**POWER WALL**

Il Power Wall è un punto oltre la quale non si è potuto andare nonostante tutti i miglioramenti tecnologici che sono stati fatti.

Non si riesce ad avanzare più di tanto perché c’è un limite invalicabile dovuto alla frequenza e al calore da dissipare e quindi la soluzione è stata quella di passare ad architetture con più processori.

Power = Capacitivo x Tensione^2 x Frequenza

Aumentare le prestazioni aumentando semplicemente la frequenza non è la cosa migliore da fare poiché se si raddoppia la frequenza si ha una quantità di calore da disperdere doppia.

La tensione non è riducibile più di tanto perché se riduco ulteriormente la tensione di alimentazione aumentano gli errori perché gli 0 si confondo con gli 1 per effetto del rumore termico.

**BENCHMARK**

**Che cos’è un benchmark?**

I benchmark sono programmi campione, appositamente scelti per misurare le prestazioni del calcolatore.

I benchmark sono pensati per fornire un carico di lavoro che possa essere significativo al fine di stimare le prestazioni sui carichi di lavoro tipici.

Ci sono dei benchmark che sono semplicemente dei riferimenti standard, sono dei programmi reali che veramente fanno qualcosa e che tutto sommato sono utilizzati perché sono rappresentativi di un’ampia classe di applicazioni.

Altri benchmark sono di tipo artificiale nel senso che sono scritti appositamente per fare quella cosa in modo tale da riprodurre il funzionamento dei programmi e quindi sinteticamente rappresentare quello che potrebbe fare un utente in certe condizioni.

Un benchmark è essenzialmente un programma utilizzato per misurare le prestazioni in maniera tale da poter fare dei confronti tra processori.

I benchmark devono rappresentare il workload (del carico di lavoro) delle cose che si dovranno eseguire effettivamente.

**AMDALH**

**Che cos’è la legge di Amdalh?**

Una regola che afferma che il miglioramento delle prestazioni reso possibile da una data modifica è limitato dalla quantità di tempo in quella modifica è effettivamente sfruttata.

Per ottenere il miglioramento delle prestazioni si deve operare sul caso più comune perché lì i miglioranti sono più significativi.

Ad esempio ho un programma che fa 10 minuti di elaborazione e 30 minuti di I/O, migliorare la parte di calcolo con un processore più veloce non ne vale la pena perché sarà preponderante l’aspetto I/O che non viene assolutamente toccato.

Per cui ci sono due fattori la parte affected quella che viene effettivamente migliorata dal mio fattore di miglioramento e la parte unaffected quella che non viene migliorata.La parte affected è divisa dal fattore di miglioramento.

Quando devo fare dei miglioramenti devo operare sulla parte che viene utilizzata di più e non su quella che viene utilizzata ogni tanto.