About the course /

Dobbiamo comunicare sempre in inglise, tranne alla Discussione auresa me, se vogliamo.

Comunication seu General Forum su Virtuale. Per vedeze i material degli sorsi anni: http://apice.unibo.it/xwiki/bin/view/Courses/Series/Ds

Non registros a meno one qualcuno aboia un valido motius.

Esame = progetto finale + discussione au progetto (ed eventualmen te aomande di non no apito re possiono prins singolo o gruppo in quanti teoria) FARE UN INCONTRO COL PROF POPTANO CI POUR MA GIONDUMA

TARE UN INCONTRO COL PROF PORTAND QUI UN PO'DI UES DI PROCETTO E PO'SI DECIDE MSIENE ONDIE FARE.

an program are scalare can il num al pertecip.

PLARE ATTENTI A L'ILITARE QUI OBIETTIVI DI PROLLETTO PER NON FARE QUALCOSA DI TROPPO GRANDE E METTERCI TROPPO TEMPO.

CREDO CHE CI SERAND ANCHE CONTPOLU INTERNEDI DEL PROCLETTO IN MODO CHE I PROBLEMI POSSAND ESSERE SISTEMATI PRINY DELLA DISCUISION SPIECARE DOUE LE COSE VISTE À LEZIONE VENCIONO IMPLEMENTATE E SERVITATE NEL PROCLETO, SIRIVENDONO NEL REPORT.

E disponibile a aborci dui link per i pat di libri di testo ma dobbiamo amederei via e-mail credit.

## Why Distributed Systems? F

Computational systems are now pervasive (= are everywhere) and are at the core of more artificial systems. Thus, they affect the moduling and engineering of almost all kinds of artificial systems (sometimes software, other times naraware, or even computational units and computational systems).

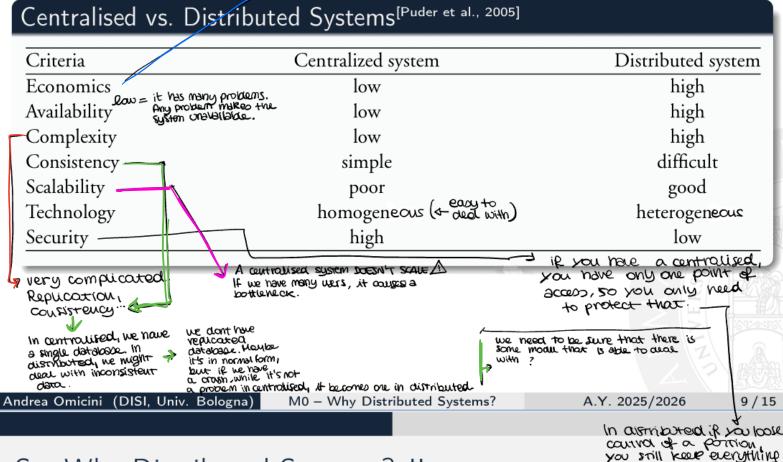
we see computations that can be:

DISTRIBUTED CONCURRENT (Controlled, triggered, Cout-onomous) they wait for something to nappen)

Not only chips and dompotation as	
units are distributed, but	
channels.	
Charmers.	
Computational devices are always	us interacting with
eachother, with humans or with	the physical environment
and its resources.	
Because artificial systems are pr	nysical, they are un predicta
ble something might happen in the e	nvivonment they are in. we
need to consider	
	7-10-11
SPATIAL DISTRIBUTION	TEMPORAL DISTRIBUTION
mainly $\rightarrow$ but also	
	examples:
	events: they are scottered,
examples of things that are	its not aways possible to have them as a
sparious distributed	
- computational units	totally-ordered set.
- communication channels - data/information/knowledge 7	ex datapage
- sensors /actuators	when computations happen
30/100/13/00/00/01/01	autonomously we can't
	know if we can order
	events or not.
	At most, we might be abu
	to create a subset of
	ordered events.
we need a way for systems to int	eract without relying on
having a DIRECT INTERACTION IN ST	PAR AND TIME.
We need distributed systems for [Ghosh, 2014]	
→ geographically distributed environments	
→ computation speedup	
→• resource sharing	
→ fault tolerance △ → he will need to design our own projects keeping faunt tolerance in mind.	
7 Iduil Luicidiice (in mind.	

So, Why Distributed Systems? I

usually not outthbutea is cheaper than distributed



So. Why Distributed Systems? II

course of a position,

FAUNT TOWERANCE = the ability to HIDE faults by using the other components that are will working.

Distributed systems can be also stud so that it one component of the system fails (or beames our connected, or partitioned) other components can replace it, thus mains the failures or at least reduce the perceived impact of failure.

When a system can hide mont of the failures, it is said to be "HIGHLY AVAILABLE!" The amount of failure that a system can sustain before failure is noticed depends on how much we are willing to accept: we might be ok with a system having a 99%. Rower tolerance, or we might decide that it is still too low.

We expect distributed systems to:

(bonave correctly, by expected) be availabe (= be live, work) but they can be unreliable

The CAP Theozen is one example of a more general thradeoff between SAFETY and liveness in unreliable systems. Enemolally, according to this theorem, it is only possible to simultaneously provide any 2 our of the 3 proporties of distributed applications:

Consistency (C)

Availability (A)

Partition tolerance (P)

Replicated data is always confishent with each other

Data is HIGHLY AVAILABLE

The system can continue to provide services to its users even when the NETWORK postitions

to nice mout of the Poilures

The 1st formulation (by Eric Brewer) states:

A distributed database potentially features 3 desirable propertion:

1 C

(2) A

(3) P

According to the theorem, any shared-data distributed system can only have up to 3/3 properties. so le con either: No & but we No & but then we no tolerance to network have P and A. partition (8) but we have C and P. have C and A. We should notice that while C and A range over a spectrum of options, P does not, and can be considered as an on/off feature DID, we can't really do without P, so this sanario is not really acceptable. > we usually have to choose between A and C. esample: LOCATION-BASED GRAMES They use suformation about the position of users to evolve and progress the gamepay. Because there are millions of players worldwide, we need P, considering mobile devices are unstable in their network connectivity, and players might move in and out of number coverage. C is needed to keep the players into the game (ex: in-game purchases) & they usually for feet A