in multi-threaded application in multi-core: Assume N threaded application need to mapped to N core multi-processor, one thread to one processor. Between threads i and j, amount data need to communicate is given as es(i,j) and between core i and j link bandwidth is given as et(i,j). Now we need to map (MAP) the threads to processors (one to one) in a such a way that $\Sigma_{i=1}^{N}$ $\Sigma_{j=1}^{N}$ es(i,j) * et(MAP(i),MAP(j) is minimized. This same is facility location problem or quadratic assignment problem (which is in NP Hard). This can be solved very nicely (probably optimal in reasonable time) using Simulated Annealing. [[Task Assignment in NOCs and Section 2 of Task Assignment TIC-PIG]]

15. **01 Nov 2016: Resource Management in HPC System:**
[[SLURM]] SLURM (Simple Linux Utility for Resource Management),PBS and Torch. Architecture of SLURM, Scheduling policy of SLURM, Slurm Job submission script, Architecture of HPC (node, cpu/node, storage, network); Login nodes, seem less ssh and storages in mpi enable HPC; $sbatch test.sh, $srun a.out

Resource allocation for on line jobs: dispersion factor (corner nodes or number idle cores to a core), centrifugal factor (amount free are connected to the core). Resource allocation for 2D mesh of cores. Combined cpu and storage space allocation for high compute and space demand job. [[Section 4, 5 of ChouPaper]]

16. **02 Nov 2016: Auto Tunning MPI program on HPC, HPC System and Benchmarking :**
[[Process Placement in Multicore Clusters]] Auto tunning of MPI program; Most of the program runtime communication exhibit similar communication pattern for each run; Generate communication pattern between processor of a MPI program; Use this communication profile along with target architecture platform to reorder the rank of process; MPI_dist_ graph_create API;
Using the same technique for shared HPC where requested resource location is not know earlier only at run time you will get to know; In shared HPC environment, run a micro-kernel to get the process mapping information and calculate distance (bandwidth or latency) between the location of all pair of processes. Based on this micro-kernel result -reorder the rank of process to minimized the communication overhead. Now MPI executable have two parts (a) process rank-reordering and (b) actual execution
Most of the program run multiple iterations to accomplish a big work, and if each iteration communication pattern behaves in similar way then we have a great chance to optimize the communication of later iterations by learning/experience from some initial iterations

17. 07 Nov 2016: Project Discussions and Solution Approaches
18. 08 Nov 2016: Project Discussions and Solution Approaches
19. 09 Nov 2016: Project Discussions and Solution Approaches
20. 15 Nov 2016: Project Discussions and Solution Approaches
21. 16 Nov 2016: Project Discussions and Solution Approaches
22. 17 Nov 2016: Project Discussions and Solution Approaches

Course Contents:

- [3 Lects] Parallel Processing Concepts; Levels and model of parallelism: instruction, transaction, task, thread, memory, function, data flow models, demand-driven computation etc;
- [2 Lects] Parallel Architectures: super-scalar architectures, SIMD, multi-core, multi-threaded, server and cloud;
- [2 Lects] Basic Optimization for Serial Codes, Caches and Data Access Optimizations;

- [4 Lects + 2 Lects] Parallel languages and programming environments: Pthread, OpenMP, MPI, Java, Cilk; Performance Analysis of Parallel Algorithms;
- [3 Lects] Accelerated HPC: architecture, programming and typical accelerated system with GPU, FPGA, Xeon Phi, Cell BE, etc;

- [3 Lects] Fundamental design issues in HPC: Load balancing, scheduling, synchronization and resource management; Operating systems for scalable HPC;

- [3 Lects] Fundamental limitations in HPC: bandwidth, latency and latency hiding techniques;
- [1 Lect] Scalable Storage systems: RAID, SSD cache, SAS, SAN;
- [2 Lects] Benchmarking HPC: scientific, engineering, commercial applications and workloads;
- [2 Lects] HPC based on cluster, cloud, and grid computing: economic model, infrastructure, platform, computation as service;

- [3 Lects] Power-aware HPC Design: computing and communication, processing, memory design, interconnect design, power management, Energy Aware Scheduling; Energy Efficiency of Large System, Data Center Energy Model
- [1 Lect] Advanced topics: peta scale computing; big data processing, optics in HPC, quantum computers;

Lecture time and Venue: Time slot B1 (Mon 4-5PM, Tue 3-4PM, Wed 2-3PM), Core II, Room:2204

Rules:

- MidSem 40%, End Sem 30%, Project (Mostly theory/Algo/Design and a small amount of C++ coding) 30%.
- There will be 20 lectures before Mid Sem, 10 lectures after Mid Sem, Most of the post mid sem part will be on solving Project Problem.
- Student need to choose one project out of three Projects (will be defined later) of the course.

Text/reference Book:

1. George Hager and Gerhard Wellein. *Introduction to High Performance Computing for Scientists and Engineers*CRC Press, India, 2010.
2. Vipin Kumar, Ananth Grama , Anshul Gupta , George Karypis. *Introduction to Parallel Computing (2nd ed.)* Pearson India, 2003.
3. John L. Hennessy and David A. Patterson. *Computer Architecture: A Quantitative Approach (5th ed.)* Elsevier India Pvt. Ltd. 2011.
4. David B. Kirk and Wen-mei W. Hwu. *Programming Massively Parallel Processors: A Hands-On Approach (1st ed.)* Elsevier India Pvt. Ltd. 2010.
5. Michael T. Heath. *Scientific Computing: An Introductory Survey (2nd ed.)* McGraw Hill Education (India) Private Limited, 2011
6. Online HPC Book
7. Very good HPC web link by George Hager HPC course by Hager
8. Parallel Implementation of Meta-Heuristics on GPU Talbi IEEE TC Paper for QAP (Quadratic Assignment Problem/Facility location Problem, TSP (traveling Salesman), PPP (Permuted Perceptron Prob) and WCF (Wistress continuous Fun) problems.
9. A survey on Cloud Resource Scheduling ACM Survey Cloud Scheduling
10. Energy Efficiency Techniques in Cloud Computing: A Survey and Taxonomy PDF
11. A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing SystemsPDF
12. Cloud Computing: Survey on Energy EfficiencyPDF

- Resources for Cilk: cilktool cilk-5.4.6.tar.gz, How to Install Cilk HowtoInstallCilk.txt Test program and Makefile for cilk cilkmatmultest and Cilk Mannual And Resources at Cilk@MIT, PowerPoint: lecture-1.ppt, lecture-2.ppt, lecture-3.ppt
- Intel Xeon Phi Resources : Best practice Guide Xeon Phi , Intel Xeon Phi Programming Model
- Free Cuda Programming Book: Cuda By Example , LocalCopy Cuda By Examples.PDF
- Official Cuda Programming Guide Nvidia Cuda Prog Guide
- Guid to OpenMP: OpenMp Guide, OpenMP HandsOn A Handson On OpenMP at SC08 , An Introduction to Parallel Programming with OpenMP, OpenMP by Examples, Intro to OpenMp
- Pthread Programming Pthread Tutorial , Multithreaded Guide (SunMicro)
- Sample Pthread Examples Samples by ABS
- C++ thread References C++ Multithreading
- Optimization of Computer Programs in C : Optimizing C Code
- Optimization of C/C++ : Optimizing C/C++ Code
- C++ Concurrency in Action, by Anthony Williams Book PDF Version

- Art of Multiprocessor Programming by Herlihy and Shavit Book PDF Version