

Appunti di Green Computing 25/26

Questi appunti rappresentano una sintesi degli argomenti/articoli trattati dal professore Stefano Forti nell'anno accademico 2024/2025. Si limitano ai concetti richiesti nelle prove d'esame e non danno una visione completa della materia.

Introduzione

Come mai è così importante il Green Computing?

Il green computing è diventato essenziale con la rapida digitalizzazione degli ultimi 20 anni. La pandemia di Covid-19 ha accelerato significativamente questo processo, rendendo il lavoro da remoto una necessità per molte persone. La digitalizzazione di processi e servizi, spinta da tecnologie come Cloud, AI, Big Data, blockchain e HPC, è centrale nelle agende nazionali e internazionali. La diffusione di servizi digitali, specialmente video e immagini ad altissima risoluzione, è stata fortemente accelerata dalla pandemia di Covid-19, che ha reso necessario il lavoro da remoto.

Consumo energetico ICT → Oggi attorno al 5% della domanda energetica globale, potrà superare il 20% nel 2030

Ridimensionamento di Dennard → man mano che i transistor diventano più piccoli la loro densità di potenza ($[Watt / cm^2]$) rimane costante. ($P/S \cong k$)

E-waste → 50 milioni di tonnellate di RAEE (Rifiuti di Apparecchiature Elettriche ed Elettroniche) ogni anno

Accordo di Parigi (2015) → contenere il riscaldamento globale con obiettivi a lungo termine:

- ridurre l'impronta di ogni europeo da 10 a 2 tonnellate di emissioni annue
- riduzione delle emissioni di CO₂
- soglia per il riscaldamento globale
- finanziamenti da paesi "sviluppati" a quelli "in via di sviluppo"
- fondi a paesi con danni permanenti e irreversibili

Le aziende informatiche sono interessate al green computing per:

- ridurre i costi di gestione del prodotto, (quindi)
- aumentare gli utili
- migliorare la reputazione aziendale verso il personale (assunto e potenziale)

Definizione di Green Computing

Il **green computing** tratta la progettazione, la realizzazione e l'utilizzo di sistemi ICT, computer e dispositivi elettronici, in modo responsabile e sostenibile dal punto di vista ambientale, considerando in particolare il consumo energetico e impronta di carbonio.

Segue un approccio «**olistico**» analizzando tutto il ciclo di vita (progettazione -> produzione -> gestione -> dismissione) di un sistema.

Ingegneria del software sostenibile

Sviluppare sfruttando le peculiarità di linguaggi, delle strutture dati e/o dell'hardware utilizzato può aiutare a rendere il software più efficiente dal punto di vista energetico

È inoltre possibile programmare software **carbon-aware**, sfruttando

- **time shifting** = esecuzione di task asincroni quando l'energia è più pulita
- **spatial shifting** = esecuzione di task dove l'energia è più pulita
- **comportamento adattivo** = più versioni della stessa app da “basso consumo/basse prestazioni” a “alto consumo/alte prestazioni, a seconda del carico di lavoro e del mix energetico disponibile.

Cloud Computing

Il Cloud computing prevede l'utilizzo di infrastrutture di calcolo “come servizio” da parte di più utenti e organizzazioni supporta l'esecuzione di software all'interno di server raccolti in datacenter, piuttosto che utilizzare server “on-premise” o “in-house”.

Utilizza meno server per più utenti e rende efficiente il consumo energetico e riduce potenzialmente l'impronta carbonica

- **Virtualizzazione:** consente di distribuire il lavoro su meno server condivisi tra più utenti; consente di ottimizzare l'utilizzo hardware e riduce gli sprechi
- **Elasticità:** la quantità di risorse si adatta al carico di lavoro, consentendo un utilizzo più efficiente dell'energia
- **Investimenti:** Le grandi aziende che forniscono servizi cloud possono permettersi di investire in infrastrutture più efficienti dal punto di vista energetico (energie rinnovabili, impianti di raffreddamento tecnologicamente più avanzati)
- **Spatial shifting:** potendo scegliere dove allocare le proprie risorse, i clienti dei provider Cloud possono decidere di istanziarle in base al mix energetico disponibile

Riciclo, Smaltimento, Riuso, Riparabilità

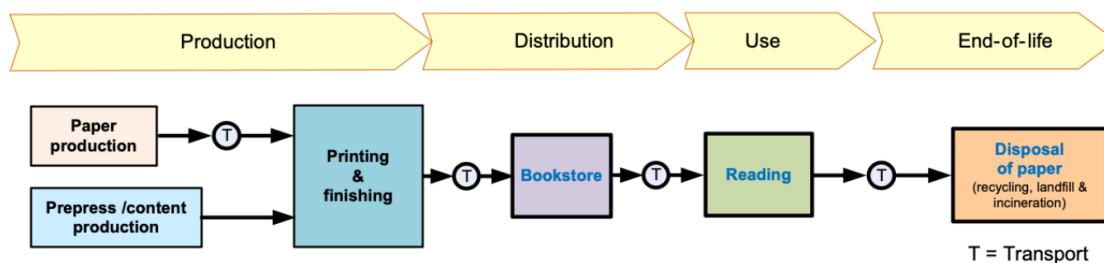
- Il **riciclo** può ridurre l'impatto ambientale dell'ICT, trasformando materiali di scarto e rifiuti in nuove risorse o beni.
- Uno **smaltimento** corretto previene la dispersione di sostanze tossiche come piombo e mercurio, evitando contaminazioni e rischi per la salute.
- Il **riuso** prolunga la vita dei dispositivi, diminuendo la necessità di nuova produzione e il consumo energetico, favorendo un'economia circolare.
- La **riparabilità** permette di aggiustare un dispositivo guasto evitando la sostituzione prematura.

Ebook VS libri

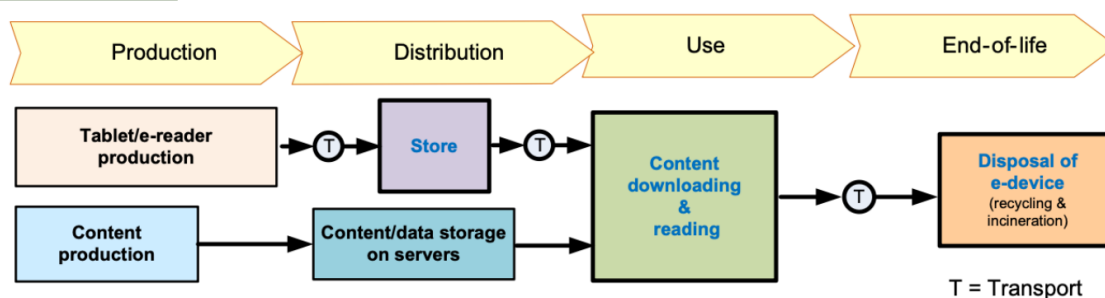
Quanti libri fanno un E-book Reader?

analizziamo il ciclo di vita di un ebook reader in maniera “olistica” e confrontiamolo con quello dei libri cartacei su impatto ambientale e impronta carbonica.

Libri



E-book reader



Facciamo un confronto considerando le materie prime, l’acqua e l’energia necessarie per produrre un libro e un e-book reader e infine consideriamo anche l’inquinamento dovuto al

- Quanti libri fanno un ebook reader? **40-50**
- Quanti libri bisogna leggere all’anno per «ammortizzare» un e-book reader su 5 anni di vita media? **8-10**

Salute

Ossidi di azoto e zolfo nocivi per la salute umana, sia la produzione di libri che di e-book reader li genera. un e-book reader ne genera 70 volte tanto.

Dismissione

Libro	E-book reader
se decomposto può inquinare le falde acquifere e generare il doppio delle emissioni rispetto all’intera produzione	possibilità di smaltimento in paesi in via di sviluppo, lavoratori bambini esposti a impatto tossico di componenti...
può essere riciclato, prestato, regalato o donato a una biblioteca	si possono recuperare molti materiali se riciclato correttamente

Bitcoin

Bitcoin, nel 2008, è stata la prima tecnologia basata su blockchain.

Blockchain è un libro mastro distribuito in grado di registrare e validare transazioni in assenza di un'entità centrale.

In particolare, ogni blockchain è:

- **Distribuita** (tutti i nodi partecipanti ne conservano una copia per trasparenza)
- **Immutabile** (i record nella catena non possono essere modificati né cancellati)
- **Marcata** temporalmente (ogni transazione ha un timestamp)
- **Unanime** (tutti i nodi partecipanti riconoscono la validità delle transazioni)
- **Anonima** (l'identità dei partecipanti non è rivelata)
- **Sicura** (tutti i record vengono crittati individualmente)
- **Programmabile** (per mezzo di SmartContracts).

Proof of work

Il Proof of Work (PoW) è il meccanismo di consenso originale utilizzato da criptovalute come Bitcoin per convalidare le transazioni e proteggere la *blockchain*.

Funzionamento: I *minatori* competono risolvendo un problema crittografico complesso (trovare un *nonce*, numeri primi) che richiede una potenza di calcolo estremamente elevata e dispendiosa. Il primo che risolve il problema convalida il nuovo blocco di transazioni, lo aggiunge alla catena e riceve una ricompensa. La verifica della soluzione da parte degli altri nodi è invece rapida e facile.

Problema Ecologico: La sicurezza del PoW è intrinsecamente legata al consumo massiccio e crescente di energia elettrica, necessario per la potenza di calcolo richiesta. Questo si traduce in un'alta impronta di carbonio, specialmente quando si utilizzano combustibili fossili, sollevando serie preoccupazioni ambientali.

Per questo motivo, molte nuove *blockchain* o quelle aggiornate (come Ethereum) stanno migrando verso meccanismi di consenso a minor consumo energetico, come il Proof of Stake (PoS).

Problema della centralizzazione: dei Rischi di centralizzazione (attacco del 51%)

Proof of Stake

Il Proof of Stake (PoS) è un meccanismo di consenso alternativo al PoW, progettato per essere molto più efficiente dal punto di vista energetico.

Funzionamento: Invece di competere per risolvere complessi problemi matematici causa di un alto consumo energetico, i partecipanti sono selezionati per convalidare i nuovi blocchi in base alla quantità di criptovaluta che hanno "messo in gioco" (o **stake**) come garanzia.

- **Stake:** I partecipanti o validatori devono bloccare una quantità minima di moneta come *stake* per avere la possibilità di essere scelti.
- **Selezione:** L'algoritmo seleziona un validatore in modo pseudo-casuale (spesso ponderato in base allo stake e all'età dello stake) per creare il prossimo blocco.
- **Validazione e Ricompensa:** Se il validatore propone un blocco valido, riceve una ricompensa (commissioni di transazione).

- **Penalità (Slashing):** Se un validatore tenta di convalidare transazioni fraudolente o agisce in modo malevolo, perde parte o tutto il suo stake.

Crypto Climate Accord (April 2021) → Si rende necessario accelerare lo sviluppo di strategie per superare i problemi ambientali nel settore delle criptovalute.

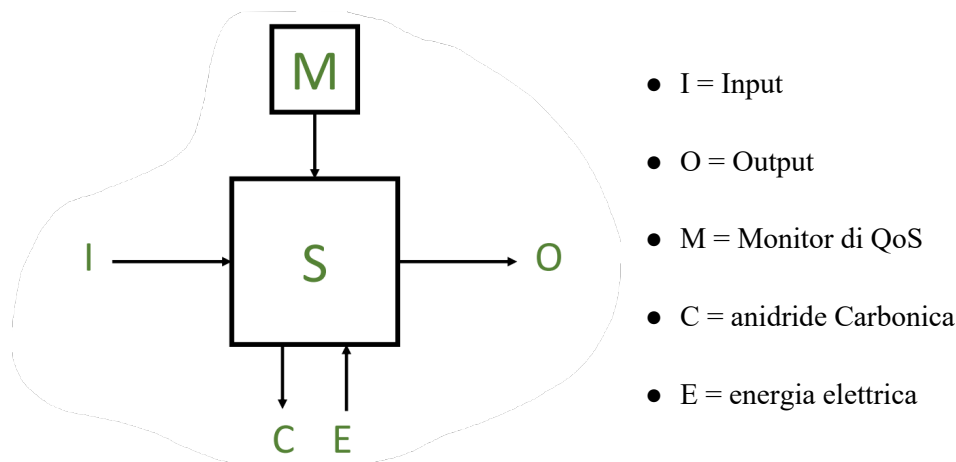
Sviluppo sostenibile e misure di performance ICT

«Lo sviluppo sostenibile è uno sviluppo che soddisfi i bisogni del presente senza compromettere la possibilità delle generazioni future di soddisfare i propri.»

- Brundtland report

Sviluppo che limiti l'impatto sull'ambiente, combatta la povertà ed eviti catastrofi ecologiche.

Modello per misurare le performance di un sistema ICT



Il monitoraggio è fondamentale per garantire il corretto funzionamento e necessita di un modello.

Service Level Agreement → contratti di servizio, descrivono la corrispondenza tra metriche ICT tecnicamente monitorabili e metriche di performance del cliente.

Qualità del servizio: effetto collettivo delle performance che determina il grado di soddisfazione degli utenti del servizio stesso.

efficienza energetica → #calcoli / Energia consumata

Legge di Koomey → La quantità di calcoli per Joule di energia utilizzata raddoppia all'incirca ogni 1,5 anni (dal 2010 ogni 2,5 anni).

Power usage effectiveness

$$PUE = \frac{P_{IT} + P_{non-IT}}{P_{IT}}$$

Datacentre Infrastructure Efficiency (DCiE)

$$DCiE = \frac{P_{IT}}{P_{IT} + P_{non-IT}} = \frac{1}{PUE}$$

Sia PUE che DCiE sono influenzati da:

- Utilizzo del sistema (nel tempo e nello spazio)
- Età e progettazione del sistema
- Efficienza complessiva del sistema

PUE	DCiE	Level of Efficiency
3.0	33%	Very Inefficient
2.5	40%	Inefficient
2.0	50%	Average
1.5	67%	Efficient
1.2	83%	Very Efficient

Consumo energetico, emissioni e riciclabilità di un sistema ICT

Nel modello matematico di Drouant et al le fasi da considerare

- Produzione/manifattura
- Trasporto
- Uso & gestione
- Dismissione
- Riciclo, riuso

Energia di uso

$$E_u = PUE \int_{t=0}^{t=1 \text{ anno}} P_u(t) dt \simeq PUE \sum_{i \in S} \varepsilon_i h_i$$

Emissioni di CO₂

$$C = \frac{\alpha_i}{\tau_i} E_i + \frac{\alpha_u}{\tau_u} E_u + \frac{\alpha_f}{\tau_f} E_f = \sum_{s \in \{i,u,f\}} \frac{\alpha_s}{\tau_s} E_s$$

$$C = C_i + C_u + C_f \simeq \sum_{s \in \{i,u,f\}} \sum_{j \in F} \frac{\alpha_j p_{s,j}}{\tau_{s,j}} E_s$$

Considerando anche la percentuale del tipo di energia utilizzata

Riciclabilità

$$R = \frac{\sum_{i \in \bar{S}_r} \rho_i}{|\bar{S}_r|}$$

La potenza assorbita da uno **Switch**

$$P_u = \phi + \sigma \sum_{i \in \text{ActivePorts}} \min\{1, \delta \omega_i\} + \sum_{j \in \text{IdlePorts}} \sigma_{idle}$$

dove:

- ϕ è la potenza base (in stato *idle*) dello *switch*,
- σ è la potenza di ciascuna porta dello *switch* attiva e a pieno carico,
- δ è l'aumento della potenza assorbita dovuto al traffico (p.e. 18),
- ω_i è il carico di ciascuna porta tra 0 e 1,
- σ_{idle} è la potenza di ciascuna porta dello *switch* collegata ma non usata.

Consumo delle reti Neurali

Per ridurre le emissioni, si possono per esempio considerare switch che consumano meno per essere prodotti/smaltiti o alimentare il nostro sistema con fonti di energia rinnovabili.

Reti Neurali

Embodied carbon → Sono emissioni dovute alla produzione e al trasporto dell'hardware stesso, che quindi sono «integrate» nell'hardware

Rete neurale → algoritmi di apprendimento automatico per associare etichette a dati in ingresso.

Addestramento

L'addestramento delle reti neurali avviene in maniera distribuita sfruttando server equipaggiati con GPU o TPU dedicate e può impiegare oltre un mese per essere completato.

Dispiegamento

I modelli neurali addestrati sono poi «dispiegati» nei datacentre e forniti come servizi ai clienti finali.

In termini energetici, l'addestramento pesa circa il 9% sull'intero ciclo di vita, mentre in termini di emissioni prodotte pesa circa il 3%.

80- 90% del consumo energetico è relativo alla fase di **dispiegamento**.

L'embedded carbon della rete neurale supera il 60% nelle nostre stime.

Probabilmente è una sovrastima perché si basa su dati al caso pessimo per hardware HPE (specializzato).

Quattro buone pratiche (da Google)

- **L'adozione** di queste buone pratiche nell'ML potrebbe rallentare e ridurre l'impronta delle reti neurali:
- **Modelli** – l'uso di modelli più efficienti (sparsi invece che densi) può ridurre di un fattore 5-10 i calcoli
- **Macchine** – l'uso di hardware più efficiente può migliorare la performance per Watt di un fattore 2-5

- **Meccanizzazione** – l'uso di risorse Cloud può ridurre di un fattore 1.4-2 il consumo energetico
- **Mappa** – la scelta di location con energia più pulita può ridurre l'impronta di CO2-eq di un fattore 5-10

Ingegneria del Software Sostenibile

Il triangolo di ferro: tempo, costi e qualità non sempre raggiungibili tutti insieme.

Modello GREENSOFT

Il modello **GREENSOFT** rappresenta un punto di riferimento per l'ingegneria del software sostenibile, definendo il software verde e sostenibile come quel prodotto i cui impatti negativi diretti e indiretti su economia, società, esseri umani e ambiente sono ridotti al minimo, e che possibilmente contribuisce positivamente allo sviluppo sostenibile.

L'ingegneria del software sostenibile diventa quindi l'arte di sviluppare questi prodotti valutando e ottimizzando continuamente i loro impatti durante tutto il ciclo di vita, dalla creazione allo smaltimento.

Per comprendere come il software impatti sul mondo, il modello classifica gli effetti su tre livelli distinti.

- Gli **effetti di primo ordine** sono quelli diretti, legati al ciclo di vita della tecnologia stessa, come il consumo di energia durante l'uso del software o le risorse consumate per produrre l'hardware necessario a farlo girare.
- Gli **effetti di secondo ordine** nascono invece dall'applicazione del software in altri contesti e possono portare benefici come la dematerializzazione o l'ottimizzazione di processi logistici.
- Gli **effetti di terzo ordine** sono quelli sistemici a lungo termine, che riguardano i cambiamenti negli stili di vita o nelle strutture economiche, come la diffusione del telelavoro.
 - **L'effetto rebound** che si verifica quando l'aumento di efficienza ottenuto grazie al software porta paradossalmente a un consumo maggiore di risorse, annullando i risparmi previsti.

Criteria e metriche per la sostenibilità

Si parte dalle metriche di qualità comuni del software, come la modificabilità e la portabilità, che vengono rilette in chiave ecologica perché un software facile da aggiornare o adattare prolunga la vita utile dell'hardware.

Vi sono poi i criteri direttamente correlati, che misurano impatti specifici come l'efficienza energetica del codice e l'obsolescenza hardware indotta, ovvero quanto il software spinga a cambiare dispositivi anzitempo.

Infine, si considerano i criteri indirettamente correlati, utili a valutare le conseguenze sistemiche e di secondo ordine dell'uso del software.

Dal punto di vista operativo, il modello propone delle procedure differenziate per i vari attori coinvolti, definendo linee guida specifiche non solo per chi sviluppa il software, ma anche per chi lo acquista, per gli amministratori di sistema che lo gestiscono e per gli utenti finali che lo utilizzano.

Greesoft e AGILE

Il modello Greensoft si integra perfettamente con il metodo AGILE. Includendo revisioni periodiche focalizzate sulla sostenibilità per valutare il lavoro svolto e pianificare quello futuro, l'uso di un diario di sostenibilità per tracciare le decisioni prese, la valutazione dell'impatto ambientale del processo di sviluppo stesso e, infine, una retrospettiva conclusiva per analizzare i risultati complessivi e apprendere nozioni utili per i progetti successivi.

Green Software Foundation

- **Carbon efficiency:** Scrivere applicazioni che portino lo stesso valore agli utenti, emettendo meno gas serra.
- **Energy efficiency:** Scrivere applicazioni che portino lo stesso valore agli utenti, consumando meno energia.
- **Carbon-awareness:** Scrivere applicazioni che adattino il proprio comportamento al mix energetico corrente (time-shifting, spatial-shifting, adattività).
- **Efficienza hardware:** Ammortizzare il carbonio integrato dell'hardware, usandolo più a lungo e aumentandone il rate di utilizzo (attenzione però alla proporzionalità energetica!).

Bin packing

Mettere N oggetti di taglie diverse in K contenitori di capacità C , minimizzando il numero di contenitori.

Scelta del server:

- **Esaustiva:** esplorare tutte le soluzioni possibili e sceglierne una ottima
- **Next-fit:** provare a piazzare ogni servizio sul server su cui abbiamo piazzato l'ultimo; altrimenti aggiungere un nuovo server
- **First-fit:** provare tutti i server già usati e scegliere il primo in cui il servizio può stare; altrimenti aggiungere un nuovo server
- **Best-fit:** scegliere il server con minor spazio a disposizione che può ospitare il servizio
- **Worst-fit:** scegliere il server con maggior spazio a disposizione che può ospitare il servizio

Scelta del servizio da piazzare: ordine decrescente: dal più grande al più piccolo per evitare di non avere spazio per i servizi che richiedono più risorse

Ruoli all'interno di un datacenter: caso di studio aggiornamento di un SO



Utente

- desiderano un software veloce, leggero e frequente (disponibilità di nuove feature)
- dovrebbero essere consapevoli del proprio impatto ambientale, richiedere software di qualità per evitare patch frequenti e richiedere che la sostenibilità non alteri la QoS.

Architetto

Coloro che determinano i requisiti funzionali (consegna corretta degli aggiornamenti al cliente) e non-funzionali (99,9% di disponibilità del servizio, scalabilità, velocità) del sistema e che progettano le interazioni tra componenti.

Sostenibilità → si può includere nei requisiti non-funzionali tramite;

- **CDN**, mantiene i dati (dell'aggiornamento) in «punti di presenza» locali, diminuendo la distanza verso il cliente e l'utilizzo di banda e consentendo il caching dei contenuti popolari
- **Microservizi**, per avere maggiore scalabilità
- **Bin packing**, per minimizzare il numero di server per far girare i servizi che compongono il sistema, l'architetto può disegnare componenti più piccole per evitare frammentazione.
- **Carbon-awareness** come funzionalità del software

Sviluppatore

Coloro che sviluppano il sistema software seguendo i requisiti funzionali e non-funzionali specificati dall'architetto.

Può:

- Scegliere linguaggi e/o framework opportuni anche per favorire bin packing

- Disaccoppiare implementazione da infrastruttura in modo da facilitare la migrazione a Cloud provider differenti
- Strumentare e misurare per migliorare, utile scegliere KPI che siano rapporti tra quantità
- Ridurre lo spreco di risorse durante lo sviluppo (p.e. con CI/CD)

Operatore

Gestisce l'infrastruttura e garantisce la disponibilità dell'applicazione

Cosa può fare:

- Fare scelte infrastrutturali che garantiscano performance e facilitino il bin packing (virtualizzazione, container, serverless)
- tenere di conto della località dei dati, usando se possibile istanze «spot» (AWS EC2 Spot Instances)
- Monitorare (KPI infrastrutturali vs KPI aziendali), log, backup (ma non troppo)
- Automatizzare lo scaling dei servizi (auto-scaling)
- Ottimizzare l'uso delle risorse

Quadro normativo

Le leggi e i regolamenti possono favorire una transizione sostenibile, «tassare» le esternalità negative (dallo sviluppo alla dismissione), obbligare alla trasparenza per creare consapevolezza tra i clienti.

L'Unione Europea ha definito una direttiva su «Corporate sustainability reporting», all'interno del Green Deal europeo, che obbliga le grandi aziende a relazionare sulla sostenibilità ambientale.

Dato che le leggi empiriche fondamentali come la legge di Moore e lo scaling di Dennard stanno rallentando dai primi anni duemila non possiamo più affidarci esclusivamente al miglioramento biennale dell'efficienza hardware per ridurre le emissioni di carbonio.

L'obiettivo dei servizi software di nuova generazione definiti carbon-aware deve essere quindi quello di eseguire la maggior parte del lavoro computazionale nei momenti in cui l'intensità di carbonio della rete elettrica è più bassa.

Carbonstat

Carbonstat fornisce una metodologia per progettare servizi adattivi e uno schema di ottimizzazione per configurarli.

Dal punto di vista dell'ingegneria del software la soluzione si basa sull'utilizzo del design pattern **Strategy** che consente di sviluppare un servizio in molteplici varianti o strategie intercambiabili.

Questo pattern permette di orchestrare algoritmi alternativi che offrono la stessa funzionalità ma con diversi compromessi tra tempo di esecuzione e qualità dell'output ad esempio distinguendo tra strategie a basso medio e alto consumo dove un tempo di esecuzione più lungo corrisponde a un consumo energetico maggiore.

Il funzionamento di Carbonstat può essere paragonato a un termostato per il carbonio che riceve in input i tempi di esecuzione e la qualità delle strategie disponibili insieme alle previsioni per le ventiquattro ore successive riguardanti sia l'intensità di carbonio che il tasso di richieste al servizio oltre a una soglia di qualità media target indicata con epsilon.

L'output del sistema è un piano di assegnazione che stabilisce quale strategia utilizzare per ogni slot temporale con l'obiettivo di minimizzare le emissioni complessive di carbonio pur mantenendo l'errore medio al di sotto della soglia prefissata.

I risultati sperimentali ottenuti confrontando questo approccio con configurazioni statiche o ingenuie dimostrano che l'adattamento dinamico permette di bilanciare consistentemente le prestazioni del servizio con l'impronta ecologica ottenendo **una riduzione delle emissioni compresa tra 8-50%** a seconda degli scenari considerati.

In conclusione questo lavoro rappresenta la prima proposta che combina metodologie di ingegneria del software con l'ottimizzazione per creare servizi carbon-aware che considerano congiuntamente intensità di carbonio consumo energetico tassi di richiesta e qualità del servizio senza dipendere da componenti di comunicazione esterni.

Green Coding

Abbiamo utilizzato uno strumento chiamato GreenMeter per stimare consumi ed emissioni eseguendo un mix bilanciato di operazioni CRUD.

Analizzando i risultati sperimentali emerge chiaramente che per quanto riguarda le liste l'implementazione **ArrayList** risulta **significativamente più sostenibile rispetto a LinkedList**. Nello specifico ArrayList ha completato le operazioni in un tempo notevolmente inferiore producendo una quantità di emissioni molto più bassa rispetto a LinkedList grazie all'utilizzo di un vettore ad accesso diretto che velocizza la lettura dei dati.

Passando all'analisi degli insiemi la struttura dati che si è dimostrata più sostenibile in assoluto nei nostri esperimenti è **HashSet** che ha superato **TreeSet** sia in termini di tempo di esecuzione che di **emissioni** prodotte grazie alla sua complessità costante $O(1)$ per le operazioni CRUD.

Sebbene le strutture di tipo List offrano performance peggiori rispetto ai Set esse sono necessarie quando bisogna gestire elementi duplicati. *Nel confronto tra liste sebbene ArrayList sia generalmente più efficiente LinkedList implementa l'interfaccia Queue che può essere indispensabile in alcuni scenari.*

Analogamente anche se HashSet è più efficiente energeticamente *TreeSet* diventa la scelta corretta quando è necessario mantenere i dati ordinati poiché ordinare esternamente un HashSet comporterebbe un costo computazionale aggiuntivo.

La scelta della struttura dati migliore non è assoluta ma deriva sempre da un compromesso tra i requisiti funzionali l'efficienza temporale e la sostenibilità ambientale scegliendo l'implementazione più adeguata tra quelle disponibili

EnerJ

Qualificatori di tipo per dichiarare dati che possono essere soggetti ad approssimazioni

Il compilatore:

- mappa variabili approssimate su storage a basso voltaggio
- usa operazioni a basso consumo su tali variabili
- usa algoritmi efficienti forniti dal programmatore su tali variabili.

Gli esperimenti dimostrano una **riduzione** di almeno il **10-15%** del consumo in presenza di *hardware specifico*.

Il modello è sicuro grazie agli endorsement espliciti.

- EnerJ utilizza qualificatori di tipo per decidere quali variabili approssimare e quali no.
- L'implementazione garantisce la correttezza dei calcoli sulle variabili precise e un approccio «best-effort» su quelle approssimate.
- EnerJ utilizza l'overloading per implementare operazioni approssimate.
- EnerJ è retrocompatibile con Java non approssimato

non si può assegnare una variabile precisa una variabile approssimata.

- ciò previene un flusso diretto da dati che possono essere approssimati a dati che devono essere precisi

```
@Approx int a = ...;  
@Precise int p;  
@Approx int a = ...; @Precise int p; p = a; //sbagliato
```

Endorsement → rendere un valore approssimato, preciso.

Il programmatore «certifica» che la conversione non produrrà errori sulla parte precisa del codice.

nelle guardie condizionali serve un valore @Precise, eventualmente si può usare endorse()

non si può assegnare una variabile precisa un valore di una variabile approssimata

```
@Approx int a = ...;  
int p; // preciso di default  
p = endorse(a); // ok
```

EnerJ fornisce operatori approssimati tramite l'overloading degli operatori standard.

Assegnamento bidirezionale (se a sinistra c'è una variabile approssimata, il valore a destra viene approssimato, anche se è un valore preciso o una somma tra valori precisi).

Gli oggetti EnerJ possono estendere la classe **Approximable**.

- Per le classi @Approximable si possono definire funzioni approssimate
- @Context → dipende dal “contesto” cioè da come è stato istanziato l'oggetto relativo alla classe, può assumere valore @Precise o @Approximable.

Array: i valori nelle celle di un array possono essere approssimati, la lunghezza e gli indici sono sempre precisi.

altri tipi:

@Top → supertipo comune di @Approximable e @Precise

@Lost → l'etichetta viene utilizzata per indicare che il sistema non può determinare correttamente la precisione di un dato

Un tipo preciso è un sottotipo del corrispondente tipo approssimato.

FEnerj

FEnerj → Semantica formale di Enerj.

Il **Subtyping** è il meccanismo fondamentale che garantisce la sicurezza del sistema pur permettendo il risparmio energetico.

La relazione di **subtyping** si basa su una gerarchia di etichette che indicano quanto un dato sia affidabile:

- **top**: È il super-tipo comune a approx e precise.
- **precise** <: **approx**: Un tipo preciso è considerato un sottotipo del corrispondente tipo approssimato. Questo significa che un dato preciso può essere usato dove ne è richiesto uno approssimato, ma non viceversa (per garantire la sicurezza del flusso d'informazione).
- **lost**: Viene utilizzato quando il sistema non può determinare correttamente la precisione di un dato. Qualunque qualificatore diverso da top è considerato un sottotipo di lost.

Regole di Tipo

la guardia sia di un tipo primitivo precise e e_1 e e_2 devono essere dello stesso tipo T.

Tecniche di riduzione del consumo hardware

- **Voltaggio adattivo** (da high-power a low-power secondo il tipo)
 - fino a 30% di riduzione del consumo con 1% di errore in più

- **Riduzione della mantissa** nelle operazioni floating point
 - passare da 24 bit a 8 riduce il consumo del 78%
- Riduzione della **frequenza di aggiornamento** della DRAM
- Riduzione del **voltaggio della SRAM** (cache e registri)

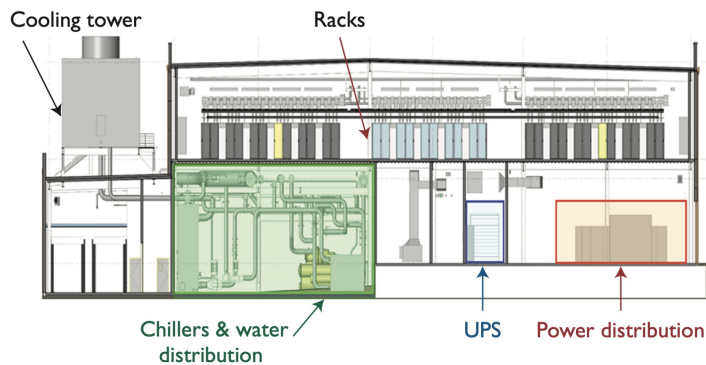
Il modello basato sull'hardware mostra un potenziale **risparmio energetico del 10%-50%**. Il migliore modello è un approccio cauto riguardo le approssimazioni per tenere bassa la percentuale di errori.

DataCenter

Massachusetts Green High-Performance Computing Center

Caso di studio di un datacenter che sfrutta il free-cooling per ridurre i consumi e quindi le emissioni.

Struttura del data center



Alimentazione

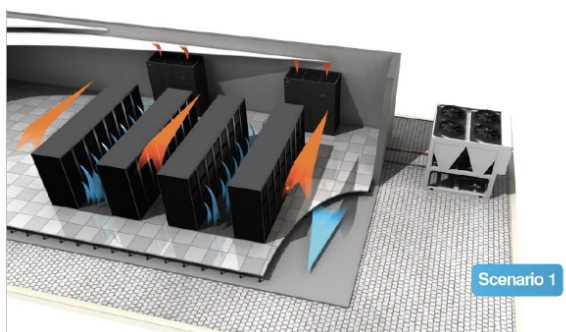
Automatic Transfer Switch (ATS) per rilevare assenza di corrente e mettere in moto dei generatori di backup a diesel attraverso un apposito switchgear.

UPS rotanti a volano per immagazzinare energia evitando batterie inquinanti

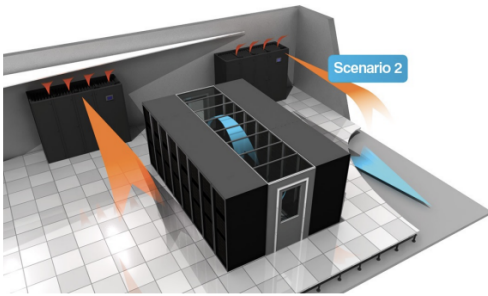
- **pro:** non ci sono da smaltire inquinanti chimici (p.e. piombo)
- **contro:** minore durata e capacità; necessità di far girare i volani anche in stand-by
- **compromesso:** solo il 20% dei rack ha un UPS

Raffreddamento

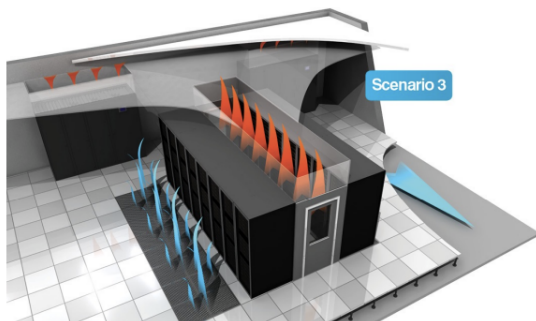
Nei datacenter tradizionali si impiegano refrigeratori (**chiller**) collegati a unità di condizionamento nella sala macchine dette Computer Room Air Conditioning (**CRAC**).



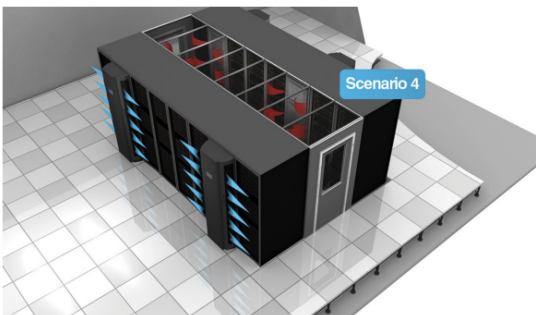
Corridoio aperto: le unità CRAC convogliano l'aria fredda dal basso verso l'alto attraverso griglie nel pavimento flottante davanti a ciascun rack.



Contenimento del corridoio freddo: l'aria fredda è contenuta nello spazio davanti ai rack e l'aria calda di torna all'unità CRAC.



Contenimento del corridoio caldo: l'aria fredda viene indirizzata sulla parte anteriore dei rack e l'aria calda viene convogliata verso le unità CRAC dal soffitto.



Contenimento del corridoio caldo + in-row cooler: Come il precedente con in più l'utilizzo di in-row cooler (IRC) montati tra i rack, negli armadi

- è usato nel MGHPCC
- miglioramento del free-cooling

Free cooling

Il free cooling all'interno del datacenter MGHPCC è progettato sfruttando due circuiti principali di raffreddamento ad acqua

- **tradizionale:** l'acqua raffreddata dai chiller circola negli IRC che rinfrescano l'aria della sala macchine, rimuovendola dal corridoio caldo. (Soluzione classica nessuna ottimizzazione).
- **adiabatico:** l'acqua nel circuito delle torri di raffreddamento è raffreddata per evaporazione. (Risparmio di energia).

Gli IRC regolano la velocità delle ventole e il flusso di acqua refrigerata in base al calore dei rack, migliorando ulteriormente l'efficienza.

La sala macchine MGHPCC viene mantenuta a 26,7°C. Se temperatura e umidità esterne sono abbastanza basse, gli scambiatori di calore raffreddano «gratuitamente» l'acqua nel circuito di raffreddamento tradizionale.

- MGHPCC usa il 70% di free cooling all'anno

Risposta del professore sui 4 tipi di raffreddamento

Nei data center esistono quattro configurazioni comuni per la gestione dei flussi di aria fredda e calda. Nel corridoio aperto, l'aria fredda viene immessa dal pavimento senza alcuna separazione dall'aria calda emessa dai server. Non c'è separazione tra aria calda e fredda, e il movimento d'aria all'interno della sala è scarso, con conseguente scarso controllo dell'aria calda. Il contenimento del corridoio freddo isola e raffredda lo spazio anteriore ai rack. L'aria calda si disperde nella sala circostante e torna alle unità CRAC. Più efficace è il contenimento del corridoio caldo, che isola invece l'aria calda sul retro dei rack e la convoglia verso l'alto, mantenendo l'aria fredda libera nella sala. Ciò permette temperature di scarico più elevate, utili per il free cooling. Infine, il contenimento del corridoio caldo con in-row cooler aggiunge unità di raffreddamento tra i rack stessi, offrendo un controllo più preciso e temperature di mandata (fino anche a 27 gradi Celsius) e di scarico più elevate migliorando ancora le opportunità di free cooling.

Monitoraggio

- Viene monitorata l'**energia consumata** a livello di Rack, Pod e DataCenter.
- Contatori separati per l'**energia di raffreddamento**
- Monitoraggio dei sistemi meccanici
- Misura della **temperatura** nella sala macchine nella parte calda dei rack.

$$WUE = \frac{\text{water used [l]}}{E_{IT} [\text{kWh}]} \quad \text{Usò dell'acqua}$$

$$\text{Impronta di CO}_2 \text{ CUE} = \alpha \cdot \text{PUE} \left[\frac{\text{kgCO}_2\text{-eq}}{\text{kWh}} \right]$$

Le emissioni integrate (embodied carbon) costituiscono il 50% delle emissioni durante l'intero ciclo di vita.

E-Waste e Clicking clean

convenzione di Basilea

Obiettivo: ridurre i movimenti di rifiuti pericolosi (compreso il RAEE) da Paesi sviluppati verso Paesi in via di sviluppo.

- 191 Paesi sottoscrittori
- USA, Timor Est, Figi, Haiti, Sud Sudan non hanno ratificato la convenzione

50 milioni di E-Waste ogni anno.

40% dei rifiuti nell'UE viene riciclato correttamente.

EU WEEE

La WEEE Directive (Waste Electrical and Electronic Equipment), risalente al 2002, richiede la gestione separata dei rifiuti elettronici, imponendo obiettivi stringenti su raccolta, trattamento e riciclo, al fine di contrastare il traffico illegale di e-waste e armonizzare i diversi sistemi nazionali.

EU directive on repair of goods [2024]

Che obbliga i produttori a garantire la riparabilità dei dispositivi in tempi e costi ragionevoli, vieta pratiche che ostacolano la riparazione, prevede l'estensione della garanzia legale post-intervento e promuove la creazione di piattaforme digitali per facilitare l'accesso ai servizi di riparazione.

Obsolescenza programmata

L'obsolescenza programmata si verifica quando un prodotto è deliberatamente progettato per avere una durata più breve nel tempo. Questo significa che la sua vita utile è intenzionalmente limitata per spingere i consumatori a sostituirlo più frequentemente. Esempi storici includono il cartello Phoebus, che nel 1925 standardizzò la vita delle lampadine a incandescenza a sole 1.000 ore, rispetto alle 2.500 precedenti. Altri esempi moderni includono batterie non sostituibili e cartucce di stampante che sono rese incompatibili prematuramente, oppure aggiornamenti software che compromettono le funzionalità di dispositivi più datati, come accaduto a Apple e Samsung, multate per aver causato problemi o ridotto le prestazioni di alcuni telefoni cellulari.

Obsolescenza percepita

L'obsolescenza percepita, invece, si manifesta quando un cliente si convince di aver bisogno di un prodotto aggiornato, anche se quello che possiede funziona ancora perfettamente. Questo fenomeno è spesso alimentato dalle tendenze di marketing e dalle rapide innovazioni tecnologiche che creano il desiderio di possedere l'ultima versione di un dispositivo. Un chiaro esempio è la frequente sostituzione dei telefoni cellulari da parte dei consumatori, spinti non dalla rottura o dal malfunzionamento, ma dalla percezione che il proprio dispositivo sia "passato di moda" o meno performante rispetto ai nuovi modelli, contribuendo in modo significativo alla generazione di rifiuti elettronici.

Ostacoli a un Internet alimentato al 100% da fonti rinnovabili

- 1) mancanza di trasparenza (riluttanza a pubblicare i dati energetici)
- 2) mancanza di accesso alla fornitura di energia rinnovabile (RE)

Report CleaningClean

- Apple leader per il terzo anno consecutivo tra gli operatori di piattaforma
- Apple e Google continuano ad essere leader del settore per la corrispondenza tra la loro crescita e una fornitura equivalente o maggiore di energia rinnovabile, ed entrambe continuano a spingere i governi e i fornitori ad aumentare l'accesso all'energia rinnovabile per le loro operazioni
- Switch è leader tra gli operatori di collocazione Cloud per i suoi sforzi di transizione della flotta di data center verso le energie rinnovabili il più rapidamente possibile.
- Non è chiaro se Amazon Web Services (AWS) sia effettivamente sulla via dell'energia rinnovabile (mancanza di trasparenza e rapida crescita in Virginia e in altri mercati ampiamente serviti da energia sporca).
- Le aziende meno trasparenti, come AWS, Tencent, LG CNS, Baidu, sono anche tra le più dominanti nei rispettivi mercati.

