# CS528
# EE-RTS
# &
# Intro to Cloud System

A Sahu

Dept of CSE, IIT Guwahati

# Outline

- **Real Time Task System**
  - EDF
  - Energy Efficient Scheduling
- Introduction to Cloud System
  - Service Model and Utilities
  - Virtualization
  - Economic Model

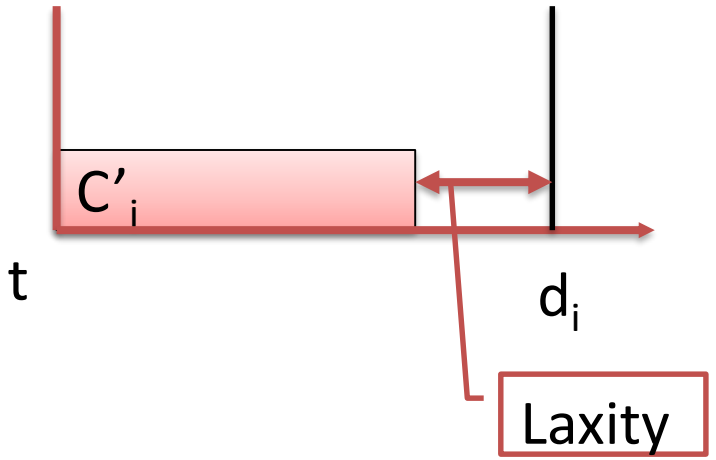
# Periodic Tasks

- Necessary schedulability test
  - Sum of utilization factors $\mu_i$ must be less than or equal to n, where n is the number of processors
  - **$\mu = \Sigma (c_i / p_i) <= n$**
  - $\mu_i$ = Percentage of time the task $T_i$ requires the service of a CPU

# Periodic Task: Real Time Scheduler

Assumptions & Definitions

- Tasks are periodic
- No aperiodic or sporadic tasks
- Job (instance) deadline = end of period
- Tasks are preemptable




- Laxity of a Task

$$T_i = d_i - (t + c_i')$$

where di: deadline;

t : current time; $c_i'$ : remaining computation time.

# Earliest Deadline First (EDF)

- **Dynamic Scheduling**
- Task with the smallest deadline/laxity is assigned the highest priority. EDF or **Least Laxity First (LLF)**
  - At any time, the highest priority task is executed.
- **Schedulability check (off-line)**
  - A set of ***n*** tasks is schedulable on a uniprocessor by the EDF algorithm if the processor utilization.

$$\sum_{i=1}^{n} \frac{C_i}{P_i} \leq 1$$

- This condition is both necessary and sufficient.

# RT task: energy minimization

- Given a system of ***n*** periodic tasks ***T={$\tau_1$, $\tau$2, .. $\tau_n$}*** and one ***Dynamic Volt-Freq Scaling Processor***
- With ***F={0, $f_1$,$f_2$,$f_3$,…$f_{max}$}*** finite number of freq
- And ***$f_i < f_{i+1}$.***
  - Assume the task system satisfy $\sum(wc_i/p_i) < 1$
    - **at $f_{max}$,** $wc_i$=worst case compute time of $i^{th}$ task
  - It ensure all the period tasks are schedulable without missing deadline if we run processor at $f_{max}$

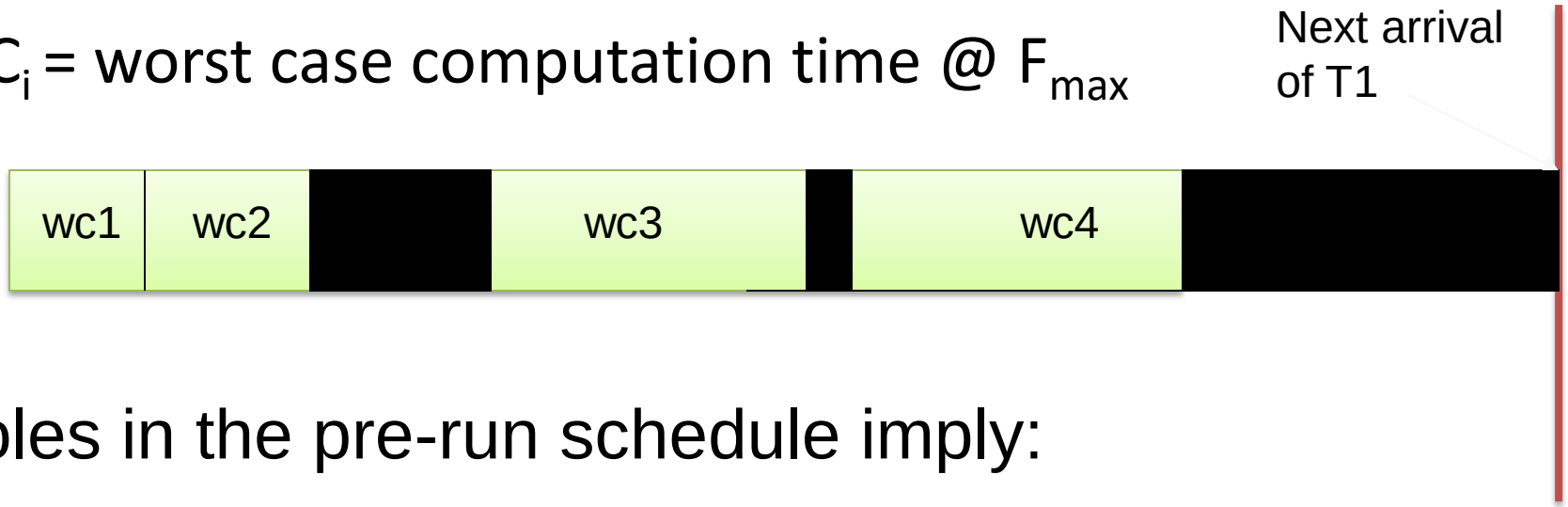
# RT task: energy minimization

- Design an efficient and elegant way to reduce the power/energy consumption ($E=f^3t$)
  - Hidden assumption: With out missing deadline of any task
- Number of processor is 1
- You may assume: all the periodic tasks arrive at time 0
- Deadline of task is period of task

# Frequency Scaling EDF: Motivation

Pre-run schedule with holes

$WC_i$ = worst case computation time @ $F_{max}$




Holes in the pre-run schedule imply:

**EDF Test:**

$$\sum(wc_i/p_i) < 1 \quad \text{at frequency} = F_{max}$$

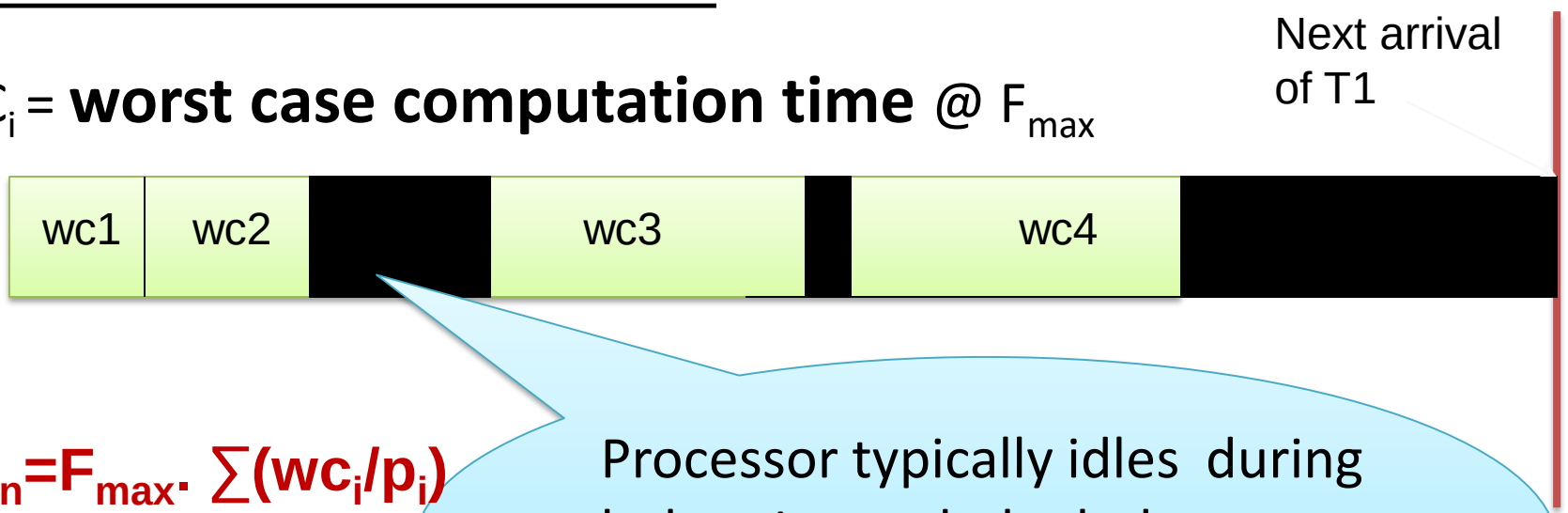
In other words, whenever

**$\sum(wc_i/p_i) < 1$ there are holes in the EDF schedule**

# Frequency Scaling EDF: exploiting holes

Pre-run schedule with holes

$WC_i$ = **worst case computation time** @ $F_{max}$




$F_{run}=F_{max}\cdot \sum(wc_i/p_i)$

Processor typically idles during holes. Instead, the holes can be exploited to slowdown the processor to save energy

How to do it ?
You need design an efficient and elegant way to reduce the Energy Consumption ?

# Introduction to Cloud Computing System

# Outline

- What is Cloud Computing?
- (HPC, Data Center, Grid) Vs Cloud
- Virtualization
- Advantage of Cloud System : User Prospects
- Dis-advantage of Cloud System : User Prospects

# HPC/Grid Vs Cloud

- Grid/HPC : Self owned
  - Too costly : CAPEX (Capital Expenditure)
  - IITG HPC Example: 10.6 Crores, 3800 J cores
  - OPEX : Operational cost, AC, electricity, AMC
- Cloud : User and Owner are Separated
  - Lets of VC own the HPC but users uses as RENT
  - User get cheaply at need time
  - Owner get a lots of demand for USE
  - Win-Win for Both, Example Public BUS
- OLA, UBER, Any Taxi Service
  - Get a CAR and used for Taxi

# Utility Model

- Do we require to own a car to ride?
- Rent a CAR for 1 month (schedule your self how you will use)
- Rent a CAR  for 1 Day  (schedule your self how you will use)
- Use Pickup or Drop service, personalized
  - Src-Dst defined
- Use shared services: Piggy back with others

# Utility Computing

- Long been a vision
- Grid computing failed to really catch on
- Technology advances as well as a viable business model have helped Cloud Computing catch on
- Cloud Computing allows for fuller utilization of hardware
- Energy consumption is turning into a major issue

# Cloud Computing Economic Benefits

- Most identifiable economic benefit of cloud computing is
  - direct cost savings, which occur from changes within the organization and the data centers that house the IT infrastructure.
  - Supply Side – Large scale data centers lower cost due to superior buying power

# Cloud Computing Economic Benefits

- Other economic benefit of cloud
  - Demand Side – Allowing multiple users across varying industries regions & time zones allowing for server utilization
  - Multi-user efficiency – Increasing # of users lowers server cost per tenant
  - Data center efficiency – Advanced data center designs reduce power loss and improved cooling

# What is Cloud Computing?

- Cloud Computing is a general term
- **Used to describe a new class of network based computing that takes place over the Internet**,
  - Basically a step on from Utility Computing
  - A collection/group of integrated and networked hardware, software and Internet infrastructure (called a platform).
  - Using the Internet for communication and transport provides hardware, software and networking services to clients

# What is Cloud Computing?

- These platforms
  - hide the complexity and details of the underlying infrastructure from users and applications
  - by providing very simple graphical interface or API (Applications Programming Interface).

# What is Cloud Computing?

- In addition, the platform provides on demand services, that are always on, anywhere, anytime and any place.
- Pay for use and as needed, elastic
  - Scale Up and Down in capacity and functionalities
- The H/W and S/W services are available to
  - general public, enterprises, corporations and businesses markets

# Cloud Summary

- Cloud computing : an umbrella term used to refer to Internet based development and services
- A number of characteristics define cloud data, applications services and infrastructure:
  - **Remotely hosted**: Services or data are hosted on remote infrastructure.
  - **Ubiquitous**: Services or data are available from anywhere.
  - **Commodified**: The result is a utility computing model similar to traditional that of traditional utilities, like gas and electricity - you pay for what you would want!