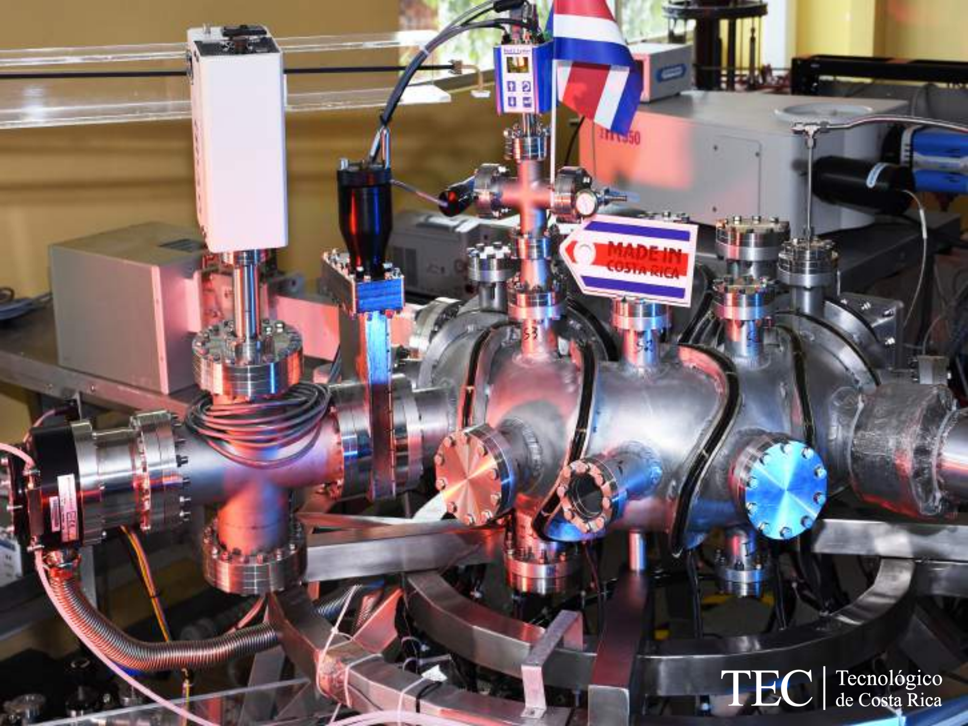
Parallel Programming Models for Computational Physics

Esteban Meneses, PhD

Advanced Computing Laboratory
Costa Rica National High Technology Center

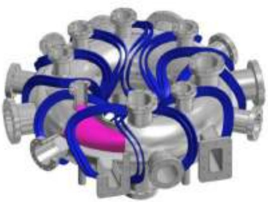
2017

Centro Nacional de Alta Tecnología


MADE IN
COSTA RICA
TEC
Tecnológico
de Costa Rica

Catalyzing Scientific Discovery

Magnetic Confinement Fusion Simulation



Particles	Steps	Serial Runtime
1	0.5M	10 min
10	0.5M	2 hours
100	2M	7 hours
1K	2M	3 days
10K	2M	30 days

30x speedup!

Outline

Parallel Computing
- Parallel Architectures
- Scientific Simulation

Parallel Programming Models
- Vectorization
- Shared-memory
- Distributed-memory
- Plasma Physics Simulation

Parallel Objects
- Load Imbalance
- Programming Model
- Execution Model

CeNAT

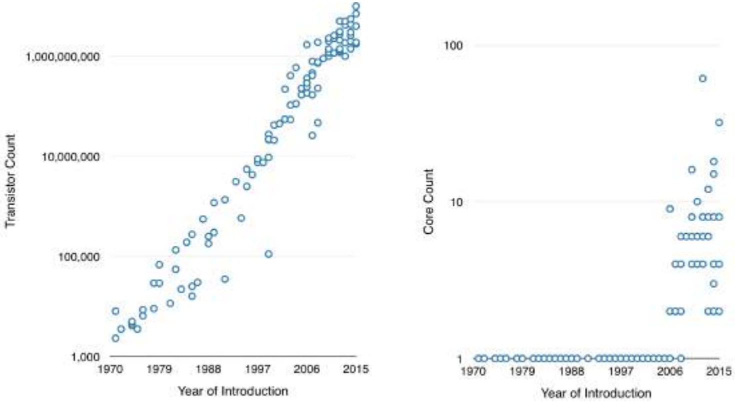
Parallel Architecture

It's everywhere

CeNAT

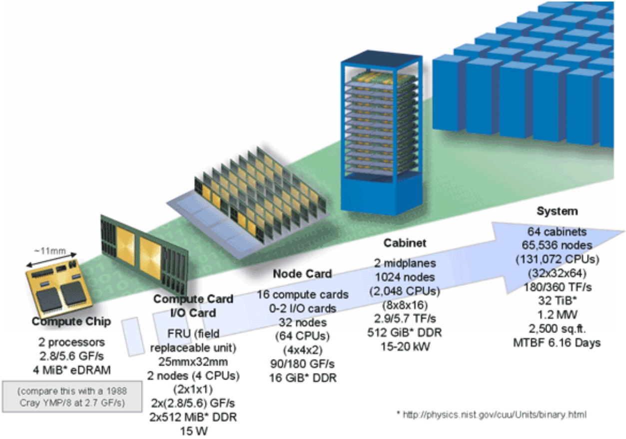
Moore's Law

Transistor density increases exponentially



Supercomputer

A hierarchical ensemble of many parts



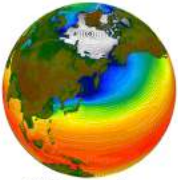
Top 500

Fastest supercomputers on Earth

Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	National Supercomputing Center in Wuxi China	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRCPC	10,649,600	93,014.6	125,435.9	15,371
2	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
3	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
4	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890
5	DOE/SC/LBNL/NERSC United States	**Cori** - Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect Cray Inc.	622,336	14,014.7	27,880.7	3,939

Simulation

Fundamental in many scientific domains



Climatology

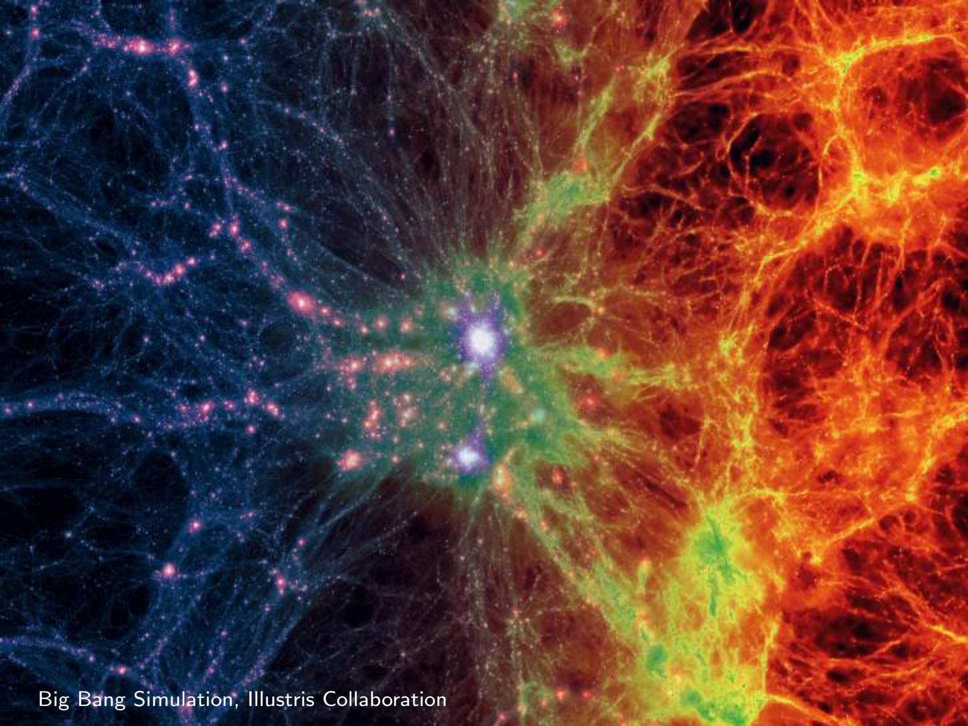
Medicine

Materials

Economics

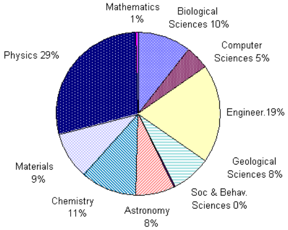
Cosmology

Entertainment

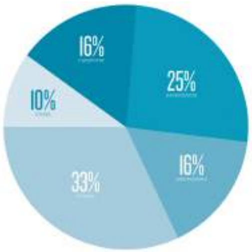

Big Bang Simulation, Illustris Collaboration

Computational Physics

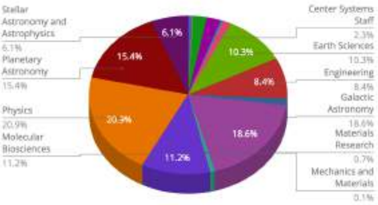
A traditional user of supercomputing power



NSF Report (1995)



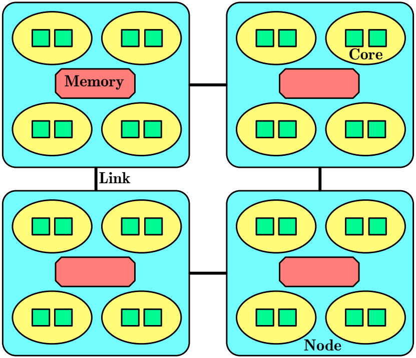
Zuse Institute Berlin (2013)



Blue Waters Supercomputer (2017)

Programming a Supercomputer

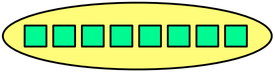
A hierarchical multi-model approach



- Models: vectorization, shared-memory, distributed-memory
- Additionally, accelerator programming

Vectorization

To exploit data parallelism

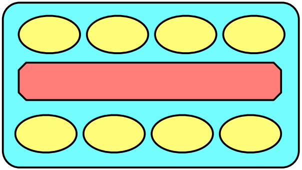


- Vector unit applies same operation to multiple values
- SIMD: single instruction multiple data
- Instruction set: MMX, SSE, AVX (up to 512 bits)
- Represents 50% of performance in some architectures
- Usually performed by compilers
- Explicit suggestions by programmer:

```
#pragma simd
for(i=0; i<n; i++)
    a[i] = a[i] + 10;
```

Shared-memory Programming

Harnessing thread parallelism



- Concurrent execution of multiple threads
- Open Multi-processing (OpenMP) standard
- Set of pragmas (annotations) for the compiler
- Incremental parallelization philosophy

```
#pragma omp parallel for collapse(2)
for(x=0; x<n; x++)
    for(y=0; y<n; y++)
        filter(image[x,y]);
```

Race Conditions

Programmers must be careful of data dependencies

- Sequential version:

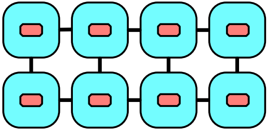
```
fibo[0] = fibo[1] = 1;
for (i=2; i<10; i++)
    fibo[i] = fibo[i-1] + fibo[i-2];
```

- (incorrect) Parallel version:

```
fibo[0] = fibo[1] = 1;
# pragma omp parallel for num_threads(2)
for (i=2; i<10; i++)
    fibo[i] = fibo[i-1] + fibo[i-2];
```

Distributed-memory Programming

Scaling parallel algorithms



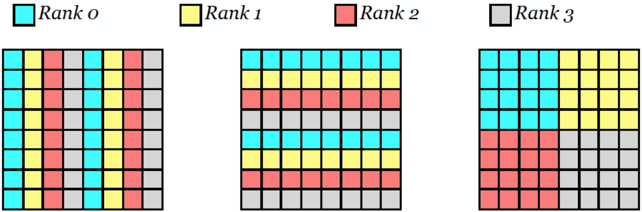
- Decomposition of problem into separate processes
- Coordination through exchange of messages
- Message Passing Interface (MPI) standard
- Design of parallel algorithm philosophy

```
// rank 0
int main(){
    MPI_Init();
    MPI_Send(data,1);
    MPI_Recv(data,1);
}
```

```
// rank 1
int main(){
    MPI_Init();
    MPI_Recv(data,0);
    MPI_Send(data,0);
}
```

Data Partitioning

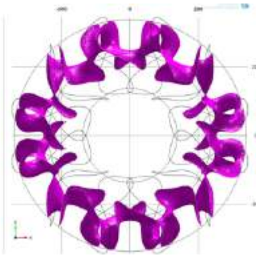
Fundamental for load balancing



- Surface-to-volume ratio
- Overlap communication with computation

SOLCTRA

Plasma physics simulation



$P_0 \leftarrow StartPoint$

loop

$K_1 \leftarrow \text{MAGNETIC FIELD}(P_0)$

$P_1 \leftarrow \frac{K_1}{2} + P_0$

$K_2 \leftarrow \text{MAGNETIC FIELD}(P_1)$

$P_2 \leftarrow \frac{K_2}{2} + P_0$

$K_3 \leftarrow \text{MAGNETIC FIELD}(P_2)$

$P_3 \leftarrow K_3 + P_0$

$K_4 \leftarrow \text{MAGNETIC FIELD}(P_3)$

$P_0 \leftarrow P_0 + \frac{K_1+2*K_2+2*K_3+K_4}{6}$

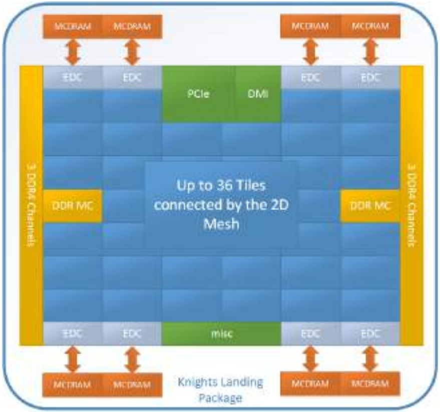
end loop

- Biot-Savart Solver for Compute and Trace Magnetic Fields
- Goal: describe the shape of confined plasma
- Classical Runge-Kutta method (RK4)

Intel Xeon Phi Knights Landing (KNL)

Architectural Overview

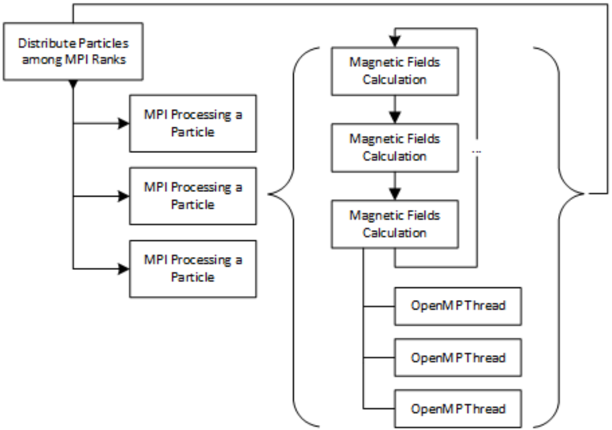
KNL Tile



KNL Mesh

Parallel SOLCTRA

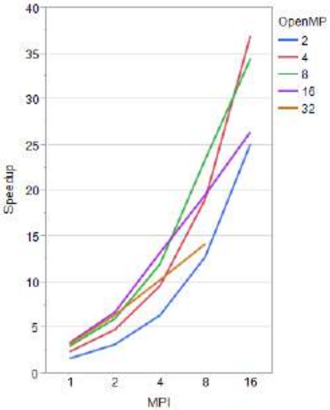
Hybrid OpenMP and MPI



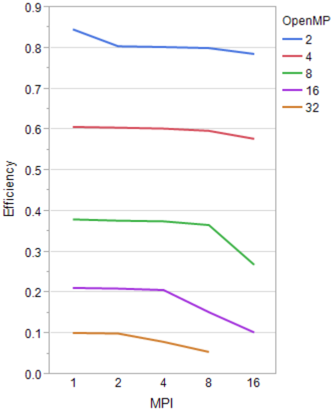
- Vectorization and other optimizations: 8.47x speedup

Parallel SOLCTRA Results

Strong scalability



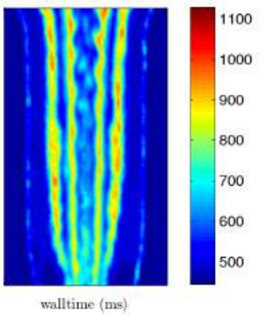
Speedup

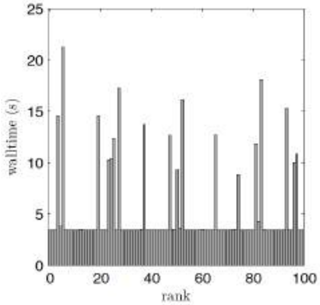


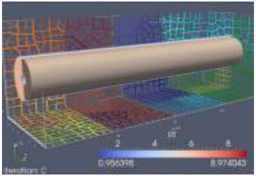
Efficiency

Simulating a Premixed Flame

Massive load imbalance





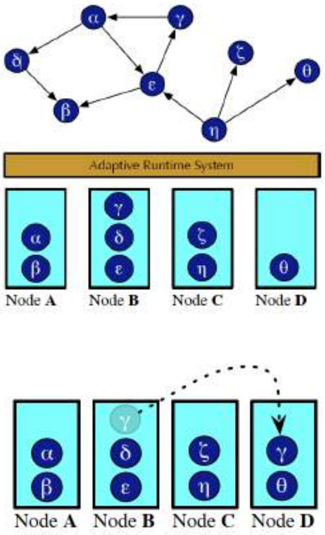


CeNAT

Parallel Objects Model

Object-oriented parallel programming

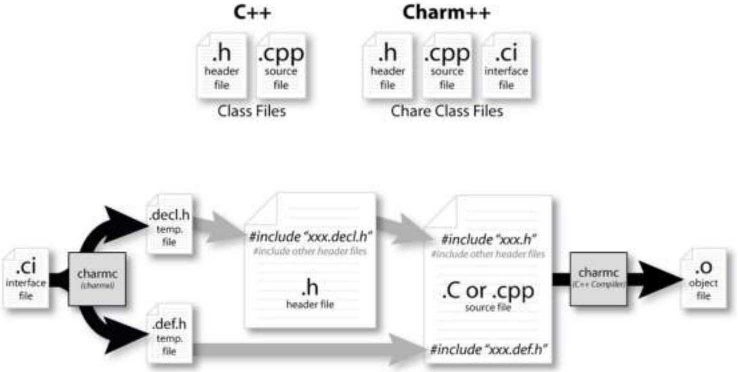
- Reactive objects with remote method interface
- Asynchronous method invocation
- Message-driven execution
- Introspective runtime system
- Pack-unpack, migration



CeNAT

Charm++

Actively developed since mid 90s

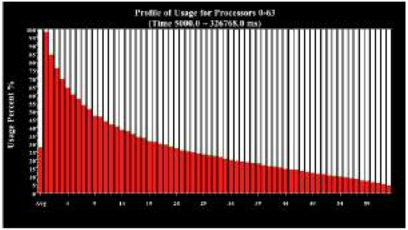


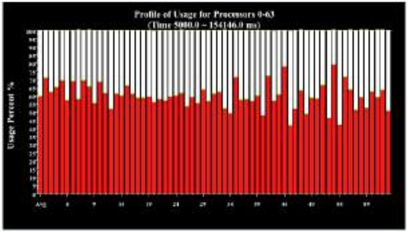
- Objects are called *chares*
- Chare arrays are the main object collection
- Chares export remote *entry methods*

Load Balance

A complex optimization problem

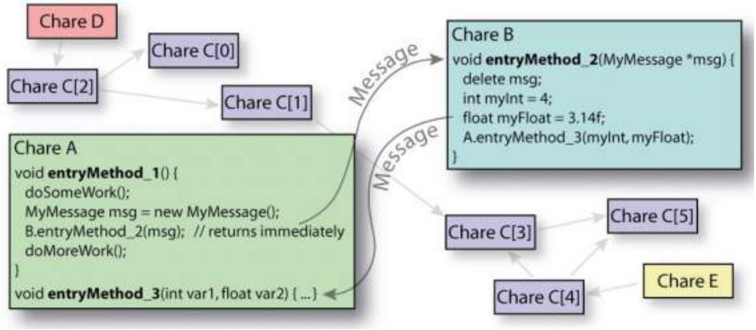
- Suboptimal, but fast heuristic algorithms
- Goal: avoid overloaded nodes
- Runtime collects load and communication data
- Dynamic load balance
- Principle of persistence





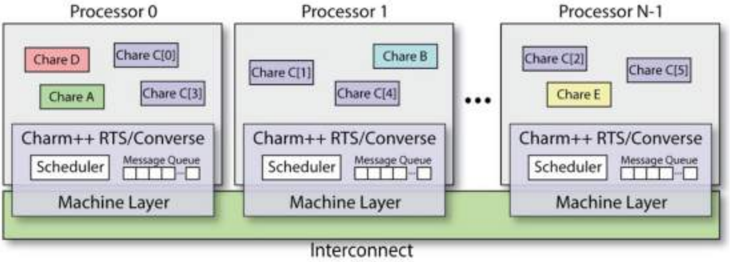
Global Object Space

Entry methods can be called from anywhere



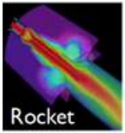
Charm++ Runtime System

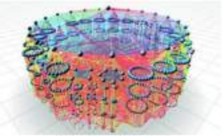
Multiple layers with different abstraction



Charm++ Applications

Span multiple scientific domains




Contagion


PSTIP


Engineering

CeNAT

Acknowledgments

- Images from stellarator simulations provided by Prof. Iván Vargas from Costa Rica Institute of Technology
- SOLCTRA experimental results collected by Luis Diego Chavarría
- Media from computational fluid dynamics courtesy of Dr. Patrick Pisciuneri and Prof. Peyman Givi from University of Pittsburgh
- Charm++ images courtesy of Prof. Laxmikant Kalé from University of Illinois at Urbana-Champaign

Advanced Computing Laboratory at CeNAT

Kabré Supercomputer

- 20 Intel Xeon Phi KNL nodes:
  - 96 GB main memory
  - 256 GB flash memory
  - 4TB storage
- 1280 cores, 5120 hardware threads
- 53.24 TeraFlops peak performance
- Other resources:
  - Advanced network
  - Training programs: R, Python, OpenMP, MPI, OpenACC
  - Scholarship program
  - Scientific Computing Research Network (RICC)

www.cenat.ac.cr/cnca

www.facebook.com/centro.nacional.de.alta.tecnologia/

Concluding Remarks

- Modelling translates problems from one domain into computer science
- The era of parallel computer architecture fuels advanced computing applications
- Parallel computing constantly redefines tractability of algorithms

Thank You!
Q&A

emeneses@cenat.ac.cr

CeNAT
Centro Nacional de Alta Tecnología