

# עיבוד מקבילי - מושגים ונושאים שכיסינו בקורס עד-כה

המידע ניתן כאן כחלק מההכנה הנדרשת למבחן  
בקורס "מבוא לעיבוד מקבילי". אין באמור  
במסמך זה להצביע על נושא מסוים שישאל בבוחן  
ואין הנאמר כאן מכסה את כל החומר העשוי  
להיכלל במבחן

# היקף

- כל המפגשים שהתקיימו
- יש לעבור על כל החומר שהוצג ותורגל לרבות מה שמפורסם באתר הקורס, מה שנאמר בכיתה ובמעבדה, תרגילי הבית וחומר הקריאה המומלץ

# מושגים

- מושגי יסוד
- חוק פלין - טקסונומיה
- חוק גוסטפסון
- מתי משתמשים בחישוב מקבילי
- יעילות מיקבול
- האצה
- חוק אמדהל
- איזון עומסים
- טופולוגיות רשת
- עצי חלוקה
- חישוב בעזרת מסרים
- חישוב בזיכרון משותף
- MPI
- OpenMP
- Condor
- Deadlock
- פרמטרים להערכת רשתות תקשורת

- **HTC**
- **SETI@home**
- **Grid Computing**
- **Cloud Computing**
- **MPMD/SPMD**
- **Point-to-Point comm.**
- **Collective commands**
- **Send/recv**
- **communicator**

- **Master-Worker**
- **Domain Decomposition**
- **Profiling**

• יחס זמן חישוב/תקשורת

• מבחני ביצועים

• צווארי בקבוק

• חומרות נפוצות

• **Top500**

• **HPC**

- **Synchronous commands**
- **Immediate commands**
- **(Non)Blocking commands**
- **מנדלברוט**
- **עיבוד תמונה בסיסי**
- **מונטה-קרלו**
- **הקצאת משימות סטטית/דינאמית**
- **Pipeline**

- **Broadcast**
- **Reduce**
- **Scatter**
- **Gather**
- **All-to-All**
- **Wtime**
- **Comm\_size**
- **Comm\_rank**
- **Barrier**
- **All Gather**

- **Forall**
- **Data parallel**
- **Jacobi Iterations**
- **Network bandwidth & Latency**
- **Block and Strip partitions**
- **Ghost Points**
- **Code Safety**

- **Partitioning**
- **Divide-and-Conquer**
- **$N$ -body problem**
- **Bucket sort**
- אינטגרציה נומרית  
מקבילית
- **Barnes-Hut Algorithm**
- **Synchronous Computations**

- **OpenMP Atomic**
- **Parallel regions**
- **Race condition**
- **Nowait**
- **- SHARED**
- **- PRIVATE**
- **- FIRSTPRIVATE**
- **- LASTPRIVATE**

- **FORK-JOIN**
- **Thread-Safe**
- **Locks**
- **Critical Section**
- **OpenMP sections**
- **Work Sharing**
- **OpenMP Reduction**
- **OpenMP critical**
- **OpenMP Barrier**

- **Gauss Elimination**
- **Gauss-Seidel Relaxation**
- **Red-Black Ordering**
- **Cilk principles**
- **MatlabMPI**
- **pMatlab**
- **Parallel FFT**
- **Xeon Phi**
- **Xeon Phi: N-Nody optimization**
- **Xeon Phi: Particle Binning example**
- **vectorization**
- **simd**
- **Load Balancing**
- **Termination Detection**
- **Static/Dynamic load balancing**
- **Centralized/De Centralized load balancing**
- **Cilk keywords**
- **Strands and Knots**
- **Parallelization**
- **Critical Path**
- **ivdep**



# OpenMP summary - I

Clause	Description
firstprivate	Specifies that each thread should have its own instance of a variable, and that the variable should be initialized with the value of the variable, because it exists before the parallel construct
lastprivate	Specifies that the enclosing context's version of the variable is set equal to the private version of whichever thread executes the final iteration (for-loop construct) or last section (#pragma sections)
nowait	Overrides the barrier implicit in a directive
ordered	Required on a parallel for (OpenMP) statement if an ordered (OpenMP Directives) directive is to be used in the loop
private	Specifies that each thread should have its own instance of a variable
reduction	Specifies that one or more variables that are private to each thread are the subject of a reduction operation at the end of the parallel region
schedule	Applies to the for (OpenMP) directive

# OpenMP summary - II

Directive	Description
atomic	Specifies that a memory location that will be updated .atomically
barrier	Synchronizes all threads in a team; all threads pause at the .barrier, until all threads execute the barrier
critical	.Specifies that code is only executed on one thread at a time
flush (OpenMP)	Specifies that all threads have the same view of memory for all .shared objects
for (OpenMP)	Causes the work done in a for loop inside a parallel region to be .divided among threads
master	Specifies that only the master thread should execute a section of .the program
ordered (OpenMP Directives)	Specifies that code under a parallelized for loop should be .executed like a sequential loop
parallel	Defines a parallel region, which is code that will be executed by .multiple threads in parallel
sections (OpenMP)	.Identifies code sections to be divided among all threads
single	Lets you specify that a section of code should be executed on a .single thread, not necessarily the master thread
threadprivate	.Specifies that a variable is private to a thread

# מה עוד...

- משוואת החום
- MPI ב PI תכניות לדוגמה של חישובי OPENMP
- Game of Life
  - **Cellular Automata**
- **Iterative Methods**
- **Laplace's Equation**
- **Finite Difference Method**
- **Hybrid MPI+OpenMP**
- **HTCondor concepts**
- **HTCondor commands**
- **HTCondor universes**
- **HTCondor DAG**
- **HTCondor use cases**
- פקודות לינוקס בסיסיות
- ניתוח תכניות מחשב
  - מה התכנית עושה
  - איזה קונספט היא מבטאת
  - היכן צווארי הבקבוק
  - מיקבול של לולאות
- מה עושה קונדור
- **Matrix multiplication**
- **Solving a system of linear equations**
- **Parallel Image processing**

# Parallel Techniques

- Embarrassingly Parallel Computations
- Partitioning and Divide-and-Conquer Strategies
- Pipelined Computations
- Synchronous Computations
- Asynchronous Computations
- Strategies that achieve load balancing
- Sorting Algorithms
- Image Processing