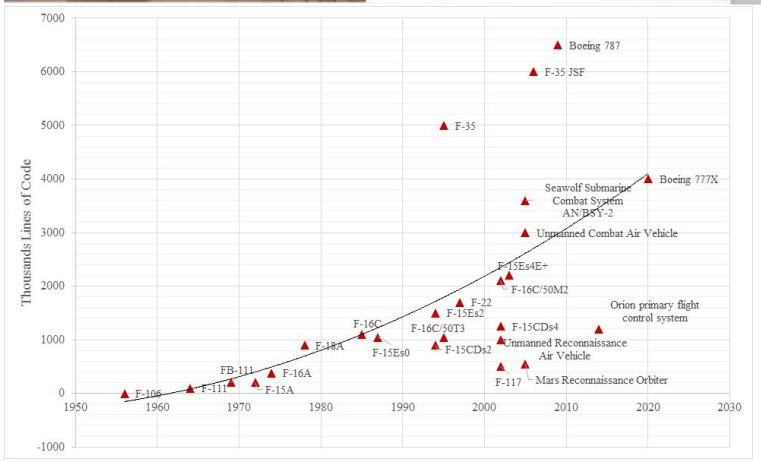
UNIVERSIDAD COMPLUTENSE MADRID

Plataforma unificada de simulación secuencial, paralela y distribuida. Aplicación a la gestión de nodos de irradiación solar.

José Luis Risco Martín
Katzalin Olcoz Herrero
Íker Prado Rujas
Luis Fernando Almendras Aruzamen
Javier Campoy

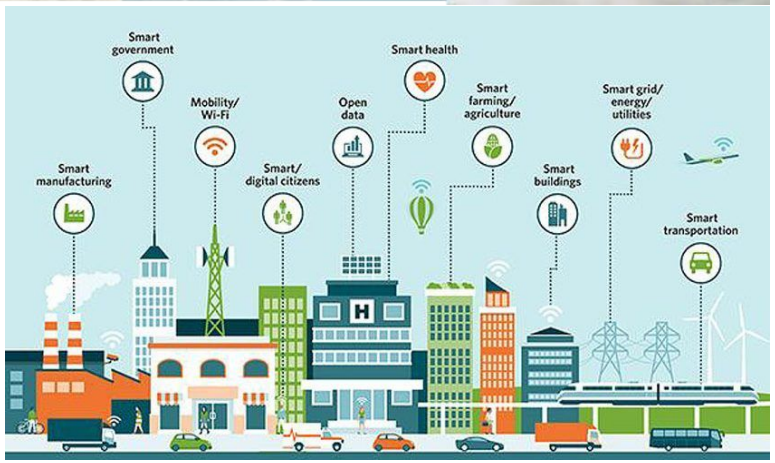
ArTeCS
CABAHLA
Comunidad de Madrid
UNIÓN EUROPEA
Fondos Estructurales
Invertimos en su futuro
UNIÓN EUROPEA
Fondo Social Europeo
El Fondo Social Europeo invierte en tu futuro

Introducción y motivación



Complejidad vertical

Modeling and simulation ***is a vital ingredient in creating the connected elements at the heart of*** ***Complex Systems.*** *It can support early evaluation and optimization of designs and ongoing verification as changes occur—to make sure the right product is developed and delivered with the required speed and quality. [**IBM**]*

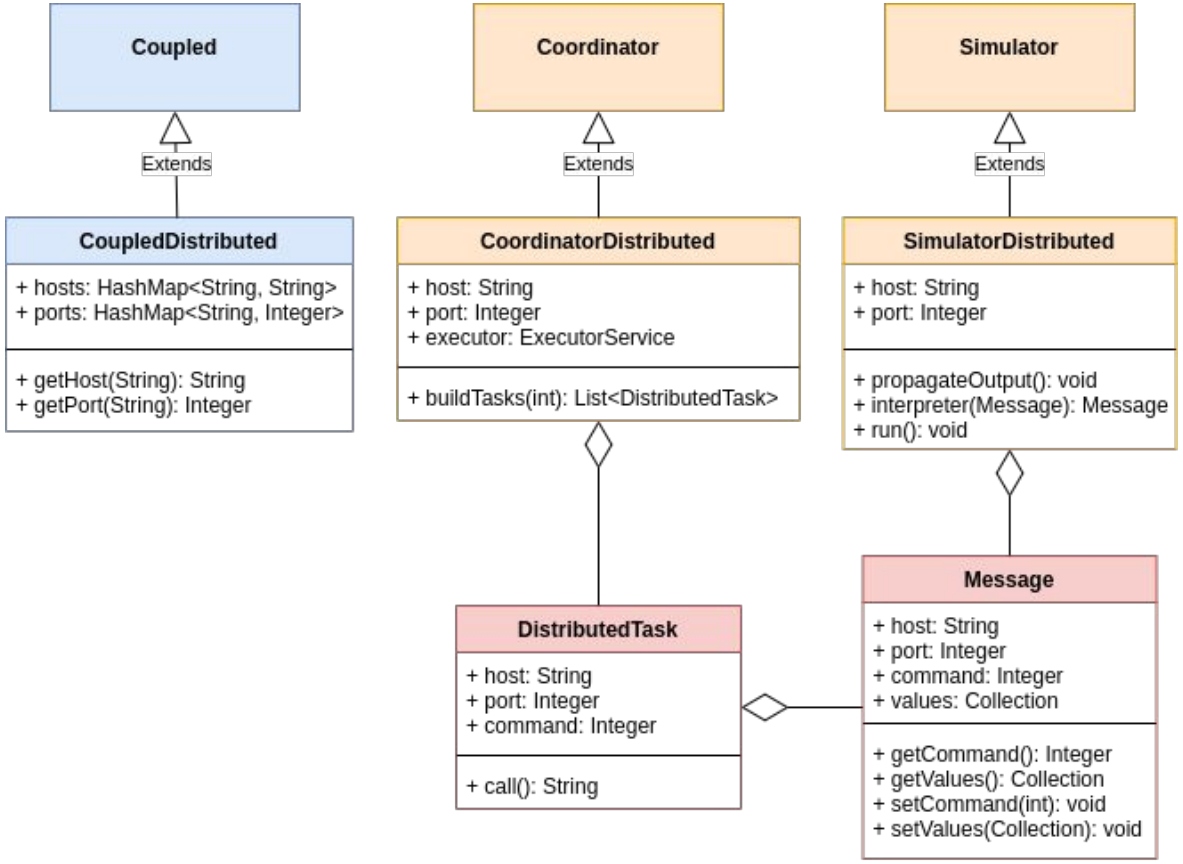


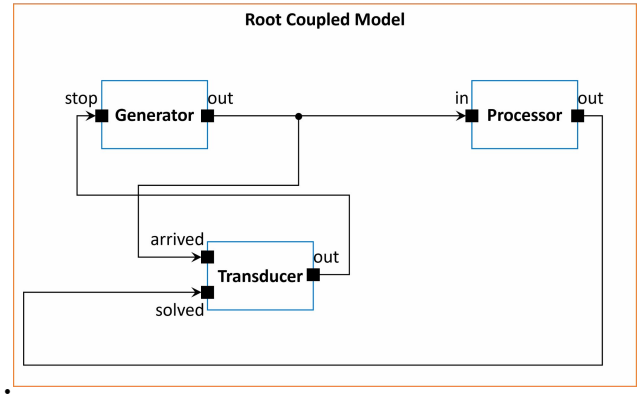
Complejidad horizontal

El uso de un ***formalismo M&S*** *permite implementar de forma precisa modelos de sistemas complejos, definiendo* ***consistencia***, ***completitud***, ***correctitud***, *y* ***verificación***, *mucho más allá que cualquier modelo ad-hoc. En nuestro caso elegimos* ***DEVS*** *porque:*

- *Permite un diseño modular y jerárquico*
- *Lidia bien con escalabilidad y reusabilidad*
- *Provee de una semántica clara para el comportamiento de los modelos*
- *Separa conceptualmente la especificación, implementación y simulación del modelo*

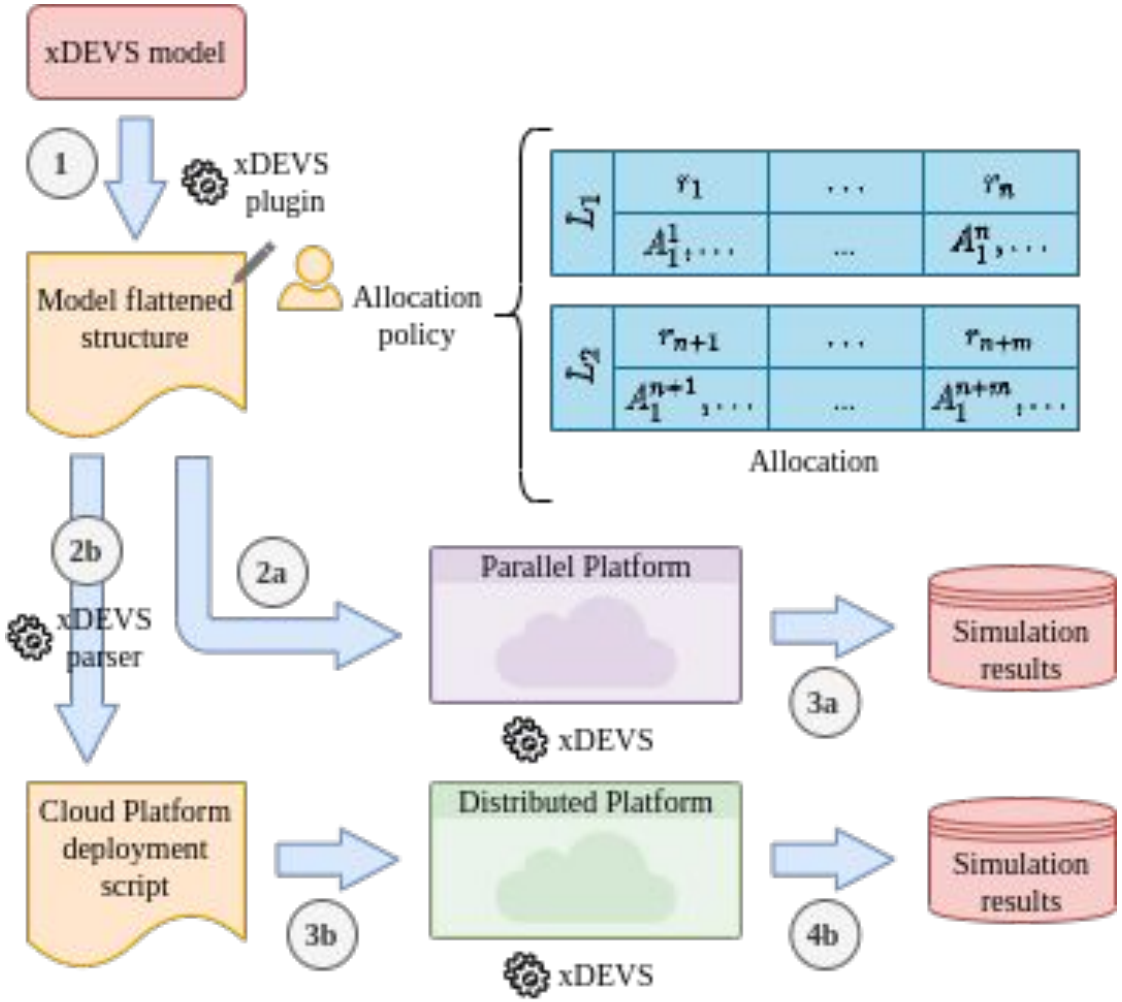
Plataforma unificada de M&S: clara y simple



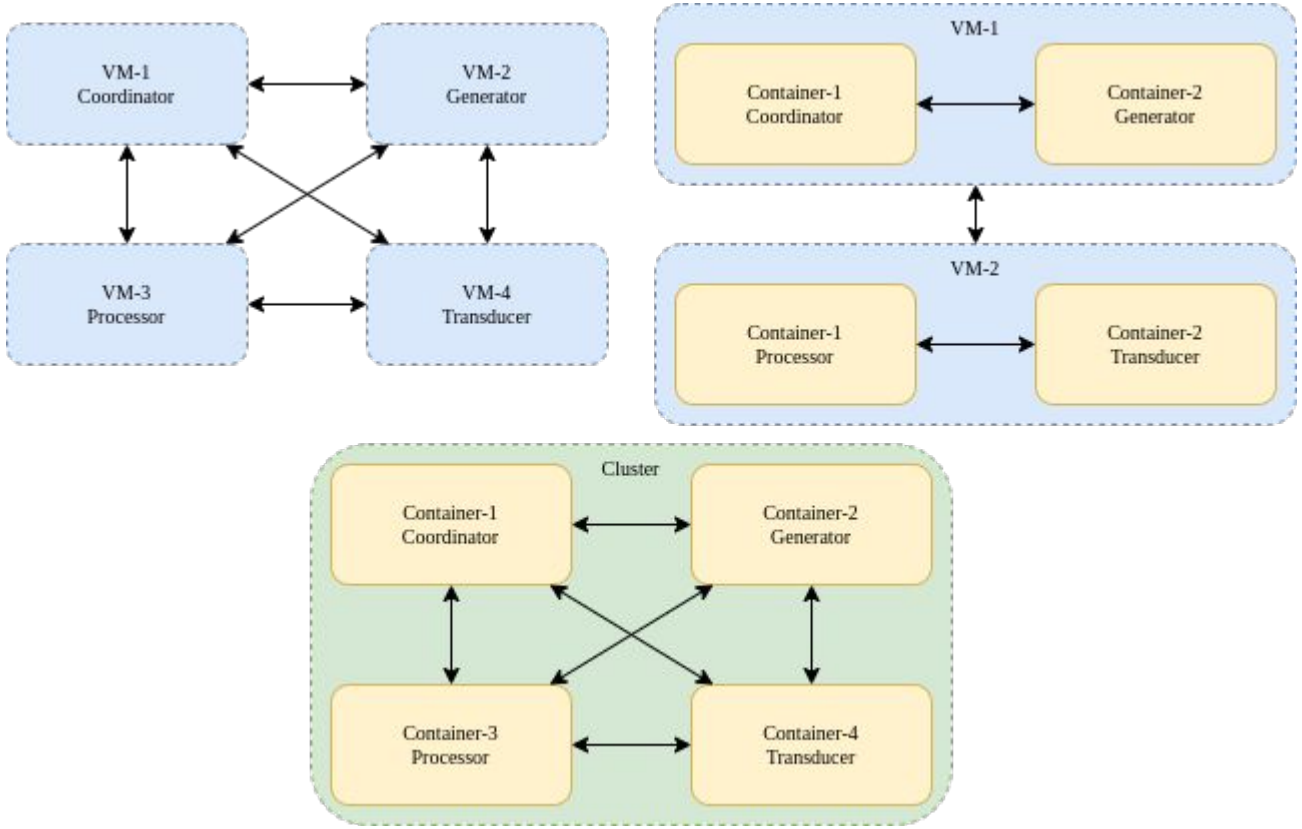



```
Coupled model = new Coupled(gpt.xml);
// Sequential execution:
Coordinator cs = new Coordinator(model);
cs.simulate(30.0);
// Parallel execution:
CoordinatorParallel cp = new CoordinatorParallel(model);
cp.simulate(30.0);
$ # Distributed execution:
$ # Simulation Entity 1, generator
$ java -cp xdevs.core.simulation.SimulatorDistributed gpt.xml generator
$ # Simulation Entity 2, processor
$ java -cp xdevs.core.simulation.SimulatorDistributed gpt.xml processor
$ # Simulation Entity 3, transducer
$ java -cp xdevs.core.simulation.SimulatorDistributed gpt.xml transducer
$ # Simulation Entity 4, run the Coordinator
$ java -cp xdevs.core.simulation.CoordinatorDistributed gpt.xml
```

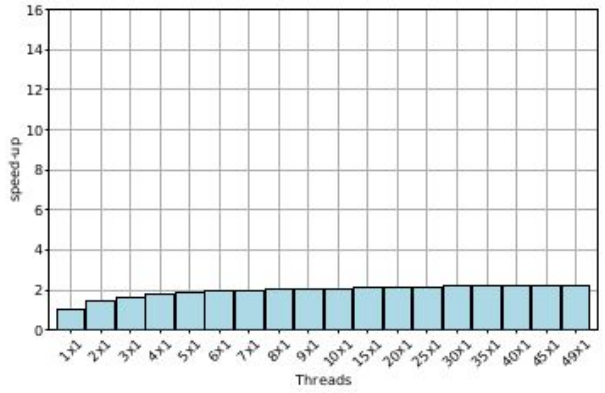
Flujo de ejecución



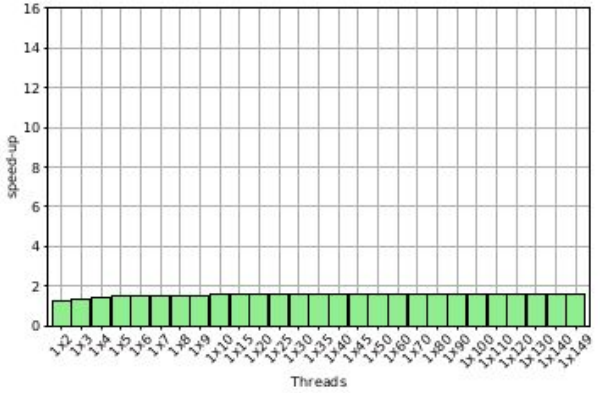
Posibilidades de despliegue distribuido



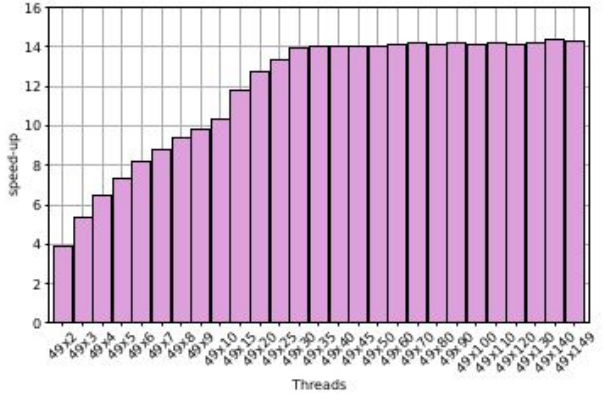
Evaluación



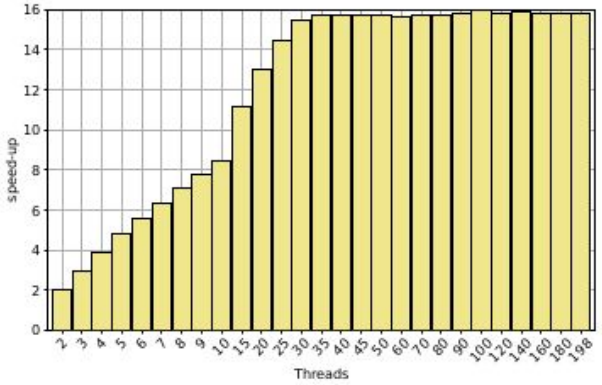
(a) Threads for slow models



(b) Threads for fast models



(c) Sub-optimal approach



(d) Balanced distribution

Figure 12: Resources (threads) distribution and speedups for the 32 vCPU parallel simulation.

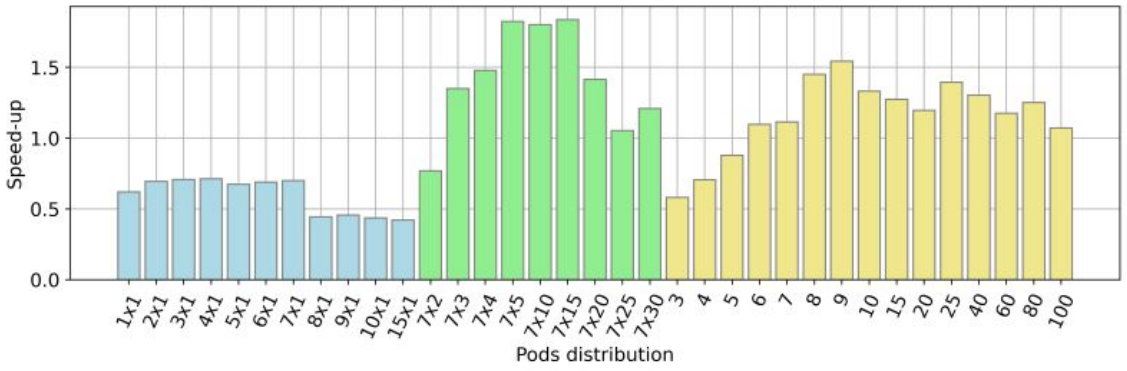


Figure 13: Distributed simulations speed-ups depending on the number of pods ($L_1 \times L_2$).

	Sub-optimal	Balanced	$/month
Parallel 4 vCPU	3.88	3.91	168.38
Parallel 32 vCPU	14.33	15.94	1413.63
Distributed 8 × 4 vCPU	1.84	1.45	1347.01

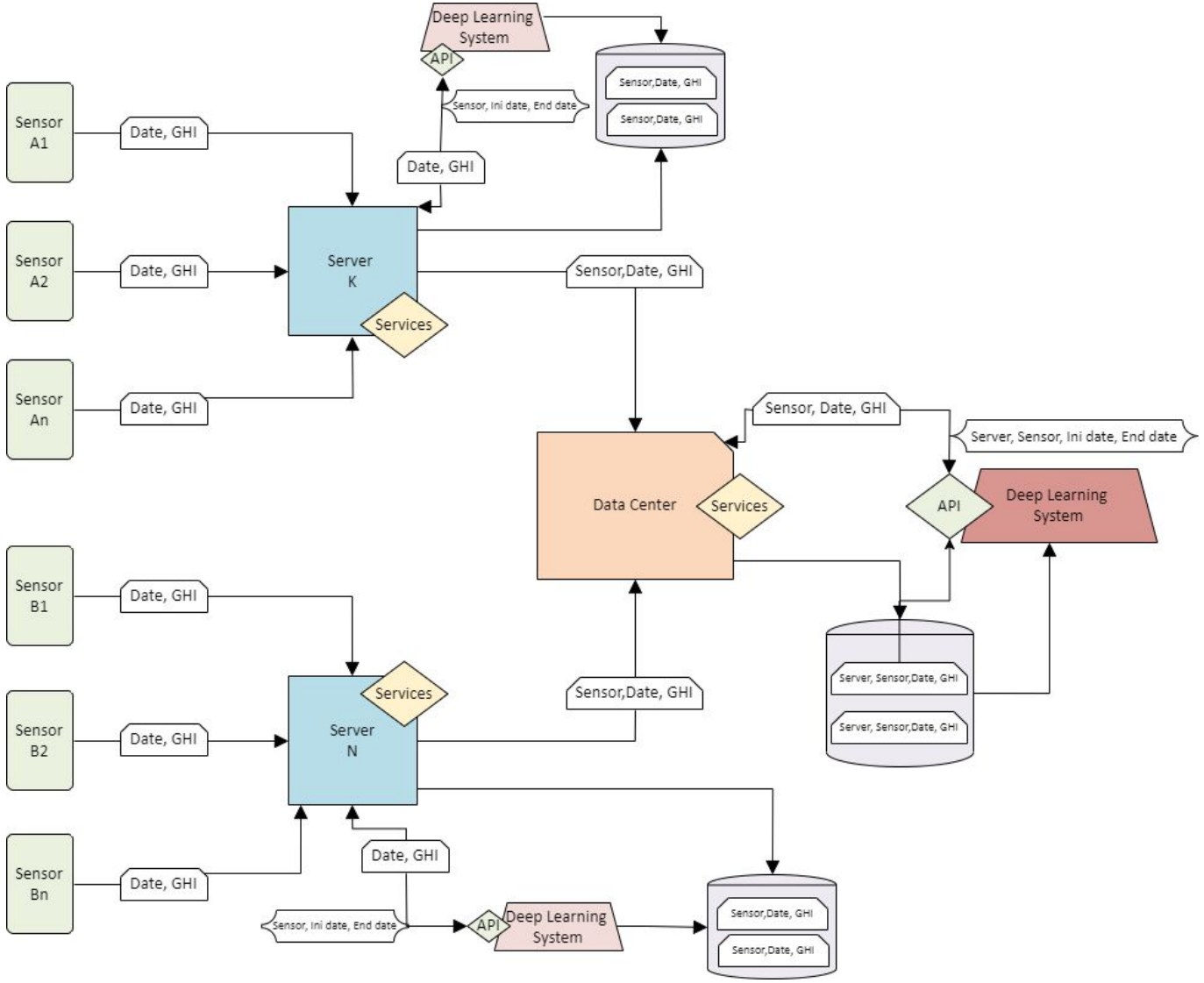
Table 2: Maximum speedup and monthly cost of the parallel and distributed simulations.

Parallel architectures are faster, cheaper, and the deployment quite straightforward …

Why do we need distributed architectures?

(Distributed) embedded systems

Caso: gestión integrada de despliegue de nodos de irradiación solar



Ruegos y preguntas