**Brents theorm** :-

• Shared memory- Easy to Program-Explicitly read and write data-Problem:- Cache coherency

• Shared Address space-Better-Logically shared physically separate

Message passing-Everything is separate-Send and receive data explicitly

Data Parallel-Vector machines (GPU)-Same task different pieces of data

Extreme is Task Parallelism-Same data different task

• Clusters-Hybrid of the above models

\*COST FACTORS\* • Control • Data • Sync

Cost of these is more than the cost it takes to do the arithmetic . [ Overhead ]

**Master theorm :-**

* 1. Multi Socket

**Speedup ( Ideal ) = Θ ( P )**

**Weak Scaling :-**

**TOPOLOGY :-**

Bisection bandwidth :- bandwidth across smallest cut that divides the

network into two equal parts

**Tree :- L = P ; D = Log P ; B = 1**

**Improve Bisection BW by adding wires near root ( Fat tree)**

**Ring :- D = P/2 ; B = 2**

**2D Mesh D = 2( B =**

**2D Torus D = B = 2\***

**Hypercube D= d B = P/2**

**Butterfly D=logP B = P**

**Work optimality :- W = W\***

Multiple Hierarchy

Speedup = Work / Span ……………………. So span should be low . !! !! !! Also keep Work Low 😃

MPI comm size :- returns no.of processors and size . MPI COMM RANK reports the rank [0 to N-1]

**Mergesort – Serial merge**

Parallel Merge :-

Overall Depth =

Overall Parallelism =

MPI Collectives :- BCAST,REDUCE,BARRIER,SCATTER,GATHER,ALLGATHER,ALLTOALL,SCAN,REDUCE\_SCATTER

MPI Primitives :- INIT , FINALIZE,COMM\_RANK,COMM\_SIZE,SEND,RECEIVE

