GPU Computing Architectures for Large-Scale Electromagnetic Transient Simulation

– Application in Power Converter Systems

Ning Lin
Ning.lin@powertechlabs.com
May 26, 2026

Outline

- Introduction
- Large-Scale DC Power System
- Massive Inverter-Based Resources
- Adaptive CPU-GPU Computing

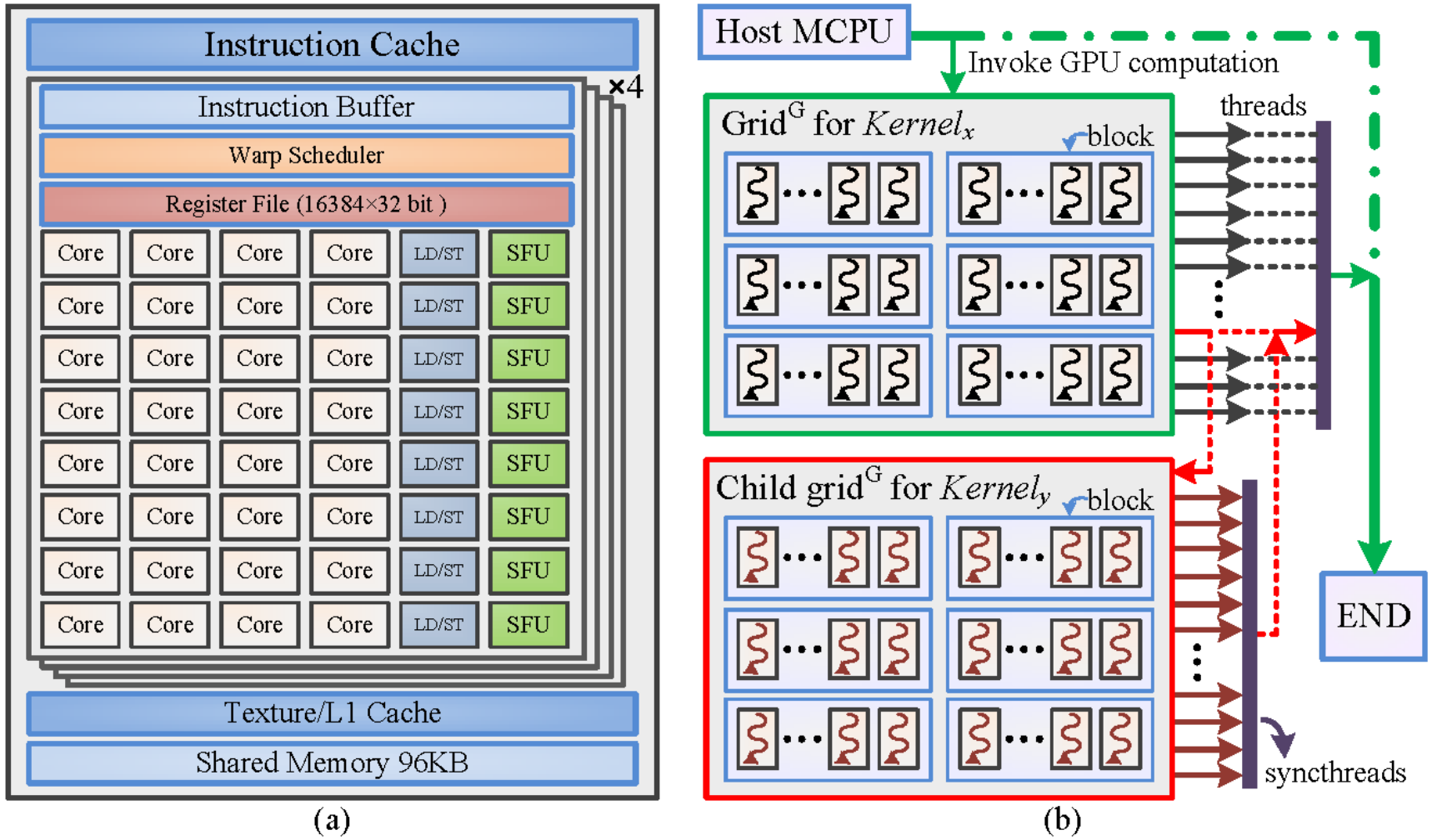
Outline

- **Introduction**
- Large-Scale DC Power System
- Massive Inverter-Based Resources
- Adaptive CPU-GPU Computing

INTRODUCTION

- Complexity of electrical power systems with increasing presence of Power Electronics
- Transition to Electromagnetic Transient Simulation for design, planning, operation, security, etc.
- Limits of EMT Simulation due to computing capability
  - Scale limit, e.g., a few hundred buses vs x100K buses in Phasor-Domain Simulation
  - Information limit, e.g., aggregation vs full-scale modeling of IBRs
  - Fidelity limit, e.g., average value model vs detailed model or device-level model
- Computing capability: sequential processing vs parallel processing
  - Parallelism of CPU (e.g., OpenMP): No. of cores, overhead
  - Massive parallelism of GPU: Single-instruction multiple threads (SMIT)

INTRODUCTION



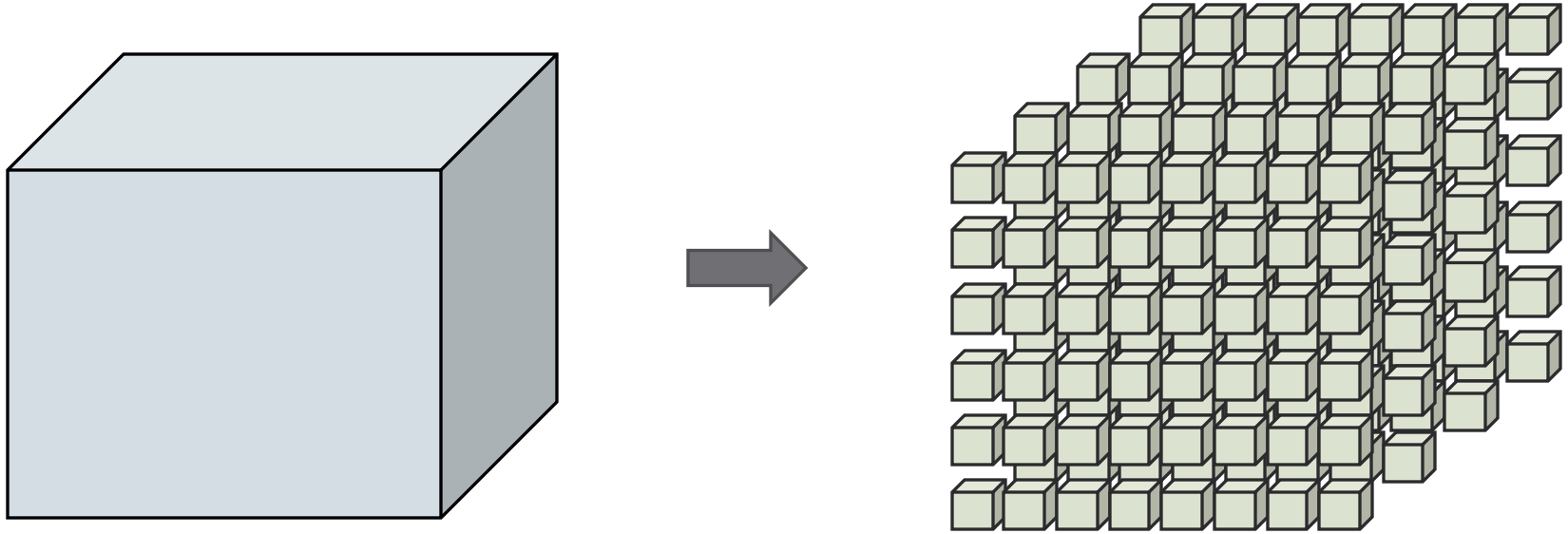
GPU streaming multiprocessor (Pascal) & Dynamic parallelism

INTRODUCTION

- GPU massively parallel processing the solution ?
- ***RECAP**: Limits of EMT Simulation due to computing capability*
  1. *Scale limit, e.g., a few hundred buses vs x100K buses in Phasor-Domain Simulation*
  2. *Information limit, e.g., aggregation vs full-scale modeling of IBRs*
  3. *Fidelity limit, e.g., average value model vs detailed model or device-level model*
- Challenges to SIMT
  1. "x100K Buses": Inhomogeneity, Matrix solver ▶▶ AC Grid, DC Grid
  2. Full-scale model: Quantity, Model complexity ▶▶ IBR Array
  3. Detailed model: Model complexity, Quantity ▶▶ Converter-dominant system
- General Principle: Enhancement of homogeneity
  – Partitioning, splitting, decoupling, decomposing

INTRODUCTION

- General Principle: Enhancement of homogeneity
  - Partitioning, splitting, decoupling, decomposing



GU=J

```
for (i=0; i<=N; i++){
...
}
```

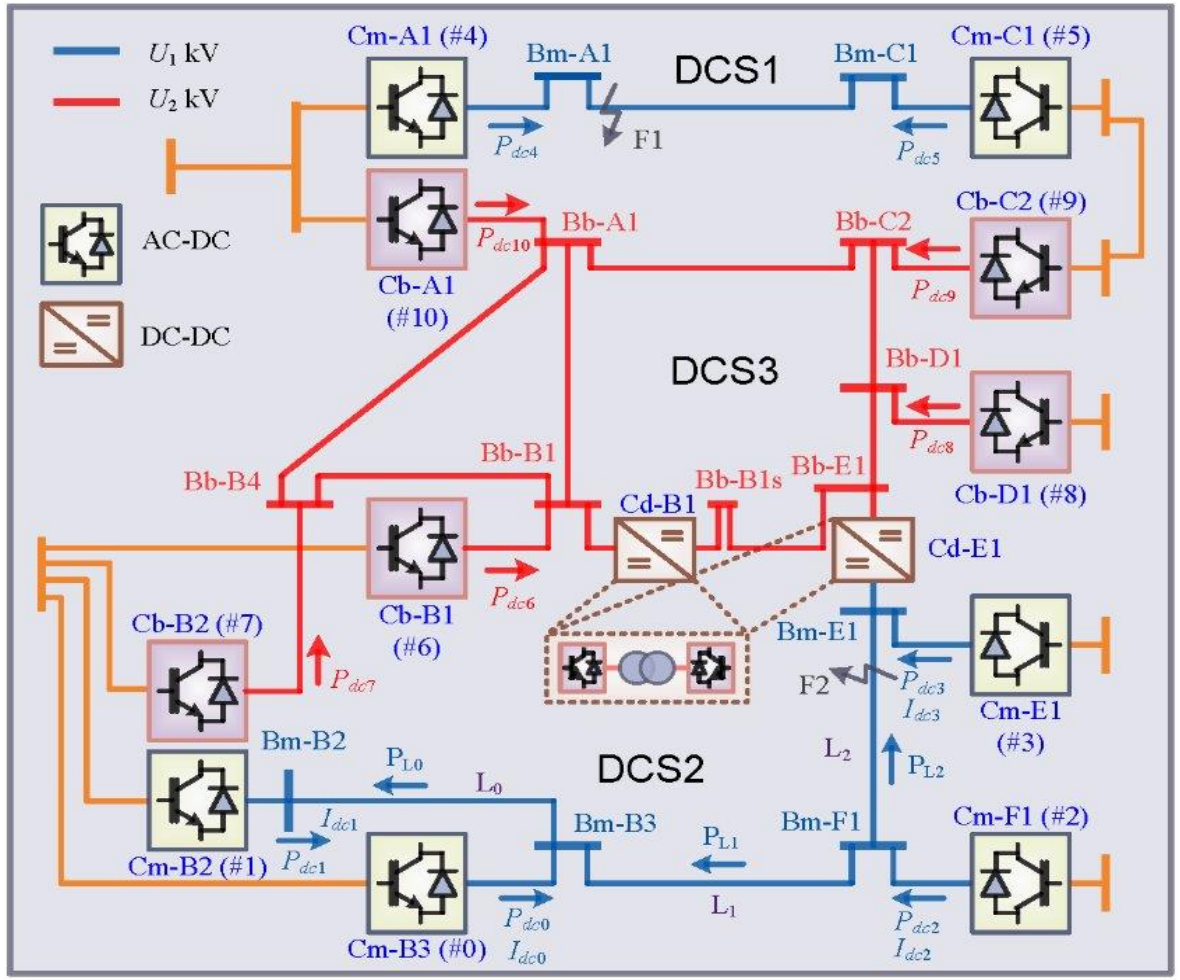
Kernel<<<M,N/M,..>>>(...)

Outline

- Introduction
- **Large-Scale DC Power System**
- Massive Inverter-Based Resources
- Adaptive CPU-GPU Computing

LARGE-SCALE DC POWER SYSTEM

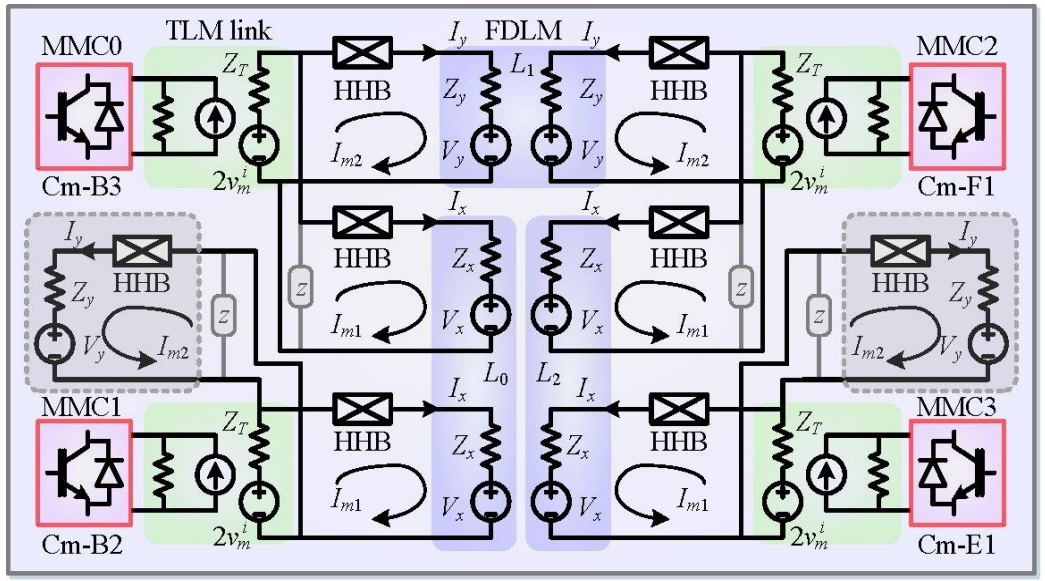
- CIGRÉ B4 DC Grid Test System
  - Eleven terminals
  - Two SSTs
- If an MMC-HVDC grid
  - x10K power switches
  - More with DCCB



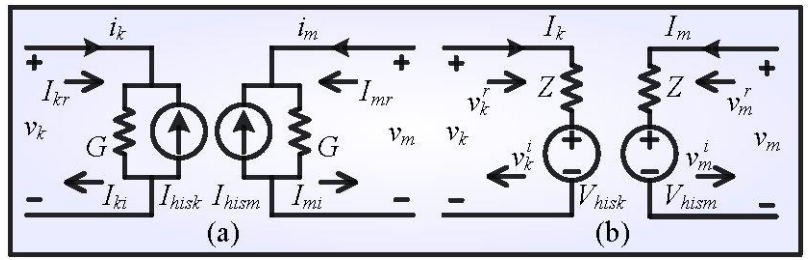
* SST: Solid-State Transformer
* MMC: Modular Multilevel Converter
* DCCB: DC Circuit Breaker

LARGE-SCALE DC POWER SYSTEM

- System decomposition: Three-level partitioning
  - DC system: Transmission line (Level-1)
  - Converter station: DC link TLM (Level-2)
  - Converter: TLM, V-I coupling (Level-3)



- DC system: Transmission line (Level-1)

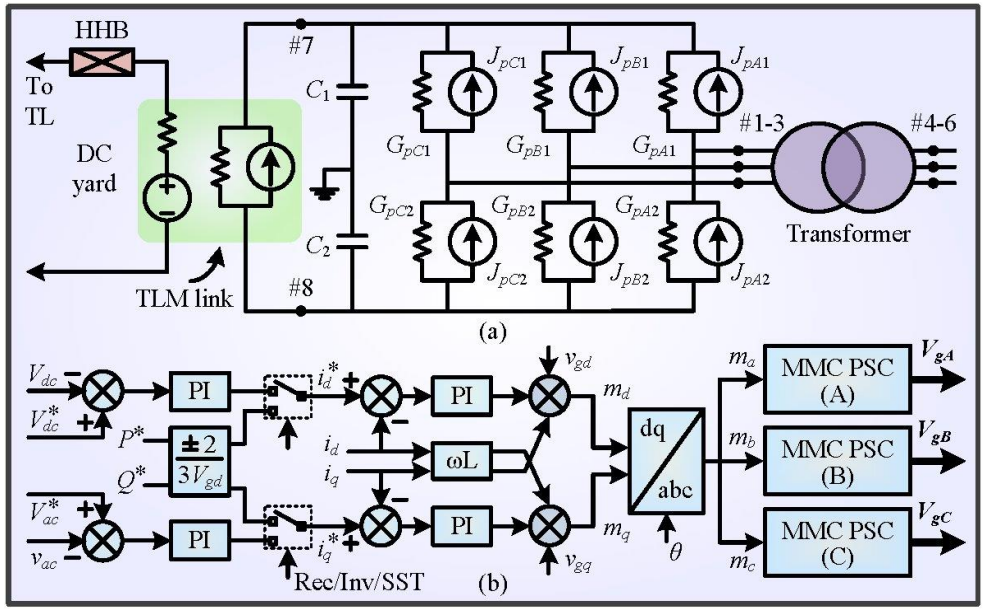


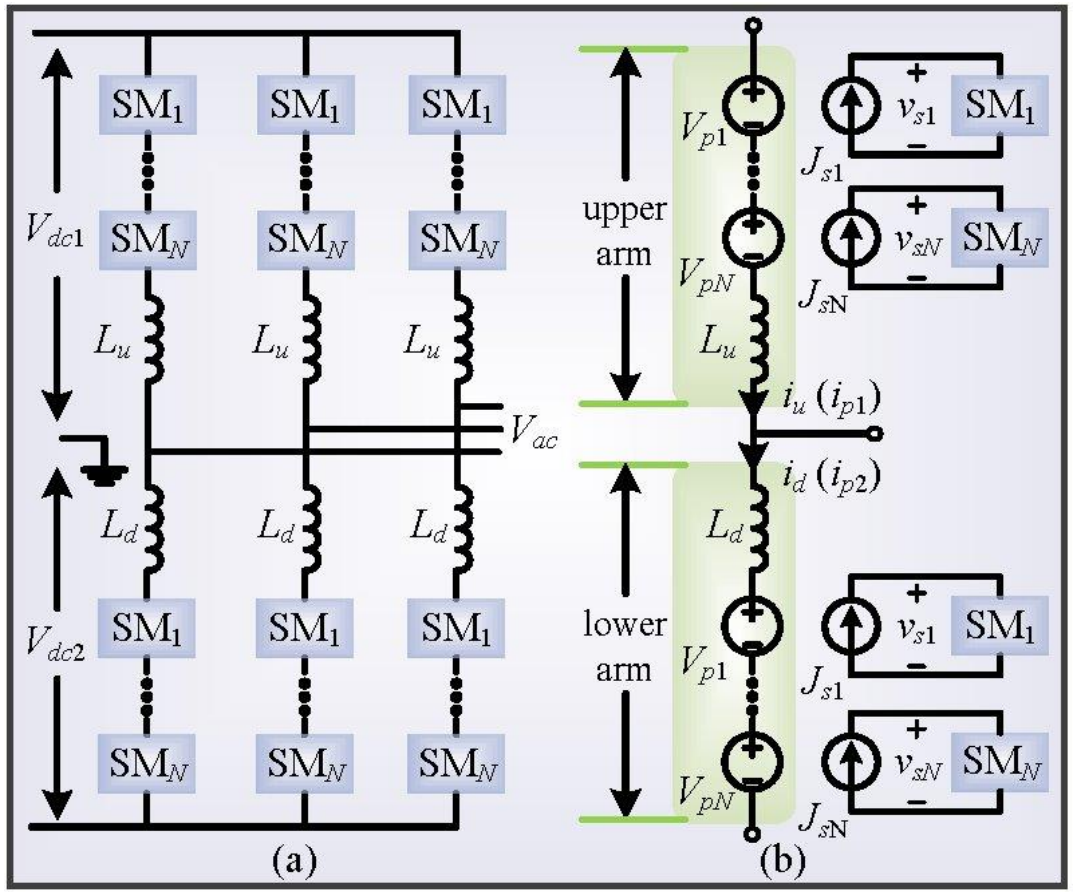
- Converter station: DC link TLM (Level-2)

* HHB: Hybrid HVDC Breaker
* FDLM: Frequency-Dependent Line Model
* TLM: Transmission Line Modeling

LARGE-SCALE DC POWER SYSTEM

- System decomposition: Three-level partitioning
  - Converter: TLM, V-I coupling (Level-3)
- Three-phase MMC & computational model

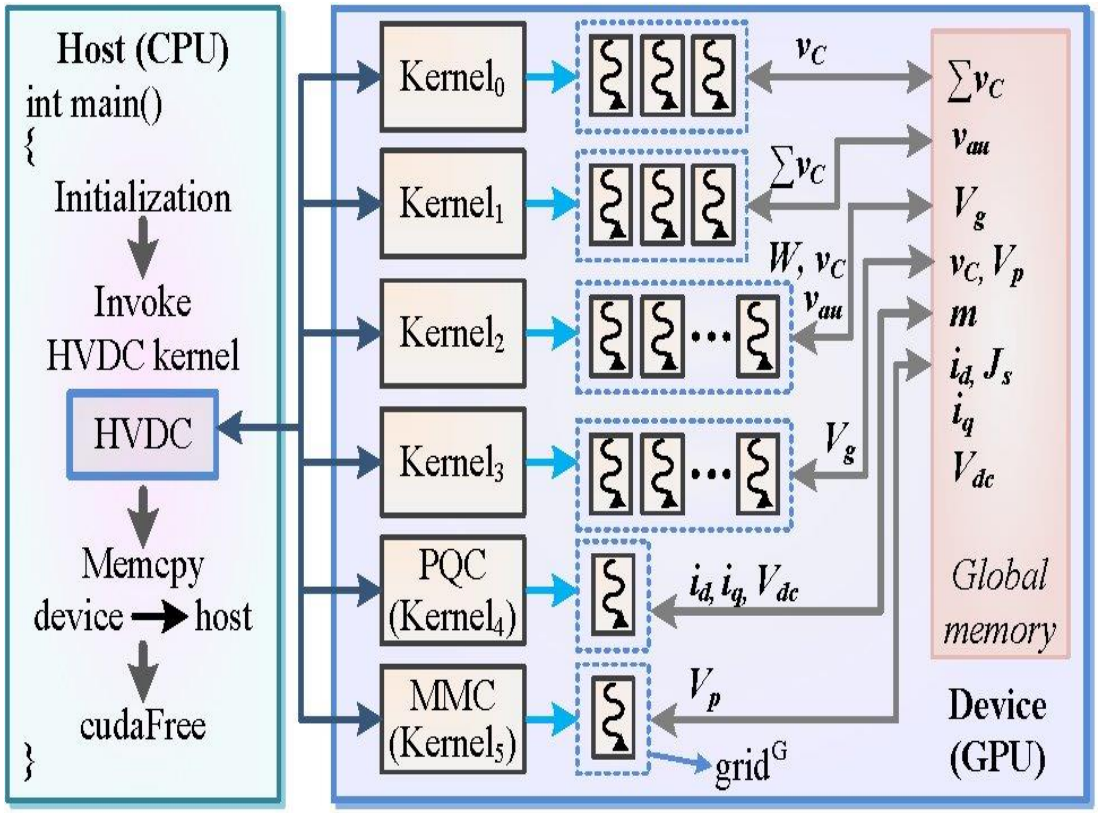




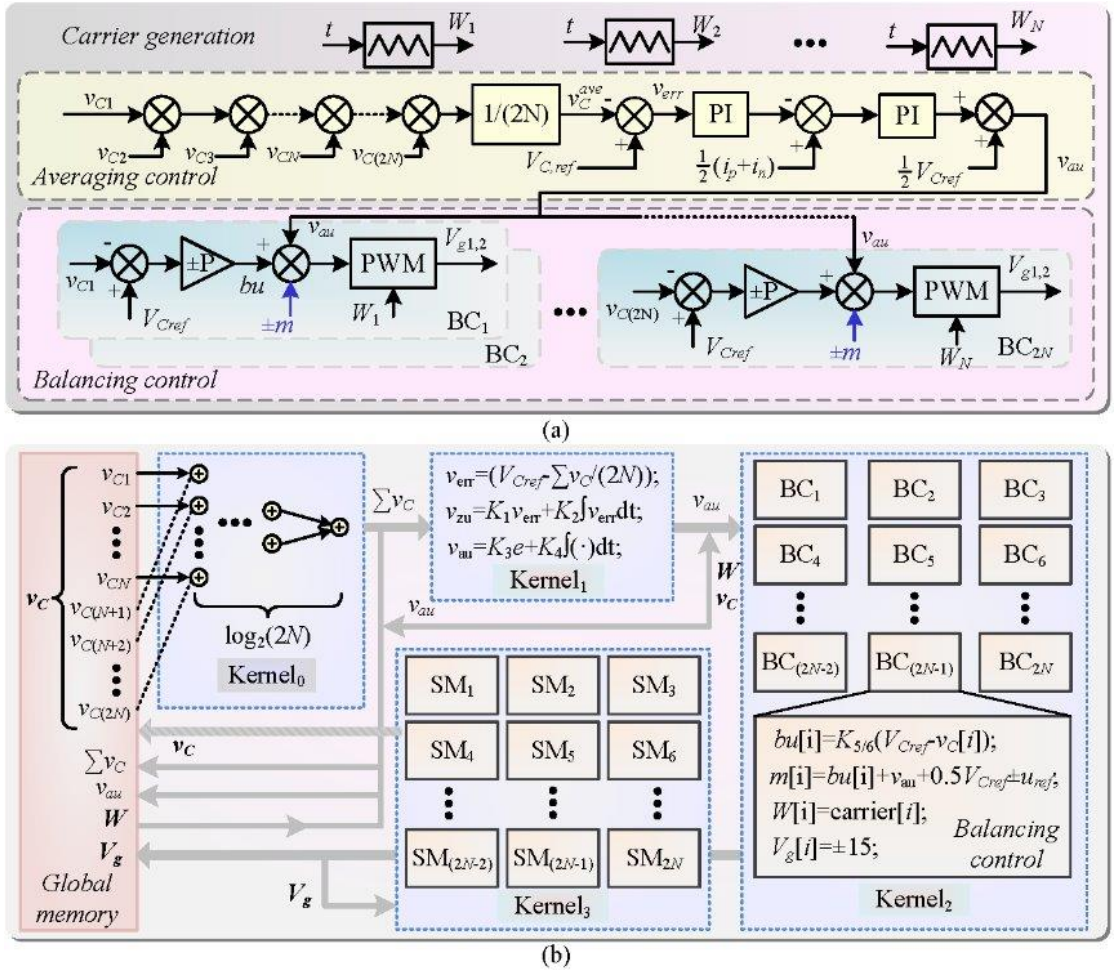
* SM: (MMC) Submodule

LARGE-SCALE DC POWER SYSTEM

- Hierarchical dynamic parallelism



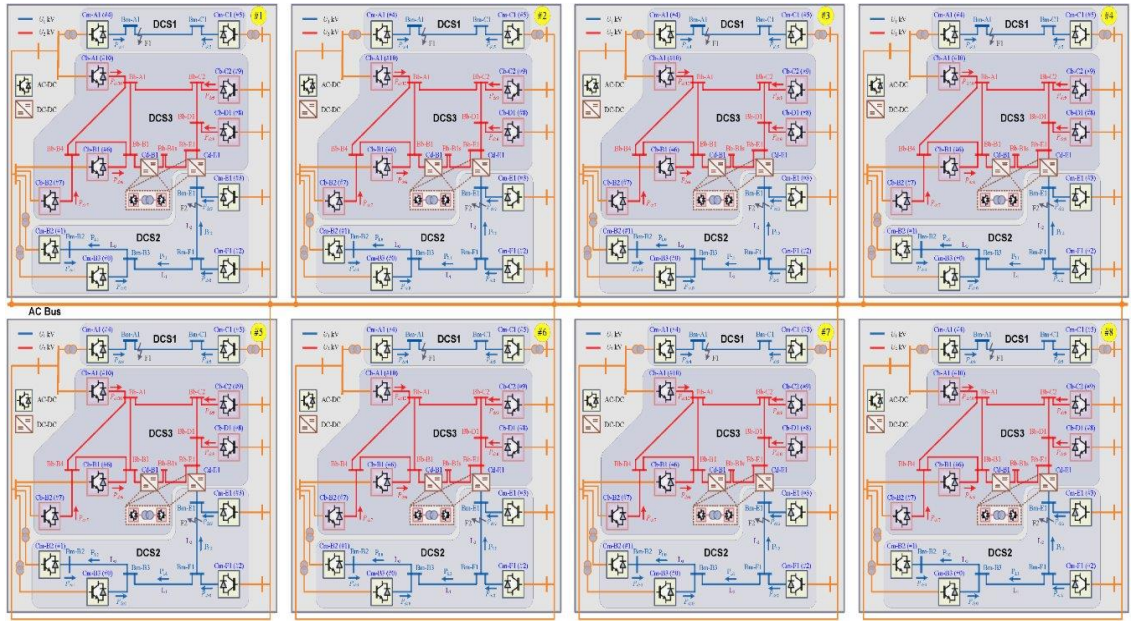
- HVDC station kernel architecture



- MMC phase-shift control (CPU vs. GPU)

LARGE-SCALE DC POWER SYSTEM

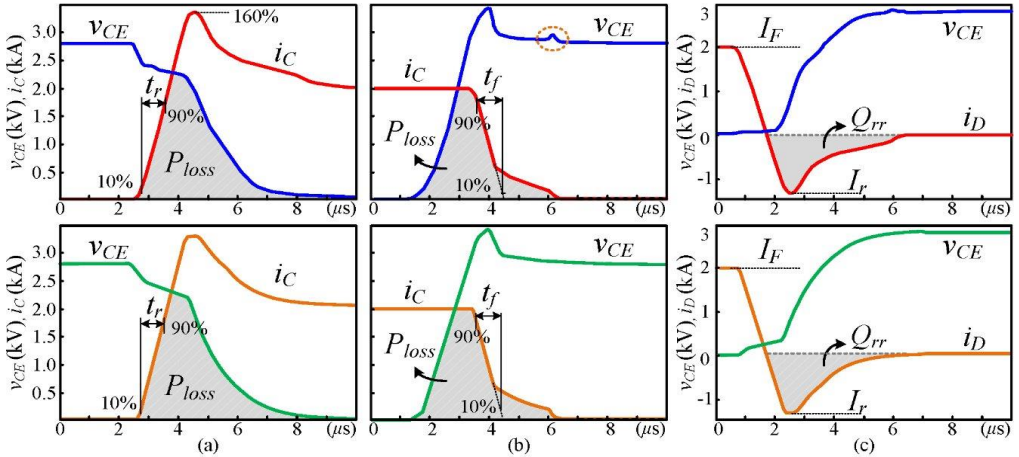
- Up to 8 x CIGRÉ B4 DC Grid



- 20 sec simulation -> 50 min ~ 1hr10min
- Power switch device-level results
- Multi-rate: 50 ns + 1 *us*

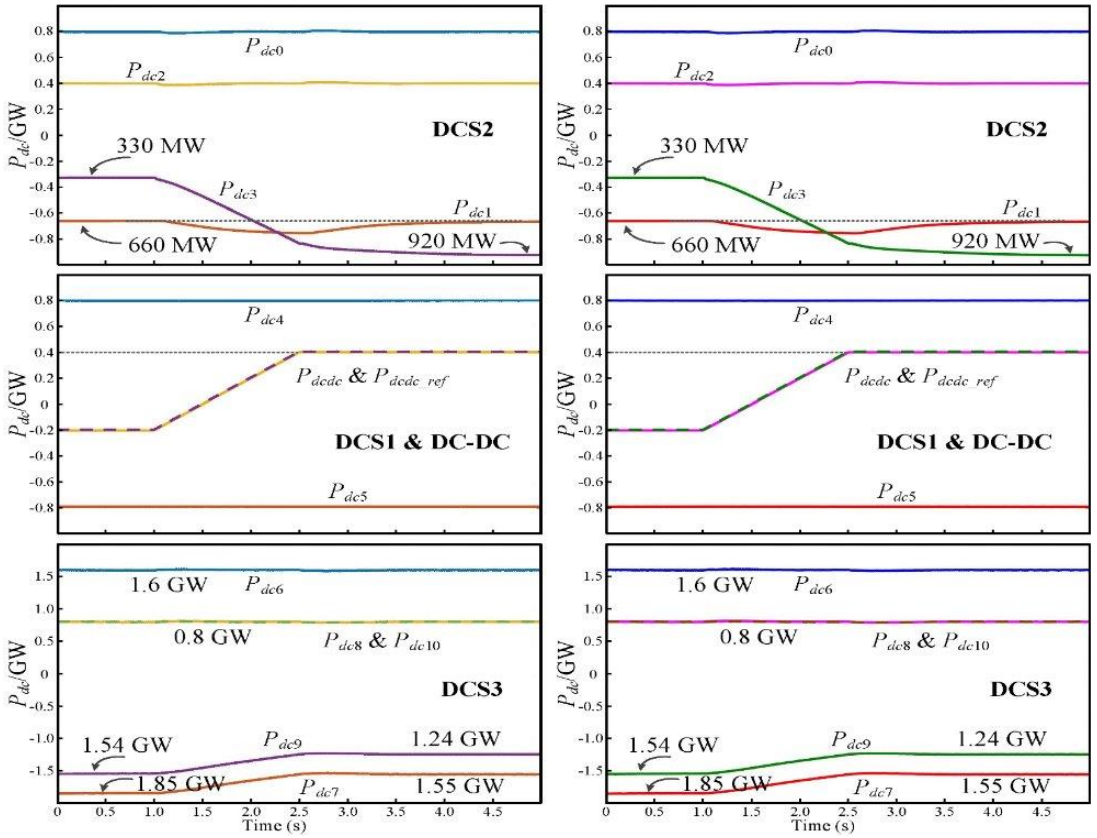
TABLE 1. **CPU and GPU execution times of the Greater CIGRÉ DC system (Fig. 14) for 1s simulation.**

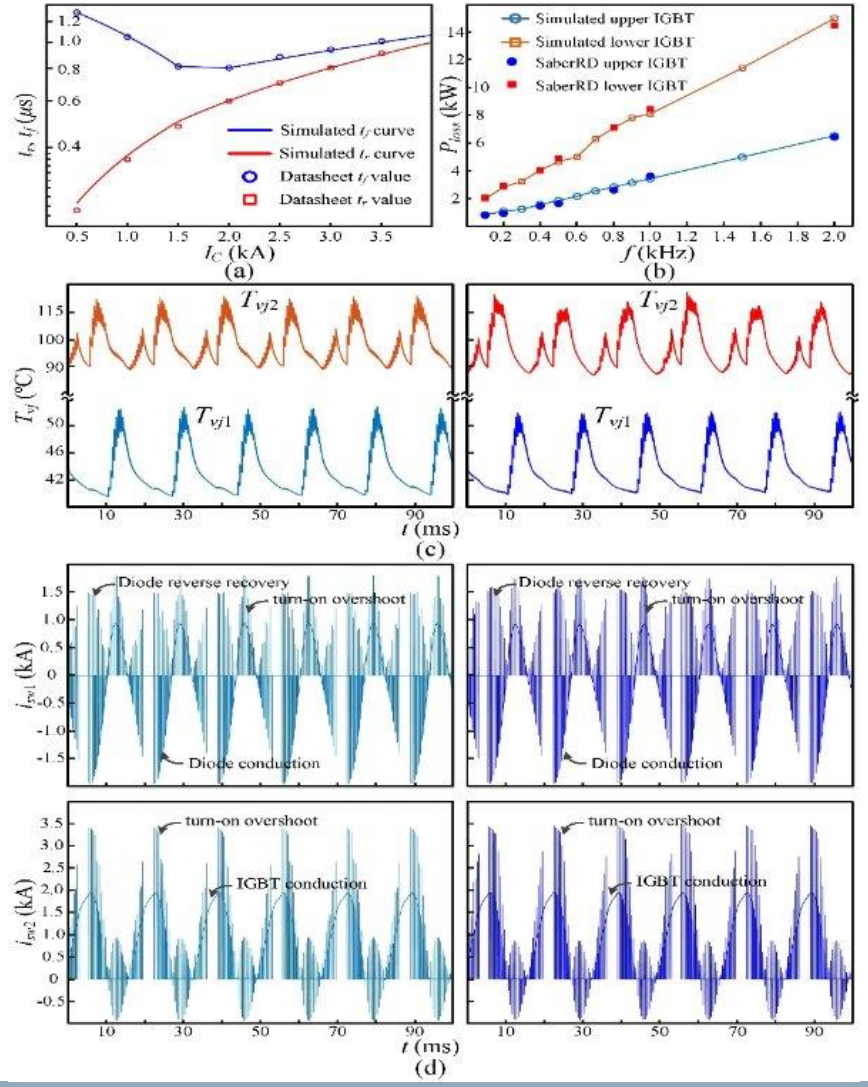
CIGRÉ NO.	Execution time (s)			Speedup		
	CPU	MCPU	GPU	$\frac{\text{CPU}}{\text{MCPU}}$	$\frac{\text{CPU}}{\text{GPU}}$	$\frac{\text{MCPU}}{\text{GPU}}$
2	14549	1995	162.1	7.3	89.8	12.3
3	22264	2810	179.9	7.9	123.8	15.6
4	30249	3103	195.6	9.7	154.6	15.9
5	36963	4067	200.9	9.1	184.0	20.2
6	44391	4284	208.4	10.4	213.0	20.6
7	52792	4868	210.0	10.8	251.4	23.2
8	60121	5538	221.9	10.9	270.9	25.0



LARGE-SCALE DC POWER SYSTEM

- System-level (L)+device-level (R) results
- Benchmark against PSCAD + SaberRD



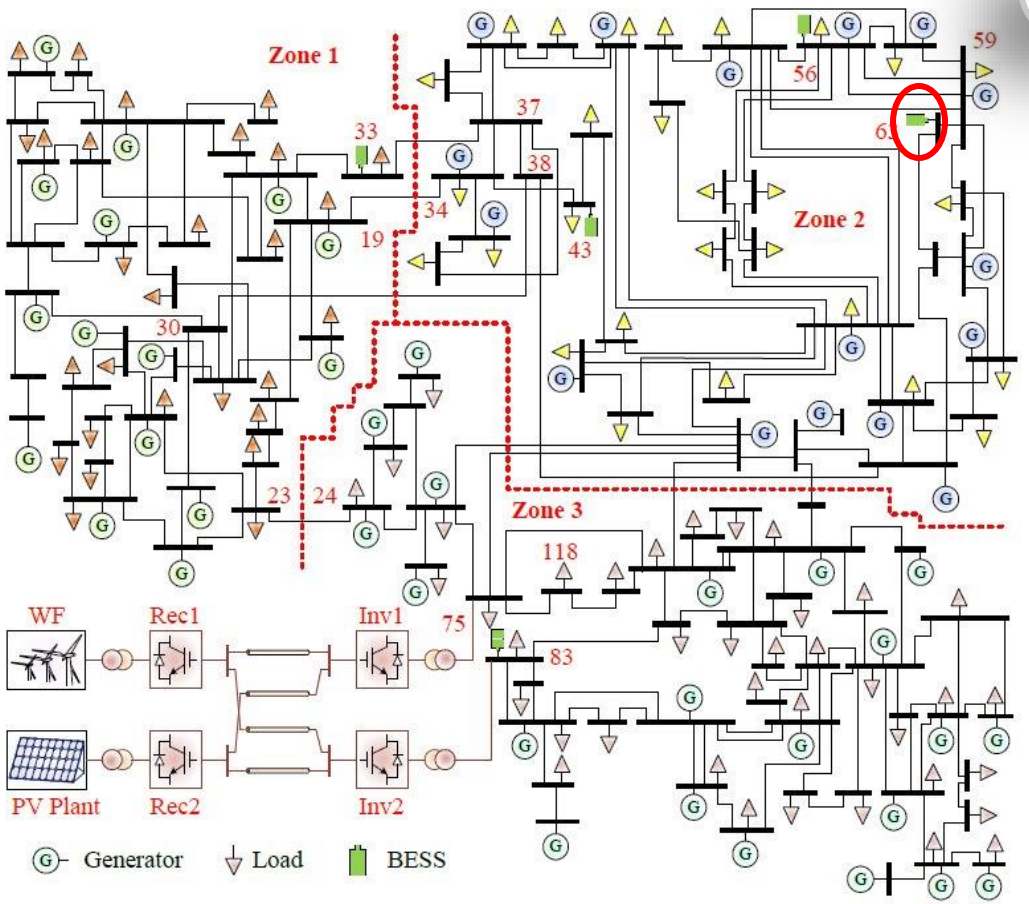


Outline

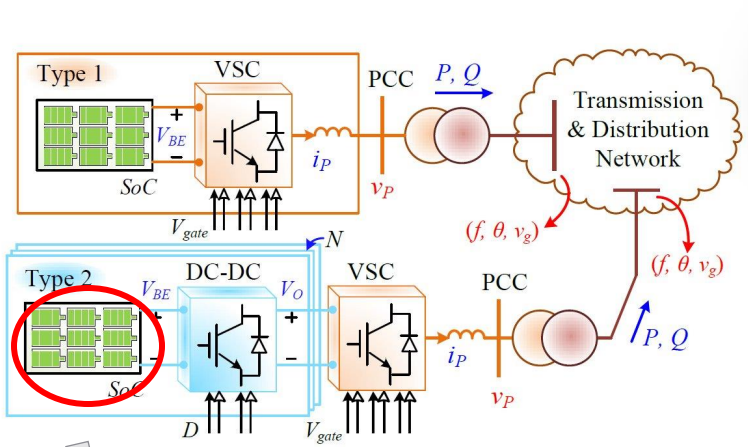
- Introduction
- Large-Scale DC Power System
- **Massive Inverter-Based Resources**
- Adaptive CPU-GPU Computing

MASSIVE INVERTER-BASED RESOURCE

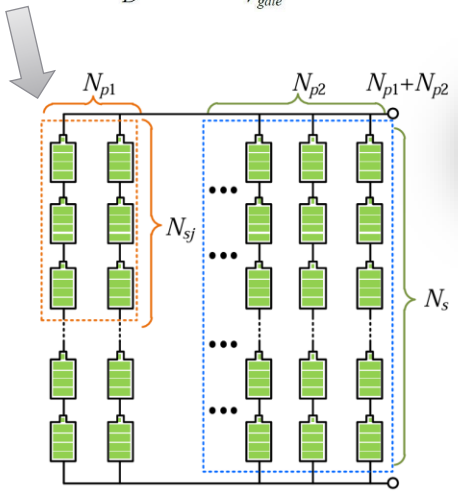
- Modified IEEE 118-bus system with BESS



Fidelity level 1



Fidelity level 2



Fidelity level 3

* BESS: Battery Energy Storage System

MASSIVE INVERTER-BASED RESOURCE

- Multi-fidelity modeling for system simulation

Fidelity level 1 – available in PDS/EMT tools

PDS: REGC_*, REEC_*, REPC_*, WTG*

EMT: Average value models, (conditionally) Detailed models

Fidelity level 2 – preferred and pursued

Detailed models with converter circuit and control structure

Challenges: Node number, Switching frequency (e.g., **DC-DC**)

Hybrid modeling?

Fidelity level 3 – challenging

Element level modeling for ultra-high fidelity

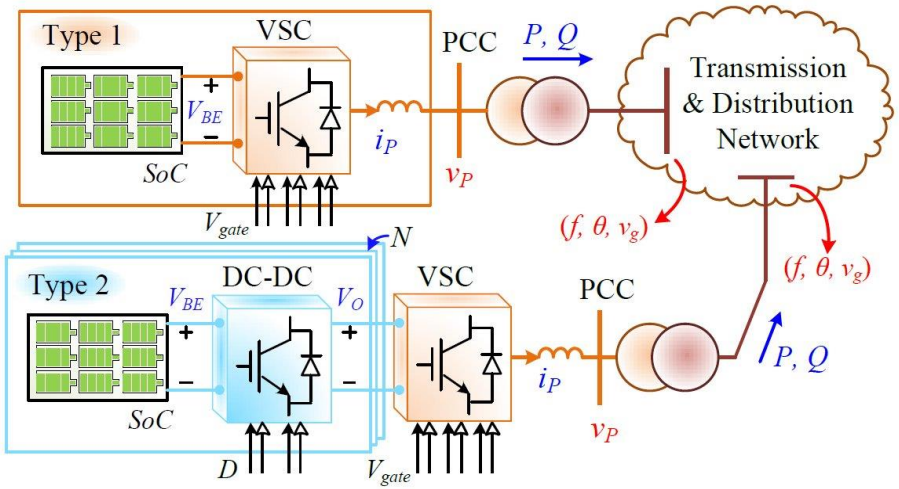
GPU SIMT

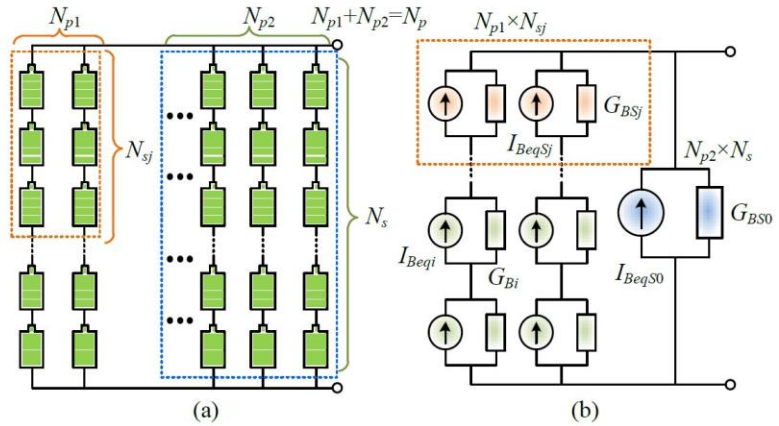
Challenge: Node number, Switching frequency, **Element number**

* PDS: Phasor-Domain Simulation

MASSIVE INVERTER-BASED RESOURCE

- Modified IEEE 118-bus system modeling



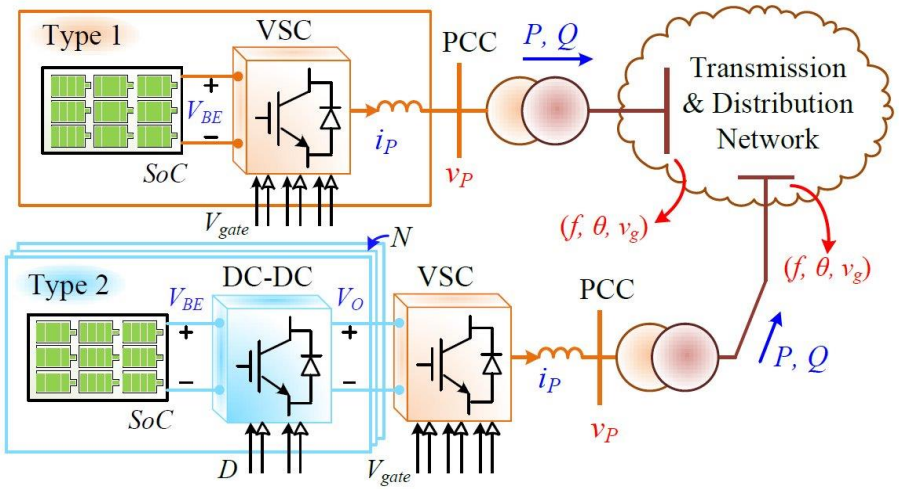


- AC grid: Regular modeling (EMT, PDS)
- VSC: Detailed modeling (power switch)
- DC-DC (SST, DAB): State-space modeling
- Batteries: Flexible-fidelity modeling

* SST: Solid-State Transformer
* DAB: Dual Active Bridge

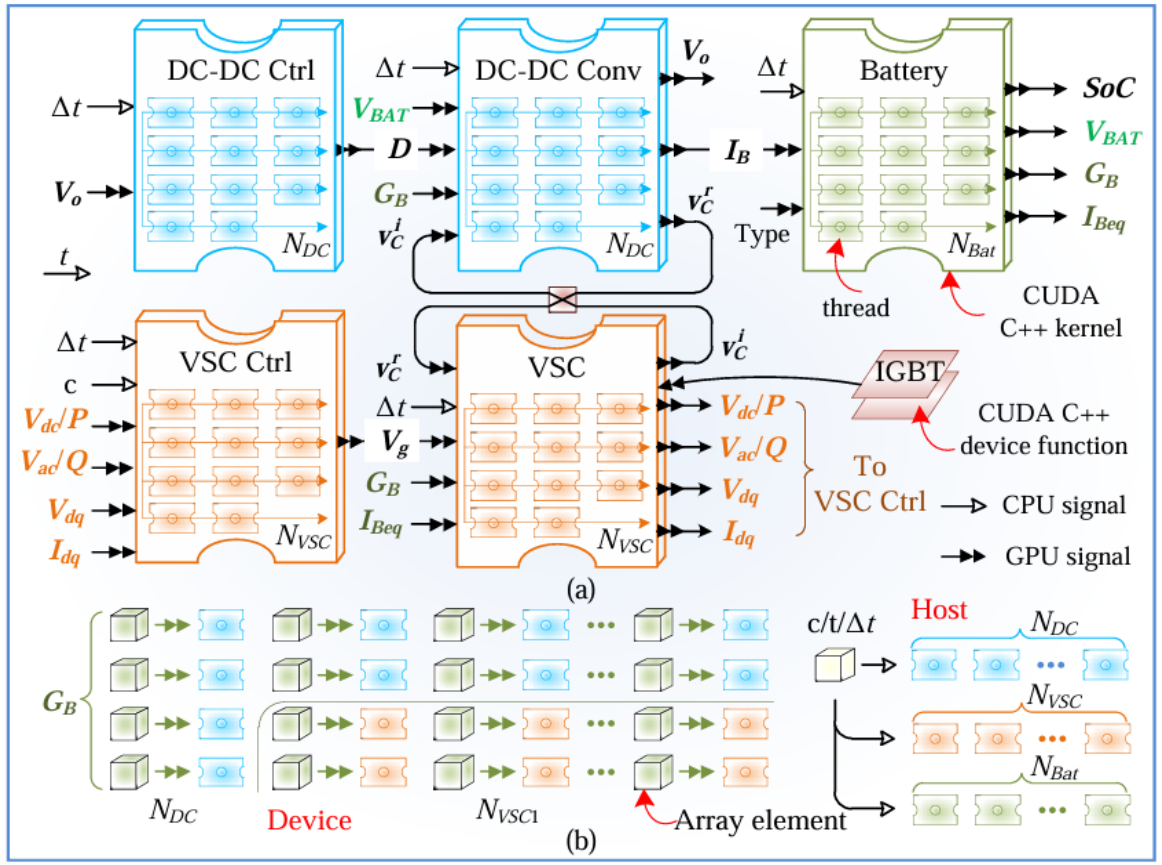
MASSIVE INVERTER-BASED RESOURCE

- Mixture of Type 1 and Type 2?



- BESS kernels
  - Common: VSC Ctrl, VSC, Battery
  - Type 2: DC-DC Ctrl, DC-DC Conv

Modular architecture

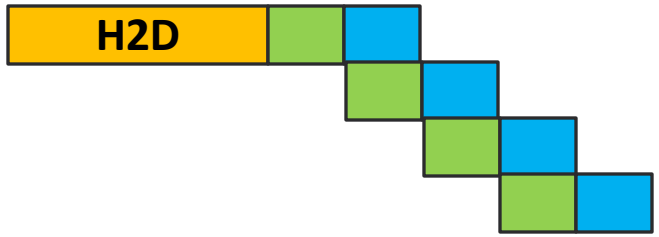


MASSIVE INVERTER-BASED RESOURCE

- Kernel implementation

Kernel<<<block,thread,M,Stream>>>(…)

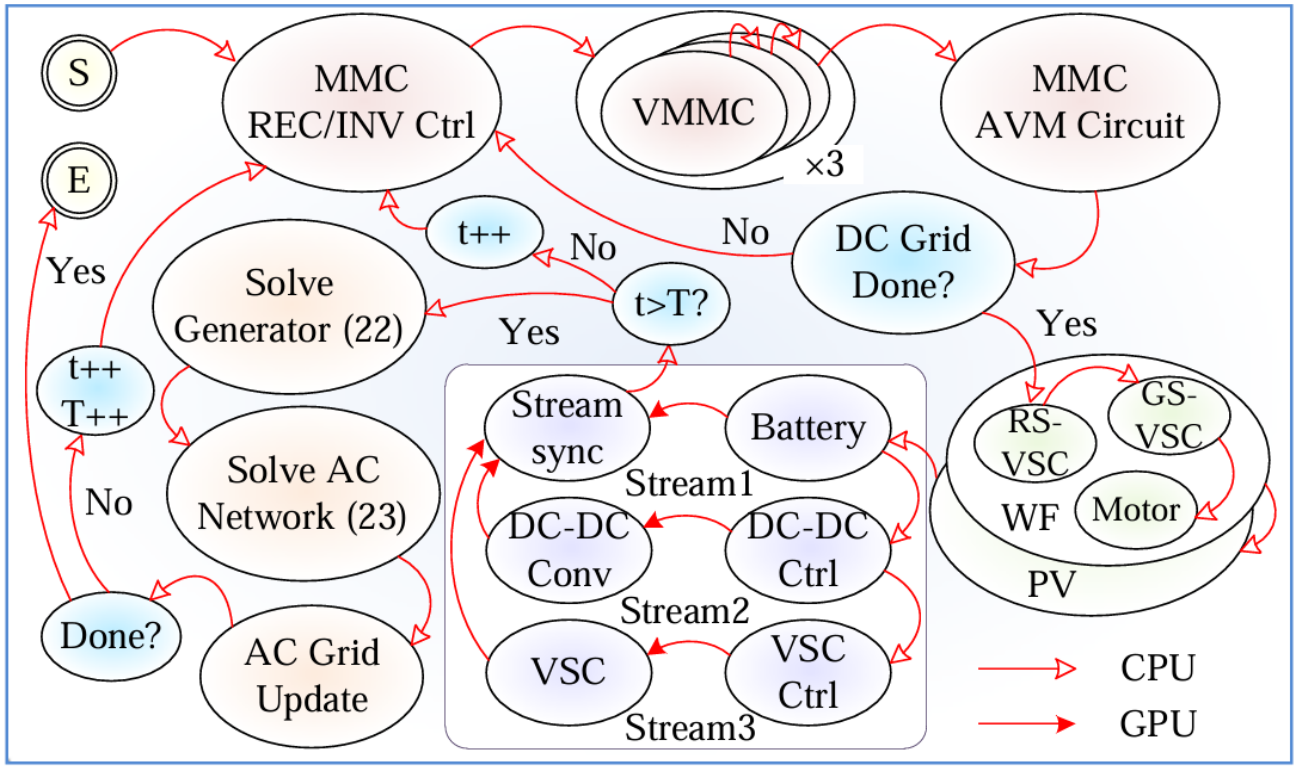
Two-way concurrency (more ways…)



H2D & D2H: cudaMemcpyAsync(d_data, h_data, size, cudaMemcpyHostToDevice, stream)

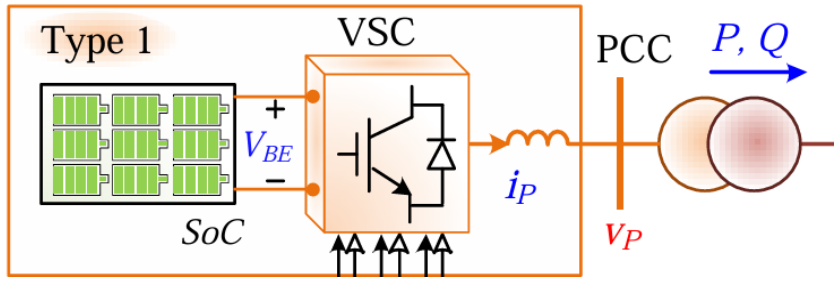
- Skip Type 2 DC-DC kernels if necessary

Multi-rate multi-stream CPU-GPU implementation



MASSIVE INVERTER-BASED RESOURCE

- Speedup (T=20 sec, dt=20μs)



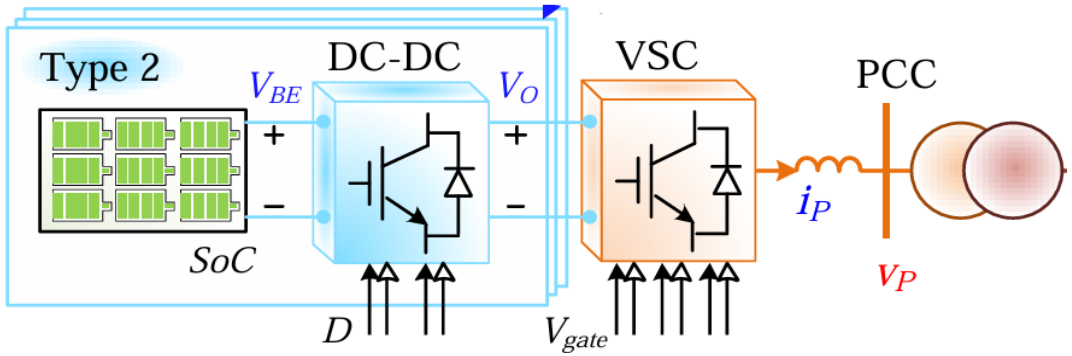


TABLE I
TYPE 1 BESS SIMULATION SPEED COMPARISON

BESS	Nonlinear IGBT (t: s)				TSSM (t: s)			
Scale	t_{CPU}	t_{GPU}	t_{MS-GPU}	SP	t_{CPU}	t_{GPU}	t_{MS-GPU}	SP
1	0.6	25	22	0.03	0.32	20	17	0.02
10	4.6	26	22	0.2	3.0	21	17	0.2
100	42	27	23	1.8	27	22	18	1.5
1K	422	27	23	18	284	22	18	16
10K	4230	34	30	141	2880	27	22	131
20K	8550	54	45	190	5600	39	33	170
30K	12800	69	60	213	9980	51	41	243
40K	17100	90	79	216	11000	63	52	212
50K	21200	108	93	228	14100	74	60	235

TABLE II
TYPE 2 BESS SIMULATION SPEED COMPARISON

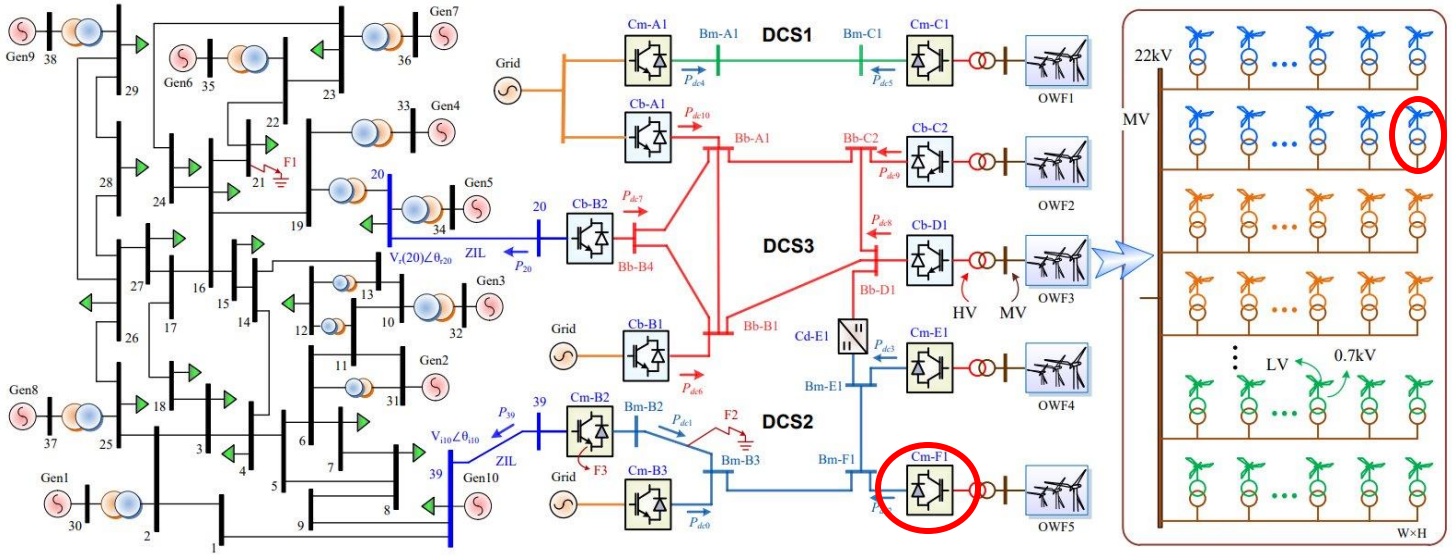
BESS	Nonlinear IGBT (t: s)				TSSM (t: s)			
VSC:DC	t_{CPU}	t_{GPU}	t_{MS-GPU}	SP	t_{CPU}	t_{GPU}	t_{MS-GPU}	SP
1:20	1.4	29	26	0.05	1.2	24	22	0.05
10:200	11.8	30	27	0.44	10.7	26	23	0.47
100:2K	117	31	28	4.2	103	27	24	4.3
1K:20K	1100	34	29	38	1000	29	25	40
5K:100K	5400	68	64	84	4800	62	59	81
10K:200K	18000	108	101	178	16300	103	100	163

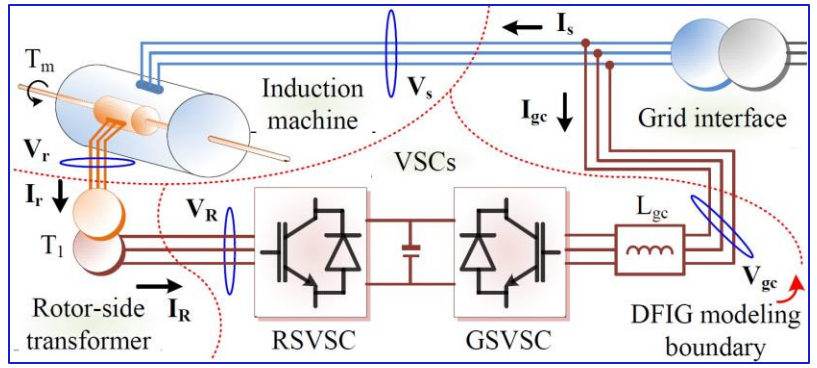
Outline

- Introduction
- Large-Scale DC Power System
- Massive Inverter-Based Resources
- **Adaptive CPU-GPU Computing**

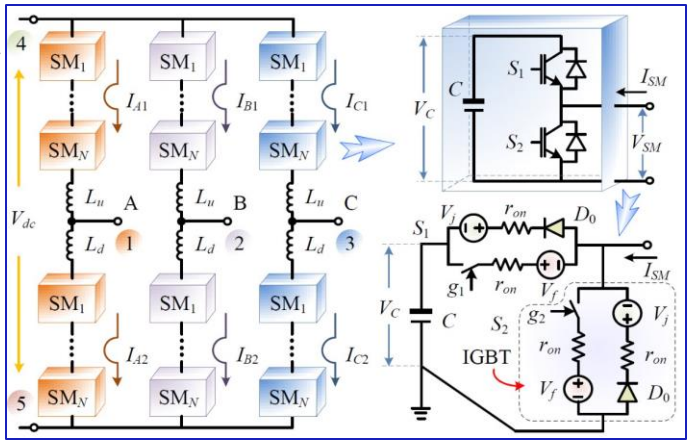
ADAPTIVE CPU-GPU COMPUTING

- AC/DC grids with large-scale wind farms





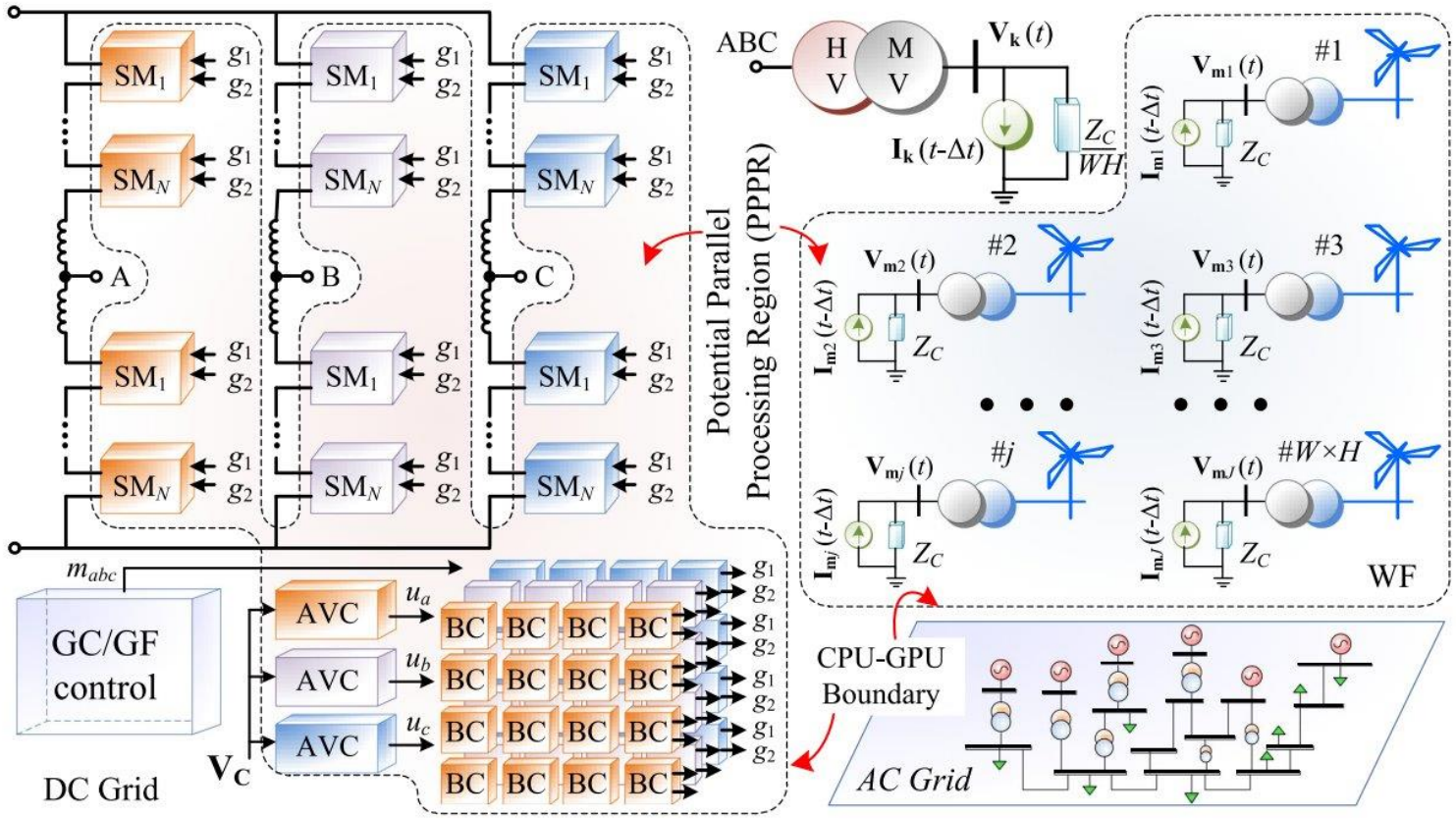
- Detailed MMC & DFIG model
  - MMC submodule
  - Power semiconductor switch



- Twofold hybrid simulation:
  - CPU + GPU
  - PDS (AC grid) +EMT (DC grid + WFs)

ADAPTIVE CPU-GPU COMPUTING

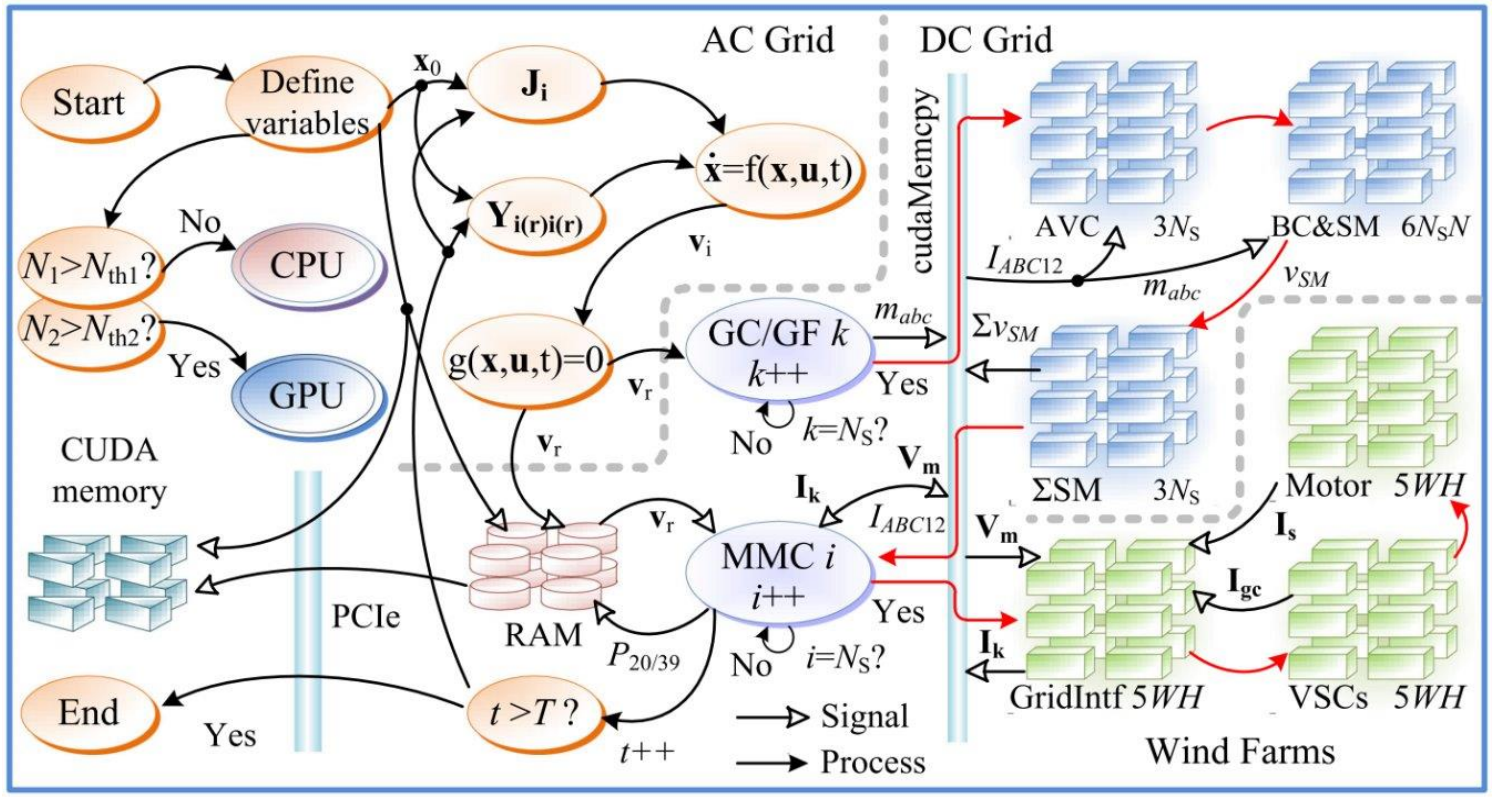
- Adaptive sequential-parallel simulation paradigm
- Sequential/parallel computing dependent on system



- Potential parallel processing region
  - MMC submodules
  - MMC inner-loop control
  - Wind turbines & control
  - Both C++ & CUDA C++
- Sequential processing region
  - MMC linear circuit
  - MMC outer-loop control
  - C++ functions

ADAPTIVE CPU-GPU COMPUTING

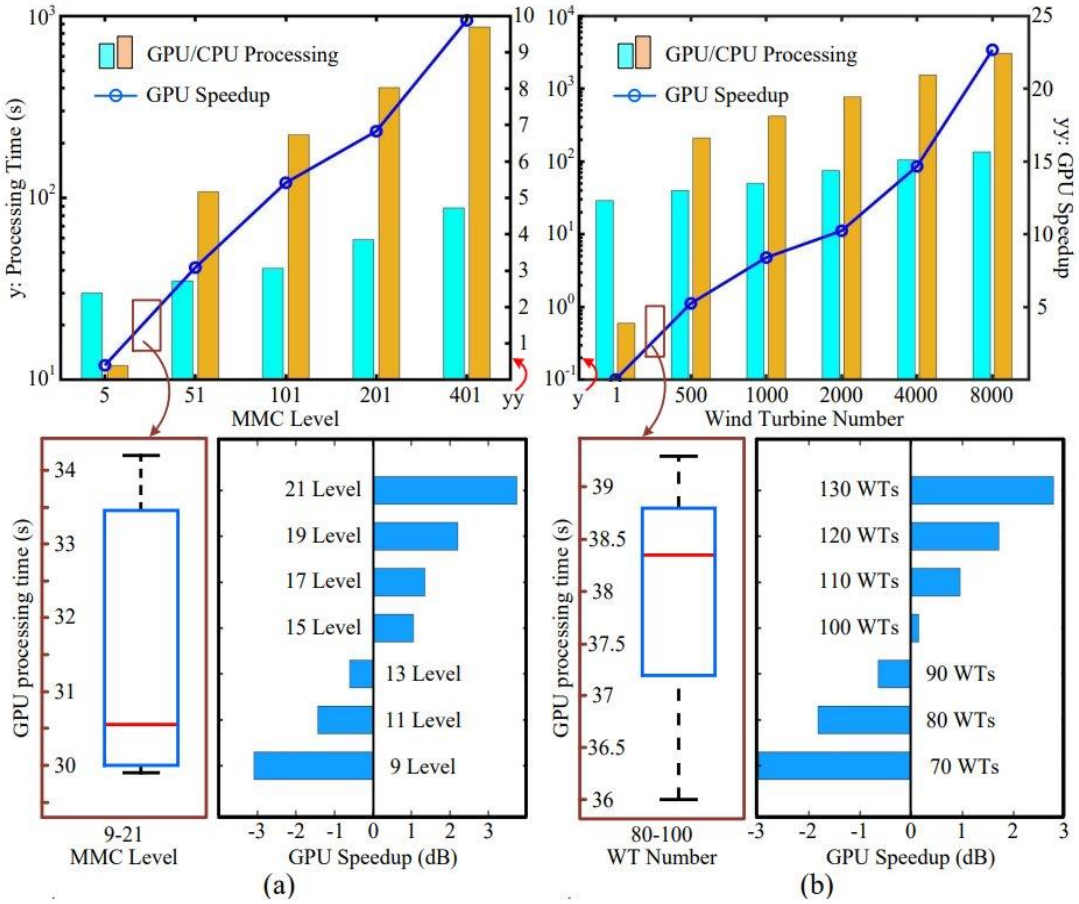
- Adaptive sequential-parallel simulation paradigm

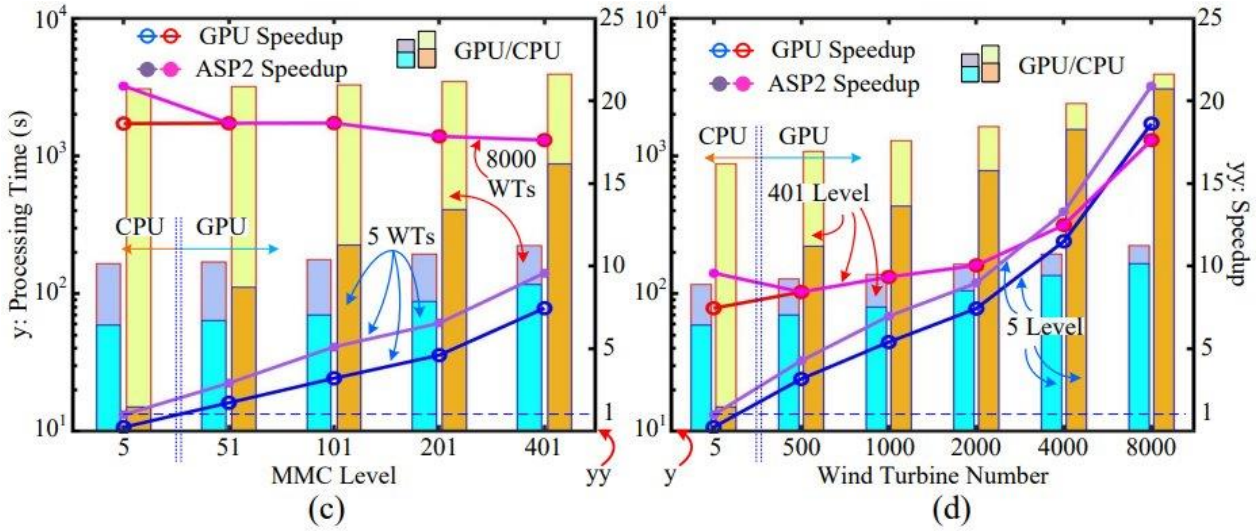


- Potential parallel processing region
  - MMC submodules
  - MMC inner-loop control
  - Wind turbines & control
  - Both C++ & CUDA C++
- Sequential processing region
  - MMC linear circuit
  - MMC outer-loop control
  - C++ functions

ADAPTIVE CPU-GPU COMPUTING

- Speed tests



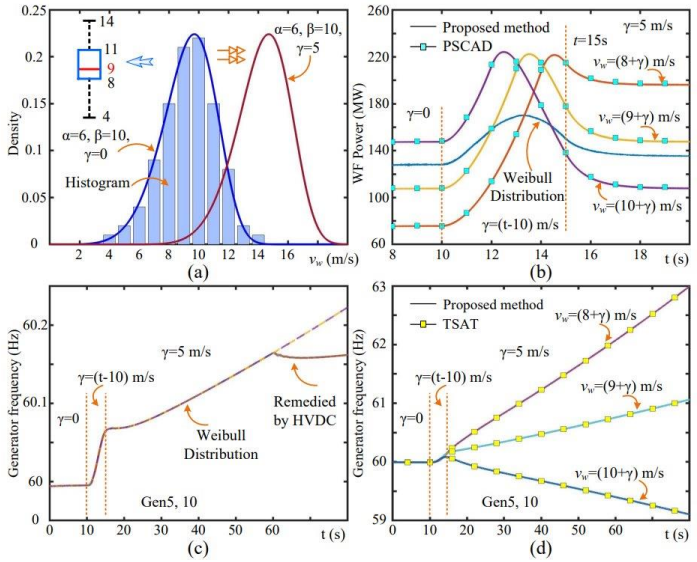


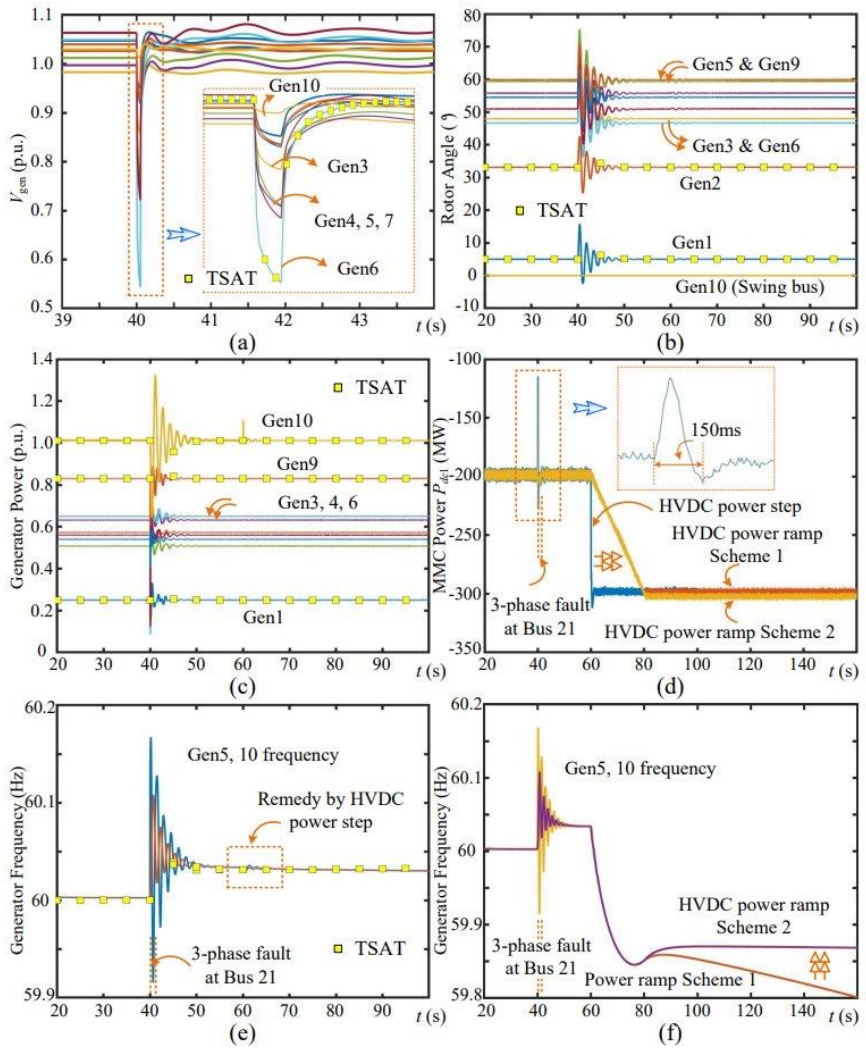
CPU & GPU simulation efficiency and threshold: (a)&(c) MMC, (b)&(d) WT

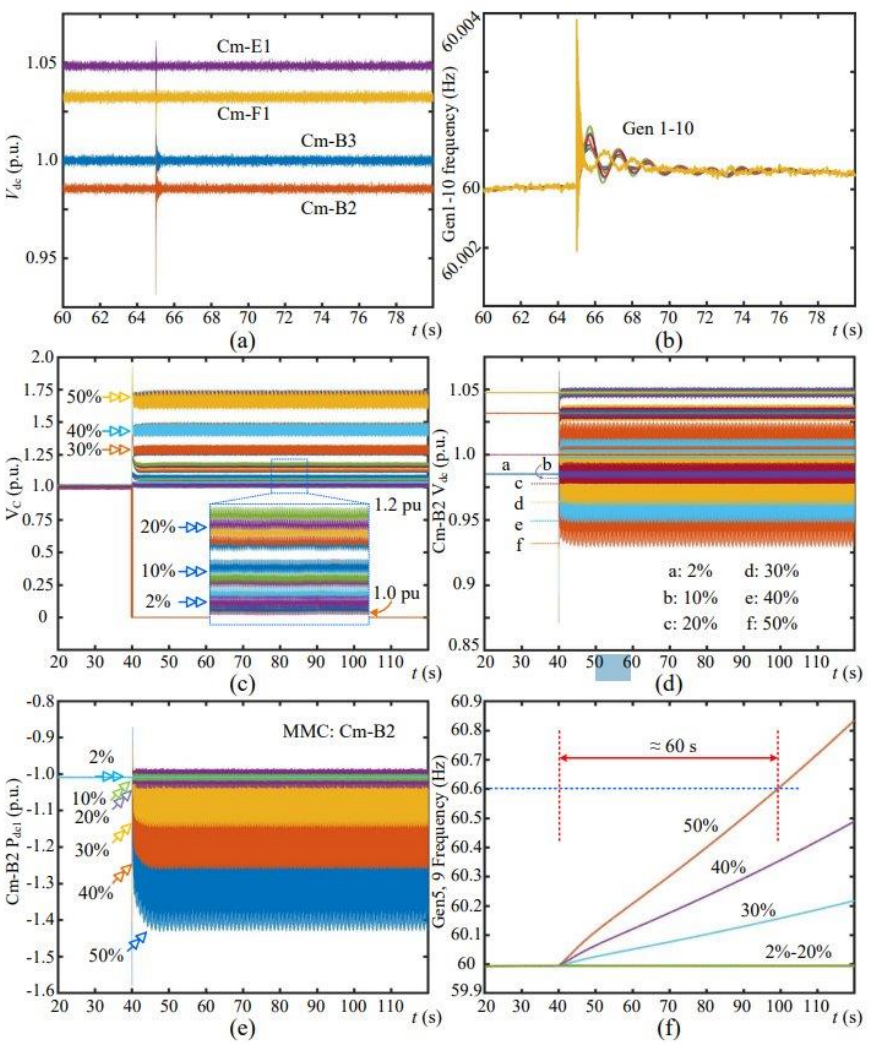
- CPU ≅ GPU when element number ≅ 100
- Dependent on CPUs and GPUs tested
- $\eta_{CPU\text{-}GPU}=\max(\eta_{CPU}, \eta_{GPU})$

ADAPTIVE CPU-GPU COMPUTING

- Simulation results
- L: WF output – Weibull distribution
- M: AC grid contingency F1
- R: DC grid contingency F2 & F3







FOR MORE DETAILS

1. N. Lin, V. Dinavahi, “Parallel high-fidelity electromagnetic transient simulation of large-scale multi-terminal DC grids”, IEEE Power Energy Technol. Syst. J., vol. 6, no. 1, pp. 59 – 70, Mar. 2019.
2. N. Lin, S. Cao, V. Dinavahi, “Massively parallel modeling of battery energy storage systems for AC/DC grid high-performance transient simulation”, IEEE Trans. Power Syst., vol. 38, no. 3, pp. 2736-2747, May 2023.
3. N. Lin, S. Cao, V. Dinavahi, “Adaptive heterogeneous transient analysis of wind farm integrated comprehensive AC/DC grids”, IEEE Trans. Energy Convers., vol. 36, no. 3, pp. 2370-2379, Sept. 2021.