# Application Features Diagnosing Methodology of High Performance Computing

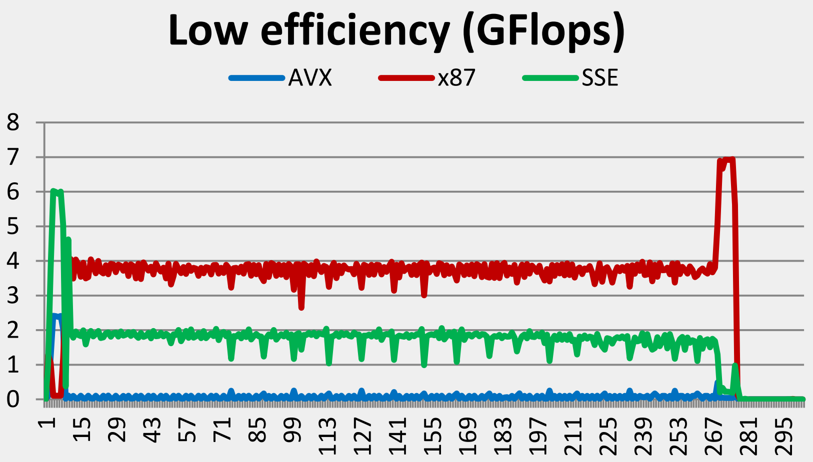
Yu Liu  Ph.D

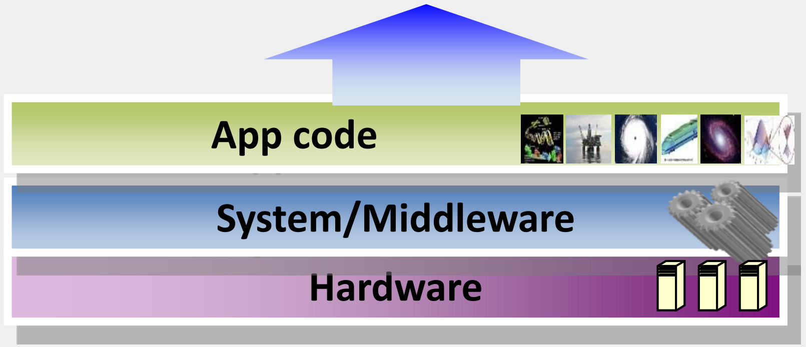
State Key Laboratory of High-end Server & Storage Technology

# The Problem of HPC

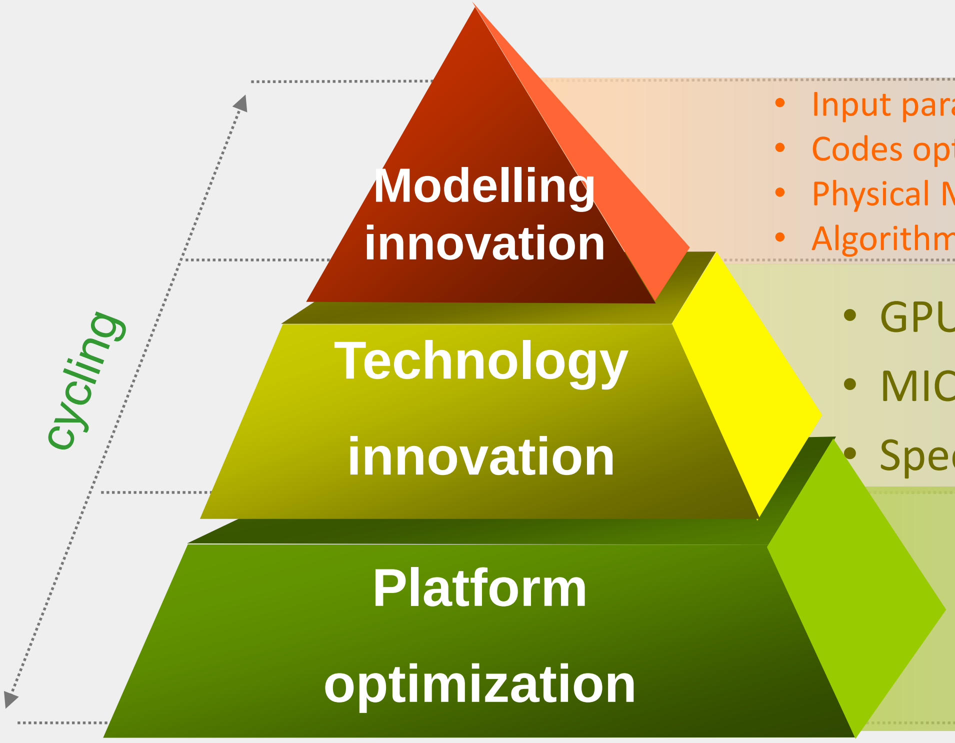
- software architecture can't keep up with the development of the hardware architecture, the execution efficiency is low
- Low hardware utilization
- obtain the real time features of app is difficult
- Hardware can not meet the requirement of different application field
- Lack of skilled engineers with both application and HPC research background

**Bottleneck 2: Low hardware efficiency**



# Our Strategy




- Input parameters optimization
- Codes optimization
- Physical Modelling innovation
- Algorithm innovation

- GPU
- MIC
- Special machines

- Application features analyzing
- Hardware optimization
- Software optimization

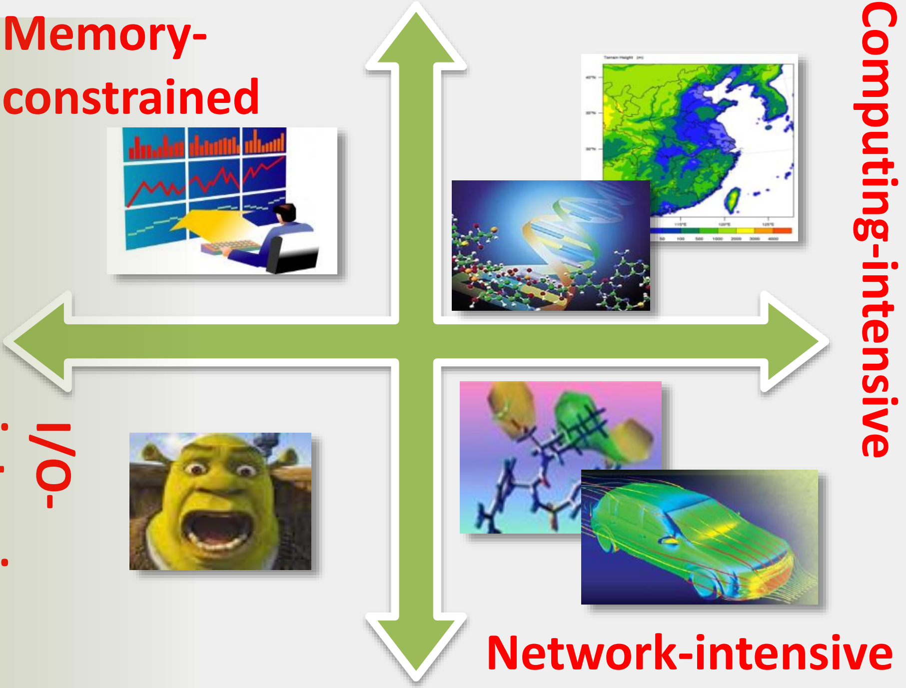
- Based on platform
- Guided by application
- Pointed by theory

**The way to achieve excellent performance**

# System Platform Optimization

**The different requirements of HPC application :**

- CPU
- Memory
- Storage
- Network

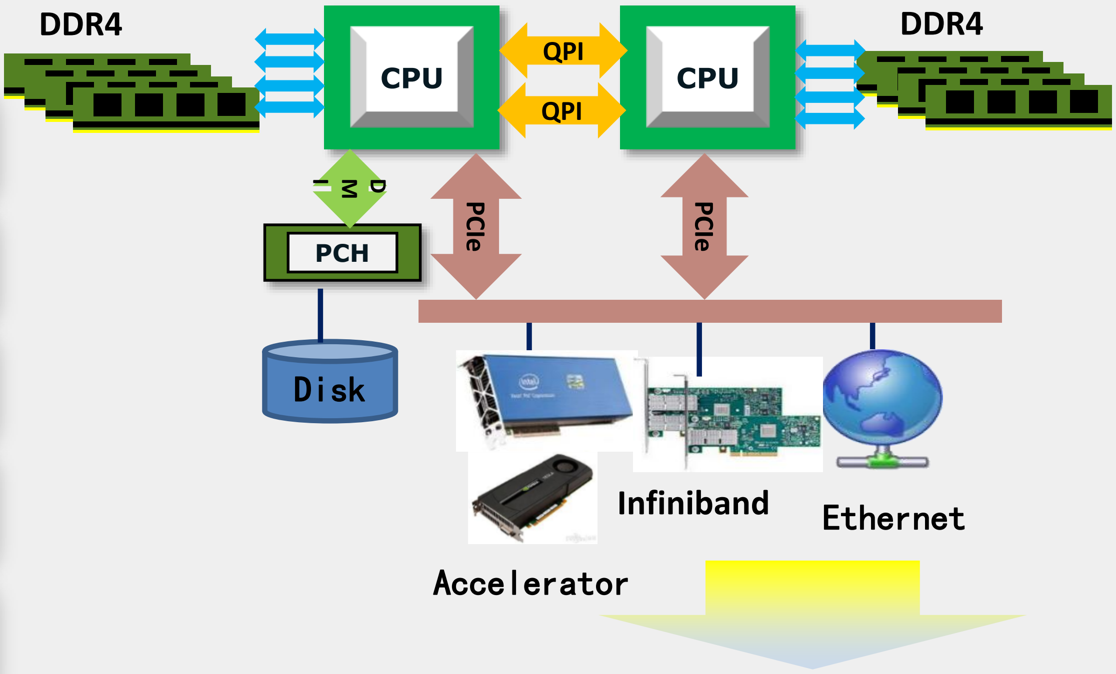



**Balance a variety of computing requirement and build a high-scalable, high-efficient HPC system to maximize the existing application's performance.**

# System Platform Optimization

**Platform Optimization**

- Process : NUMA
- Mem access : Cache
- Comm : PCIe
- I/O : Disk, SSD
- Network : IB, Ethernet
- Accelerator : GPU、MIC




CPU : Micro-architecture

II-EP Platform

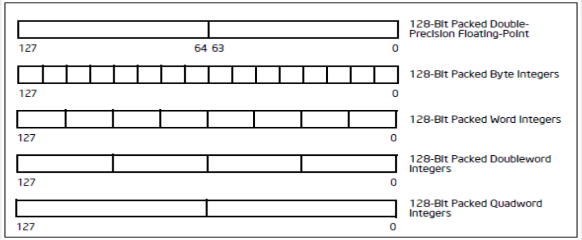
# System Platform Optimization

**Execution efficiency optimization**

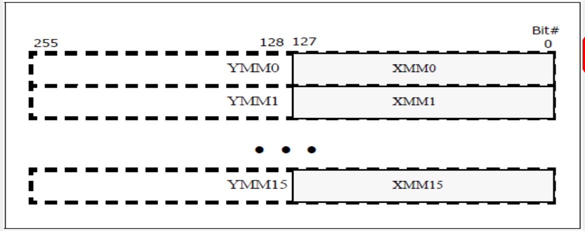
**Instruction Fetch：Men access, Cache**

**Instruction set：AVX、FMA**

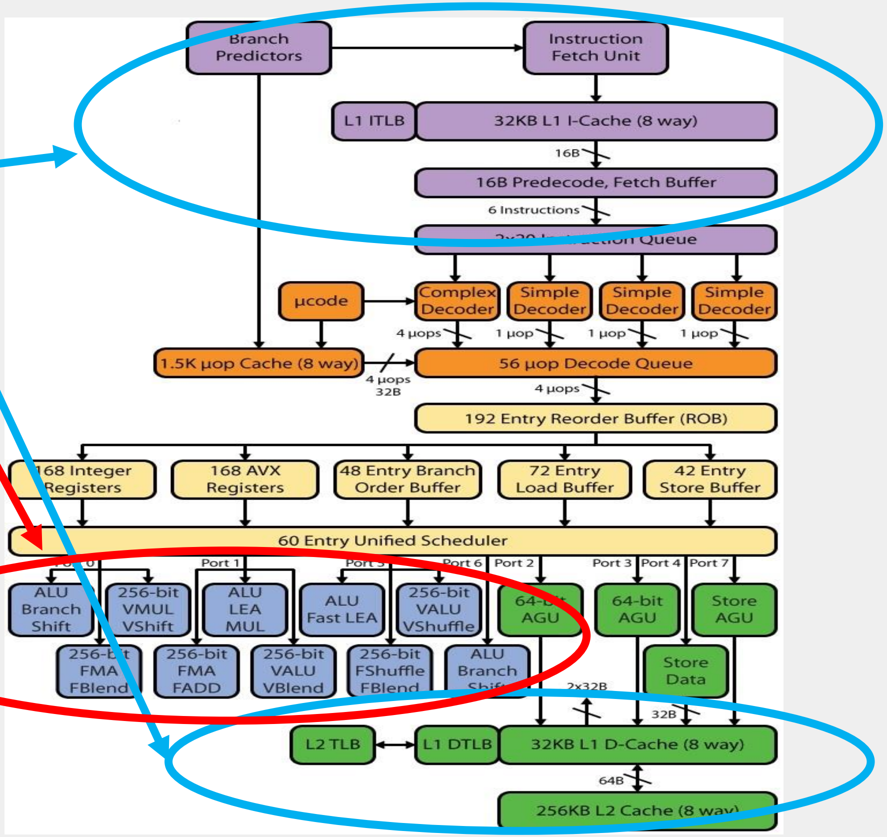
**SSE**




**AVX**




**How to get these item?**



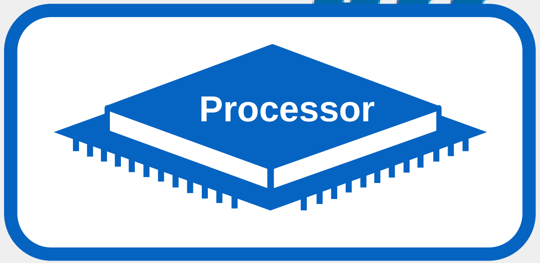

**Haswell Micro-architecture**

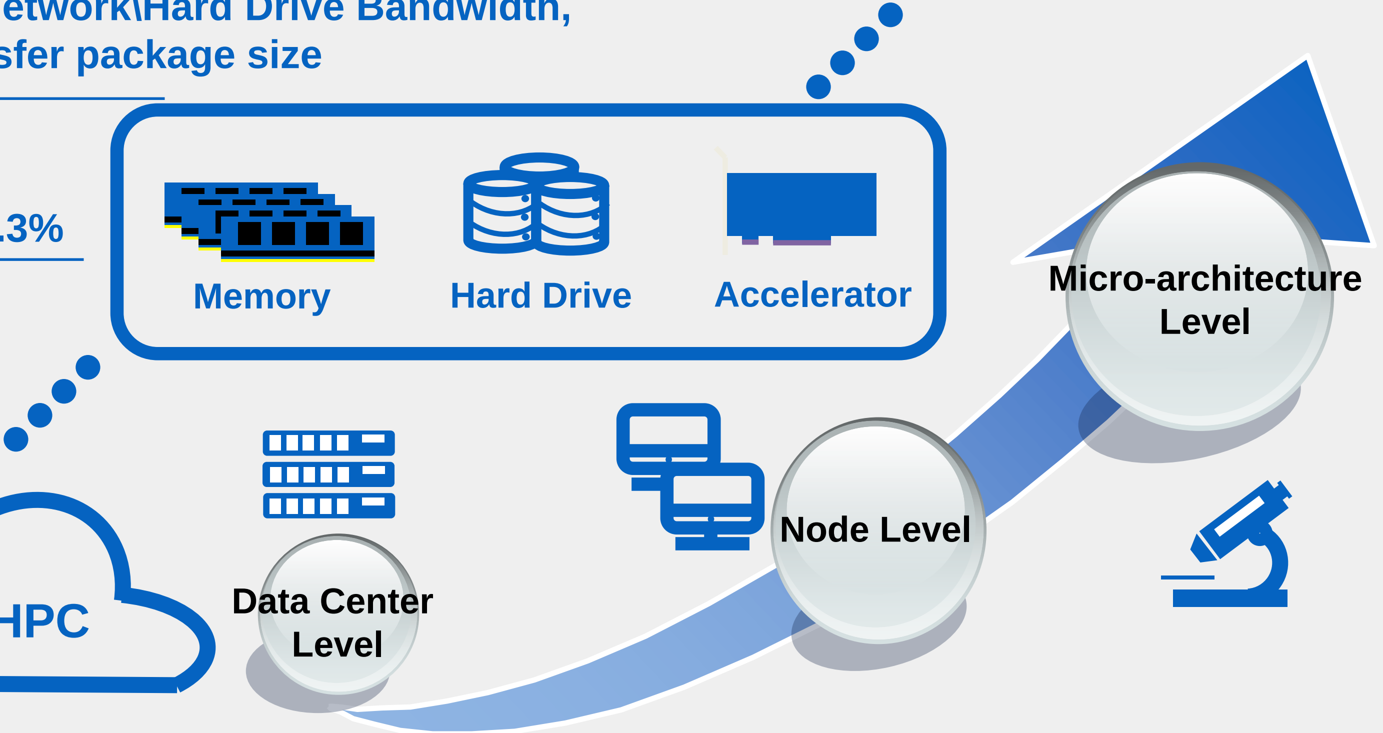
# System Platform Optimization

x87\AVX\SSE\ instruction sets, Instruction vectorization rate, Clock Cycle per Instruction, Cache miss\hit

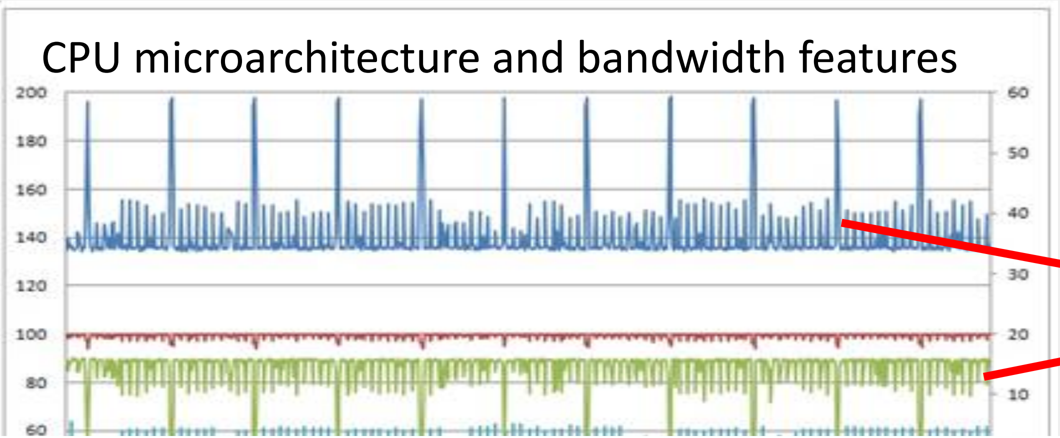
Memory\PCI-E\Network\Hard Drive Bandwidth, IOPS, Data Transfer package size

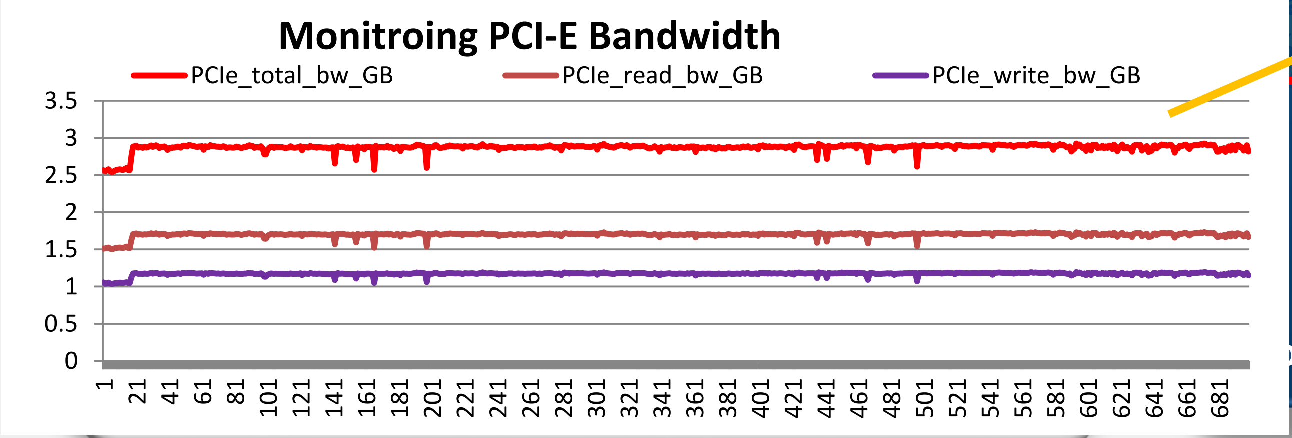
Support: > 1024 nodes
Cost: user% < 0.3%

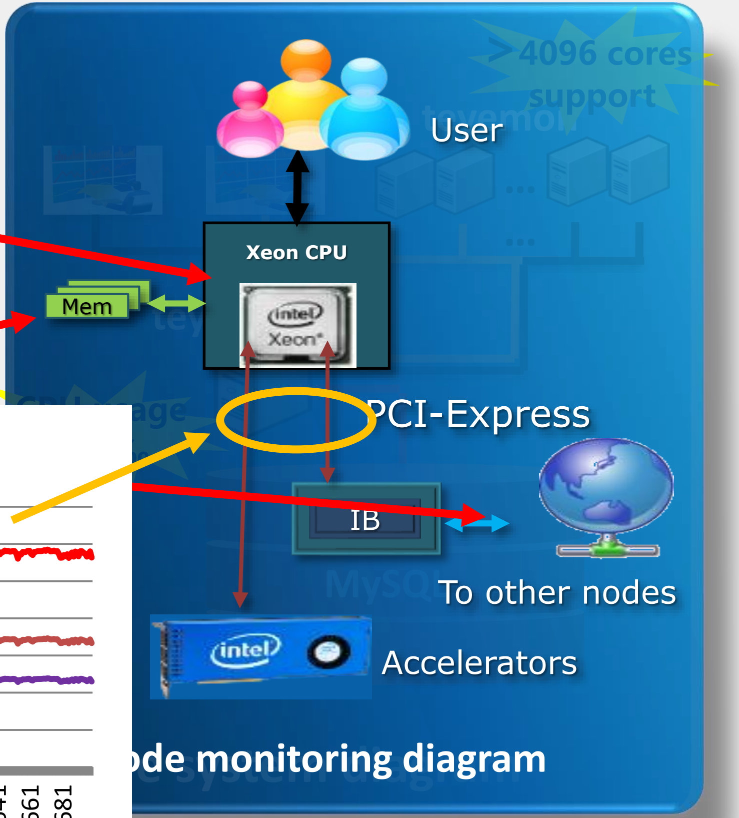


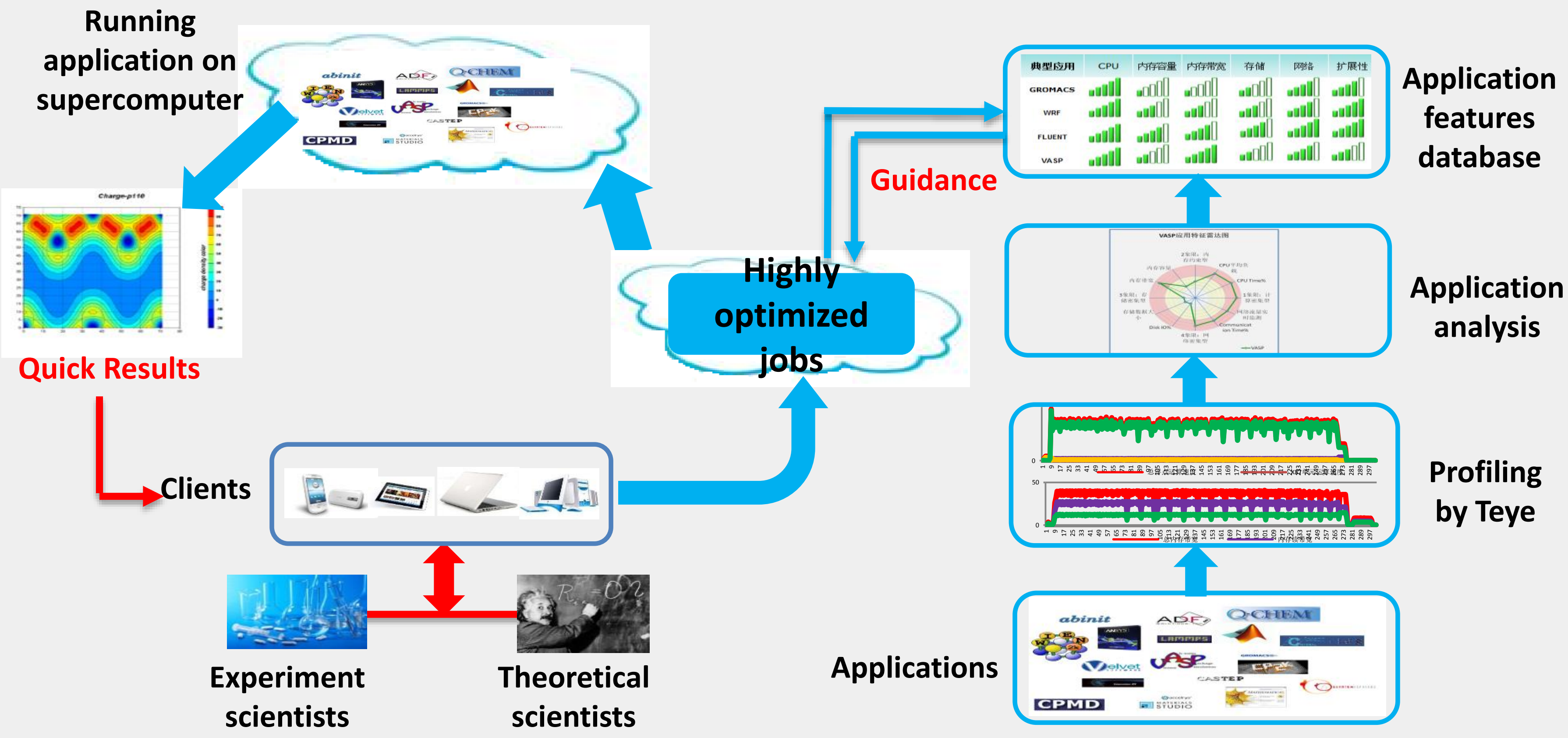

# Teye Application Features Analyzer



# Diagnose and Optimize Application



# Application optimization CASE---VASP

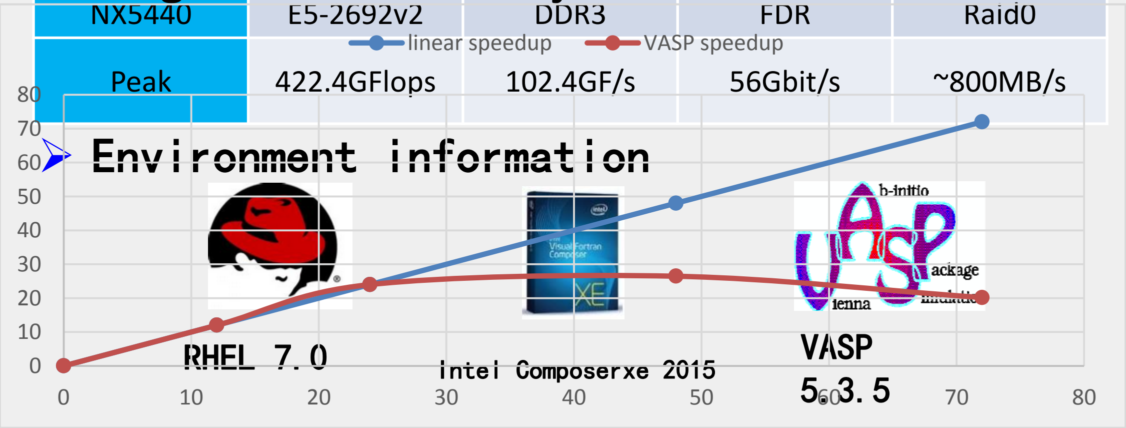
**VASP: VASP JOB optimized by Teye**

- JOB Information:
- Original Performance:

| | Nodes | Cores | Tot. time (m) | Electron relation setps | Electron relation time(m) |
|---|---|---|---|---|---|
| JOB | 2 | 48 | 285 | 33 | 8.63 |

| | CPU | Memory | Network | I/O |
|---|---|---|---|---|
| NX5440 | E5-2692v2 | DDR3 | FDR | Raid0 |
| Peak | 422.4GFlops | 102.4GF/s | 56Gbit/s | ~800MB/s |

- Original Scalability:

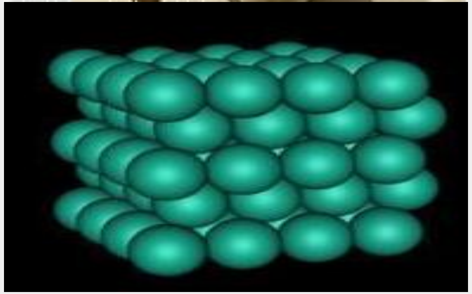



- Environment information

RHEL 7.0

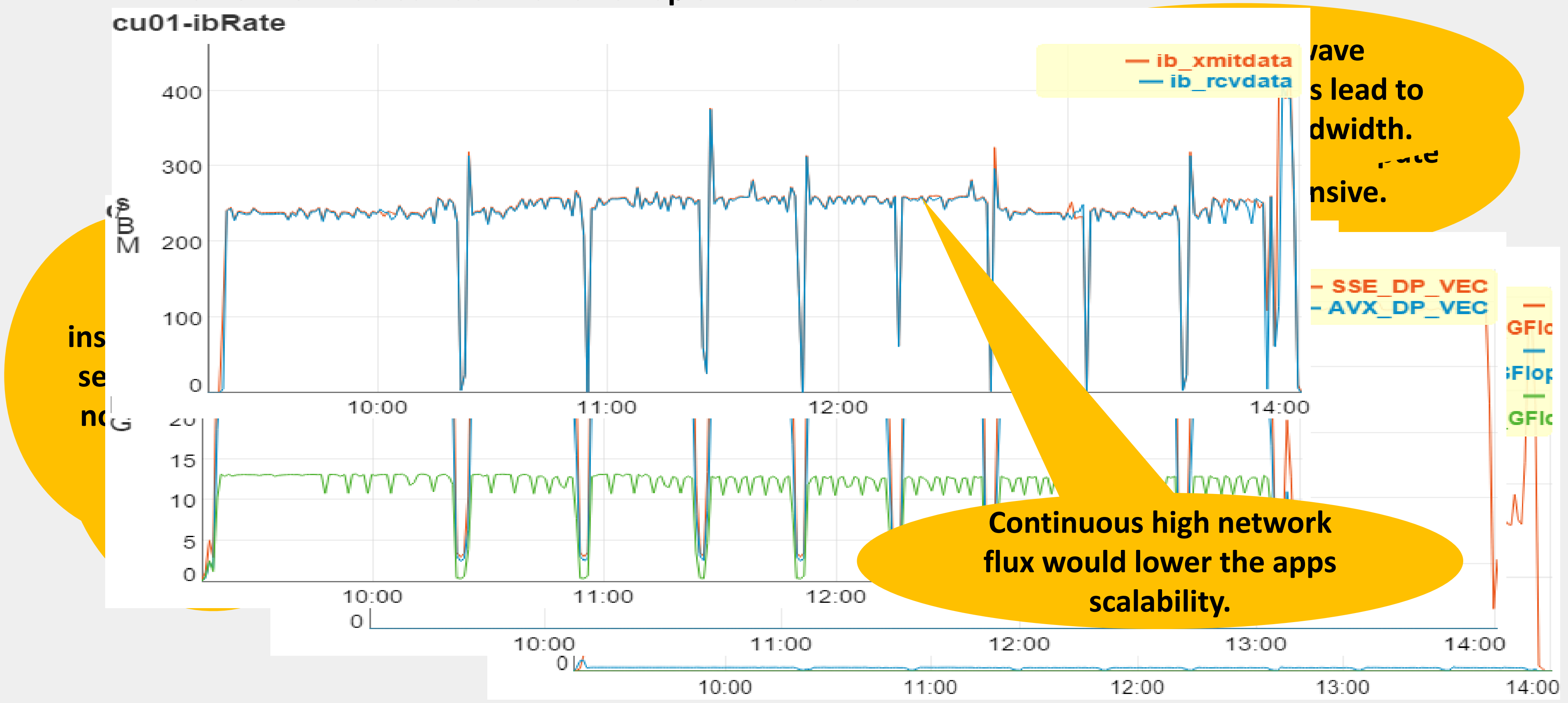
Intel Composerxe 2015

VASP 5.3.5



# Application optimization CASE---VASP

➤ Runtime Features Before Optimization



# Application optimization CASE---VASP

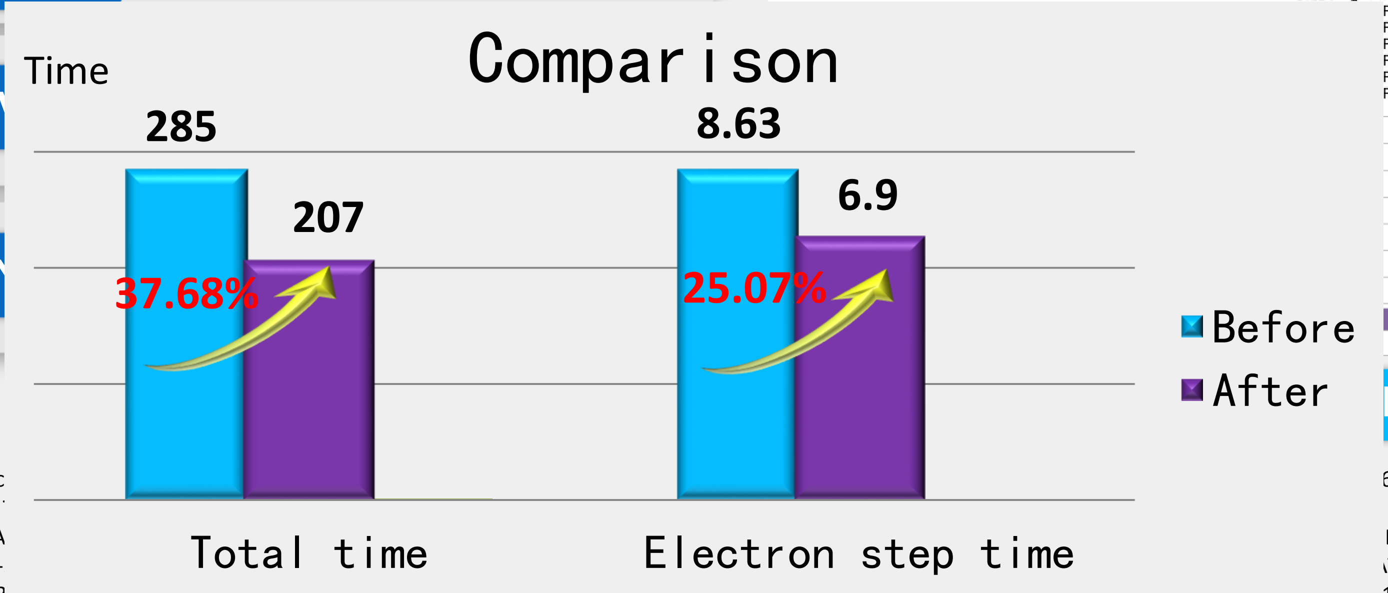
HSS

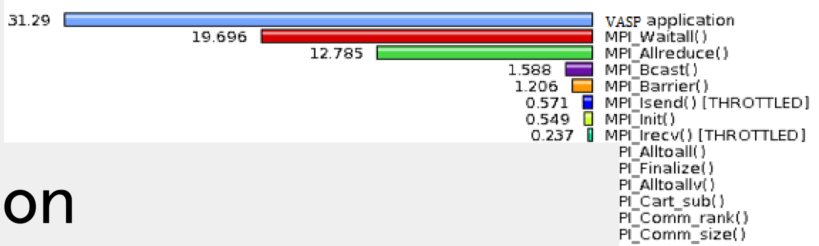
- Optimizing idea and method

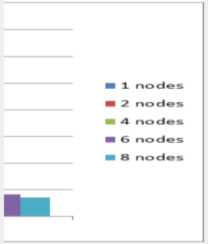
CPU 256bit AVX, FMA

M

N




block

mpiexec 6749;3:166750-
386027;
I_MPI_A ERV '2:0-0;1:1-
45;4:46- TTERV 1:0-
2147483647' -genv I_MPI_ADJUST_REDUCE '3:0-0;4:1-14;2:15-20;4:21-32;2:33-75;4:76-1345;2:1346-3809;4:3810-4626;2:4627-51262;5:51263-264282;1:264283-2147483647' -genv I_MPI_ADJUST_ALLREDUCE '2:0-0;1:1-1057;6:1058-2131;2:2132-131072;8:131073-262144;2:262145-554389;8:554390-2147483647' -genv I_MPI_ADJUST_REDUCE_SCATTER '3:0-0;5:1-4;1:5-156231;4:156232-2147483647' -genv I_MPI_ADJUST_ALLTOALL '1:0-4555;3:4556-2147483647' -genv I_MPI_ADJUST_ALLTOALLV 1 –machinefile hostfile –np XXX vasp

# Application optimization CASE---VASP

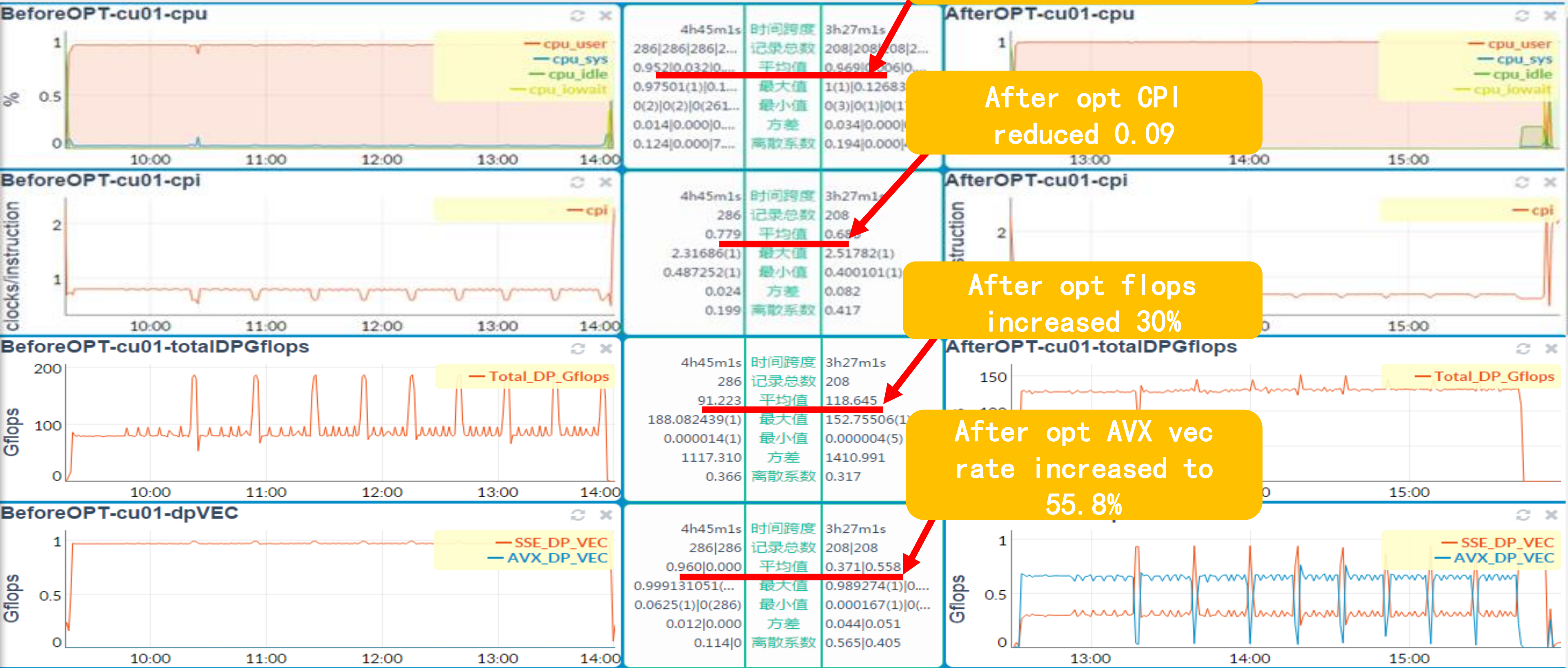
➢ Performance after opt: CPU

After opt CPU usr% increased 1%

After opt CPI reduced 0.09

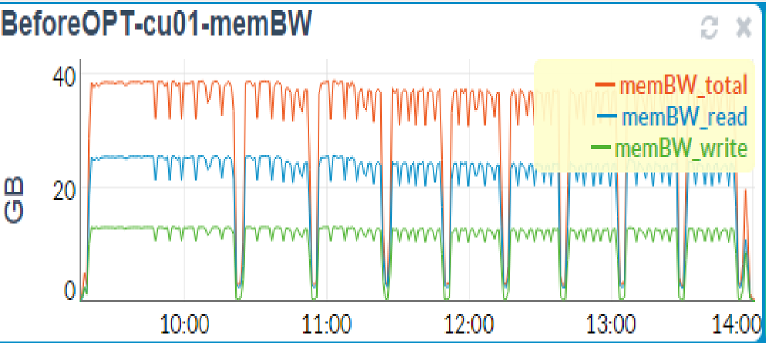
After opt flops increased 30%

After opt AVX vec rate increased to 55.8%



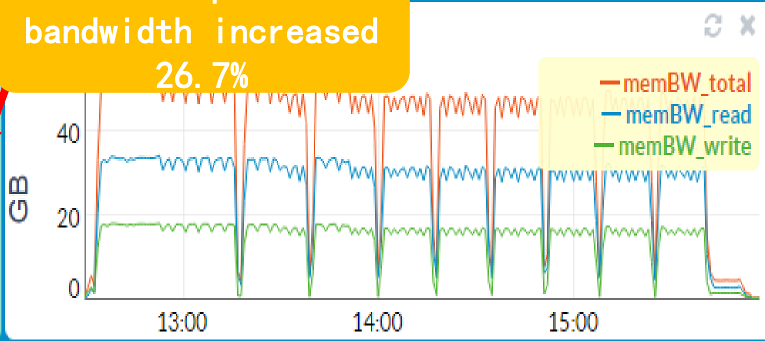
# Application optimization CASE---VASP

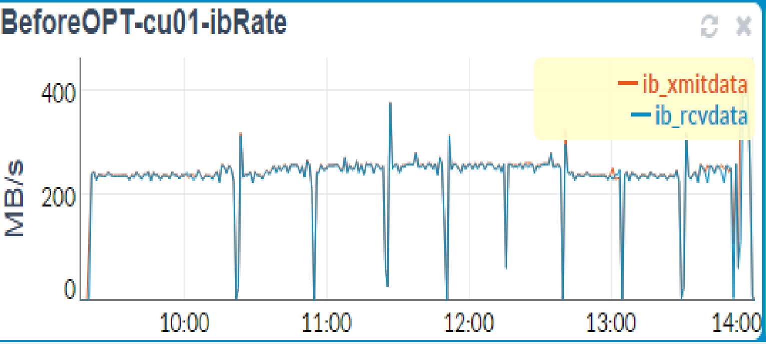
- Performance after opt: Bandwidth



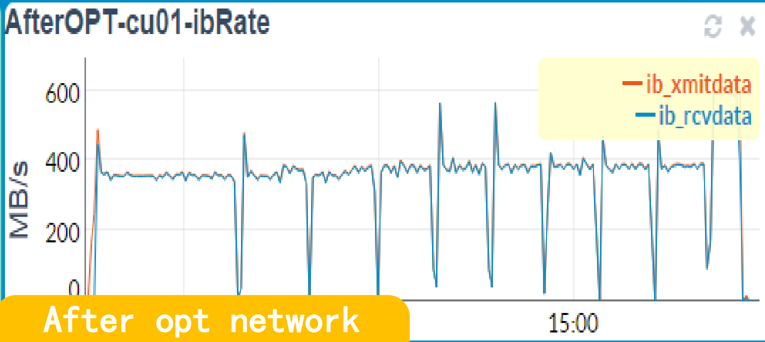

| Before | | After |
|---|---|---|
| 4h45m1s | 时间跨度 | 3h27m1s |
| 286\|286\|286 | 记录总数 | 208\|208\|208 |
| 31.850\|21.002\|... | 平均值 | 40.378\|26.301\|... |
| 38.8612(1)\|25.... | 最大值 | 52.1846(1)\|34.... |
| 0.259138(1)\|0.... | 最小值 | 0.51921(1)\|0.2... |
| 125.726\|53.00... | 方差 | 253.368\|104.9... |
| 0.352\|0.347\|0.... | 离散系数 | 0.394\|0.390\|0.... |

After opt mem bandwidth increased 26.7%



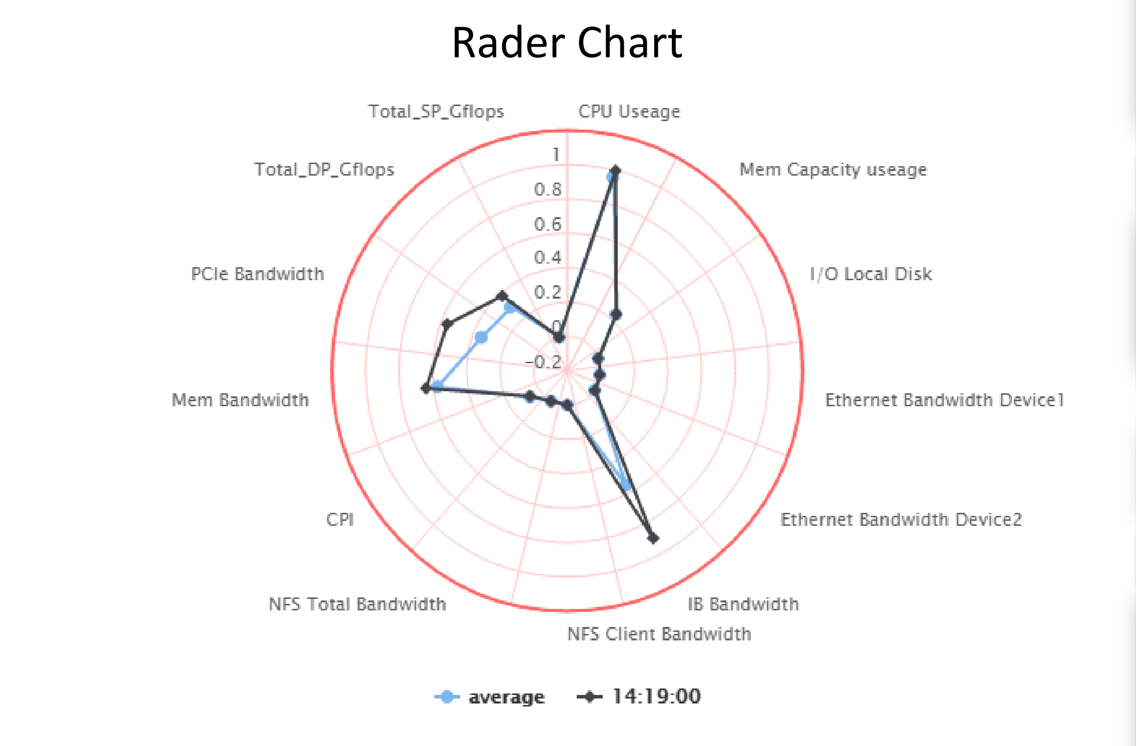

| Before | | After |
|---|---|---|
| 4h45m1s | 时间跨度 | 3h27m1s |
| 286\|286 | 记录总数 | 208\|208 |
| 235.466\|231.9... | 平均值 | 348.336\|342.7... |
| 420.25333(1)\|... | 最大值 | 630.38(1)\|609... |
| 0(2)\|0(2) | 最小值 | 0(5)\|0(5) |
| 3993.644\|415... | 方差 | 15128.731\|15... |
| 0.268\|0.278 | 离散系数 | 0.353\|0.364 |




After opt network flux increased 47.9%

# Application optimization CASE---VASP

➢ VASP Runtime Features




**Compute Intensive:**

High DP floating-point operation app. Suitable for instruction vectorization.

**Mem bandwidth sensitive:**

High mem bandwidth and low mem capacity requirement.

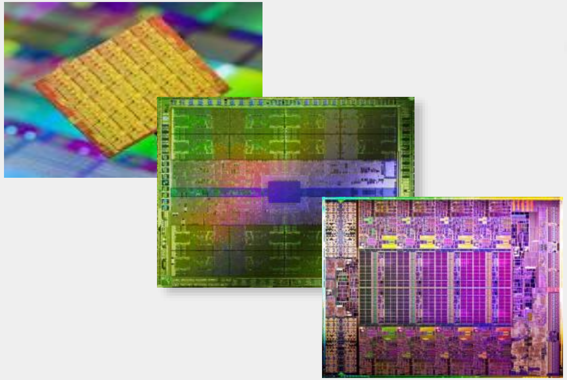
**Low I/O:**

Low I/O requirement.

**Network Intensive:**

For some Algorithm there exist continuous network flux. High bandwidth and low latency would be benefit.

# Technology Innovation



Compute

New technology

FinFET

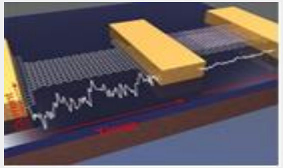
22nm → 5nm → ?

Now

2019

New material

graphene?



Power wall

Multi-cores → 16X

Accelerate computation

MIC

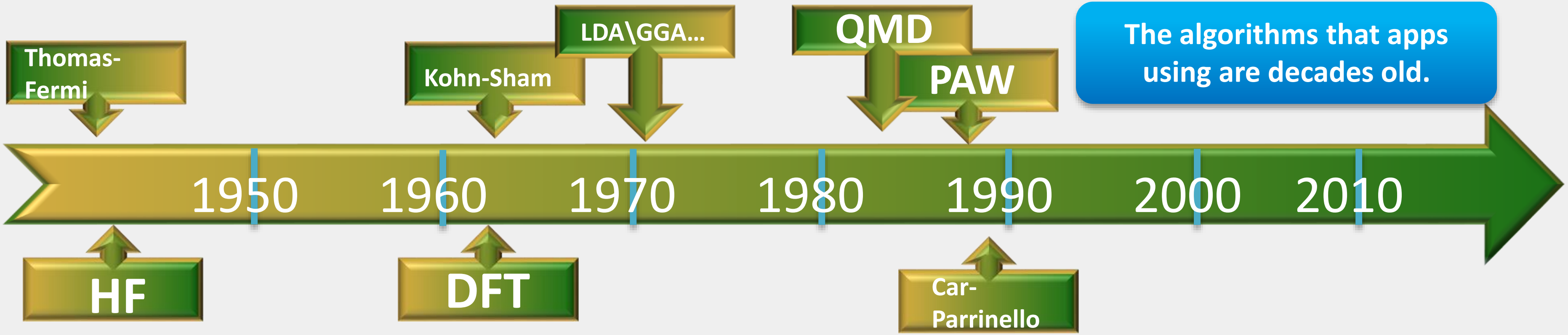
GPU

FPGA

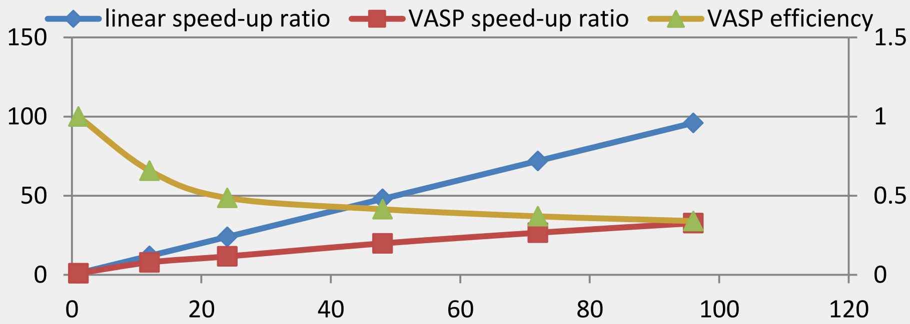
Special machines

# Modelling Innovation

The algorithms that apps using are decades old.

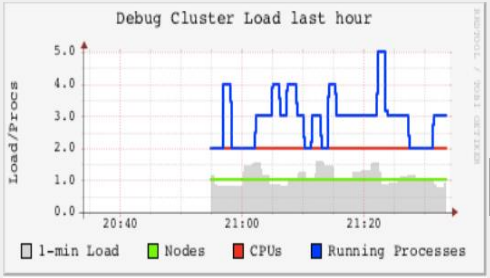


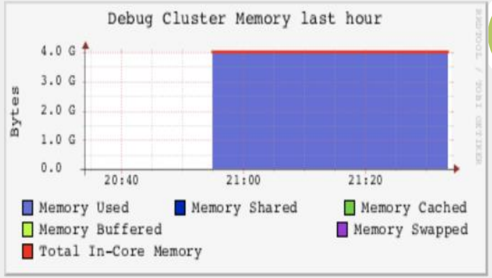

| | Parallel mode | | | |
|---|---|---|---|---|
| | band | basis | K point | Spin |
| Degree of parallelism | ~$10^2$ | ~$10^2$ | ~$10^2$ | 10 |

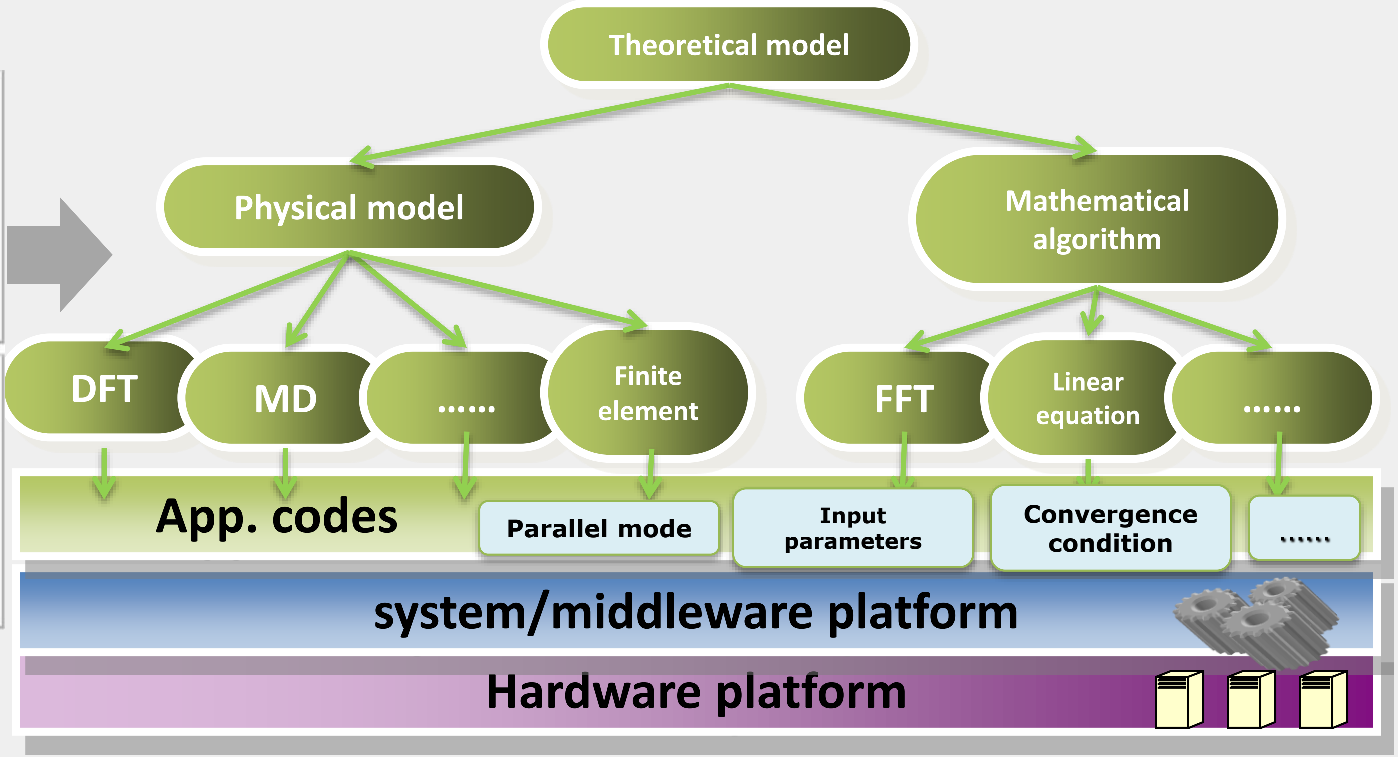



Modelling innovation is the key of breakthrough performance

# Modelling Innovation



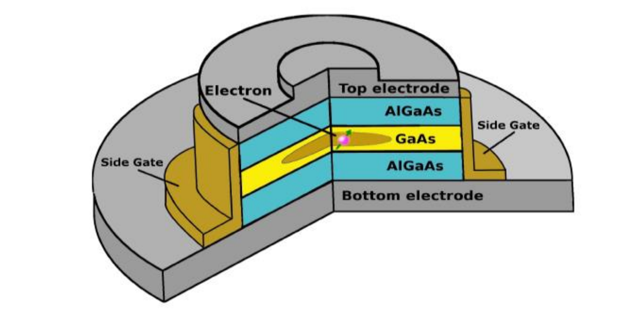



# Modelling Innovation

## CASE B. Customer in-house codes for investigating novel semiconductor materials from Chinese Academy of Sciences (The 973 Project)

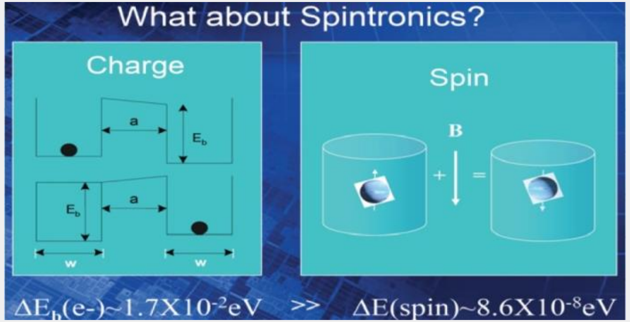
❖ **Research background:**

This project investigates theoretically the spin-related physical properties of few-electron semiconductor quantum dots. From which we can control the spin states by tuning the spin-orbit interaction strength (SOI). And the SOI can be easily controlled by external electric field. This results shows great promising for realizing useful spintronics which is significant to electronic information industry.




❖ **Physical modelling and algorithm analyzation:**

The original codes use fully Configuration Interaction (CI) method for simulating few-electron systems and use QR method for matrix diagonalization.




❖ **Performance before optimization:**

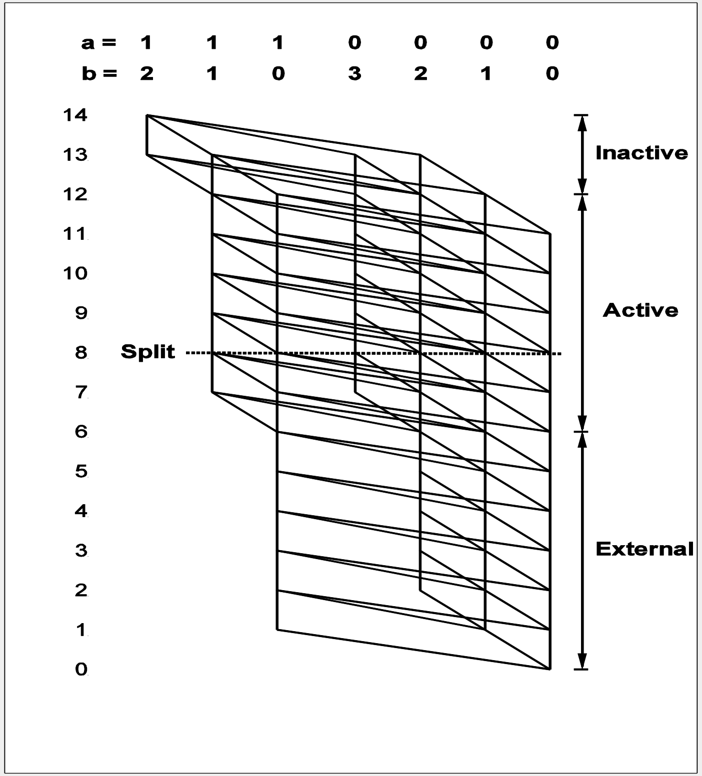
Huge memory consumption and poor parallelism. It can use only ten cores and the largest matrix scale is less than 10^4. Thus, the system scalability this code can simulate is less than 3 electrons.

# Modelling Innovation

## ❖ Optimization analysis

**(Guidance the codes implementation via theory and Modelling innovation)**

- Change the full CI to partial CI methods.(reduce the calculation of weak interactions)
- Change the QR method to Iteration method(improve the degree of parallelism)
- Develop the codes by Fortran which is 10% faster than that of C/C++
- Using vectored instructions.
- Optimize the collective communication.



# Modelling Innovation

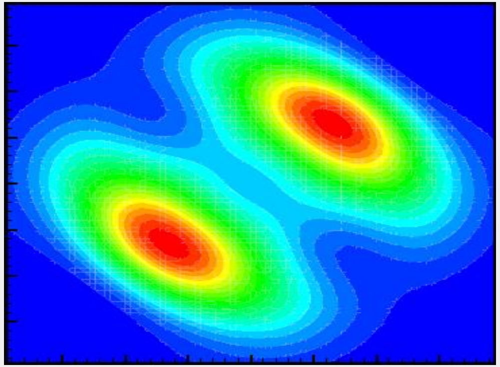
## ❖ Optimization results

**Before**

Max matrix scale~10^4

Degree of parallelism~10 cores

Simulate system~3 electrons



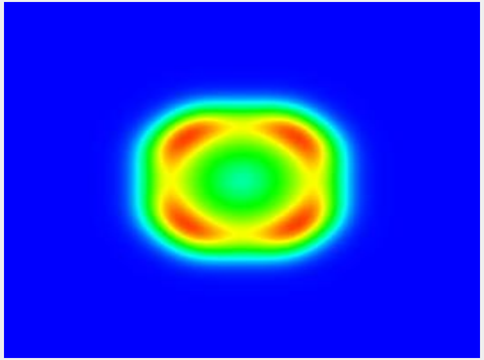
**N=2**

**After**

Max matrix scale~10^8

Degree of parallelism~1000 cores

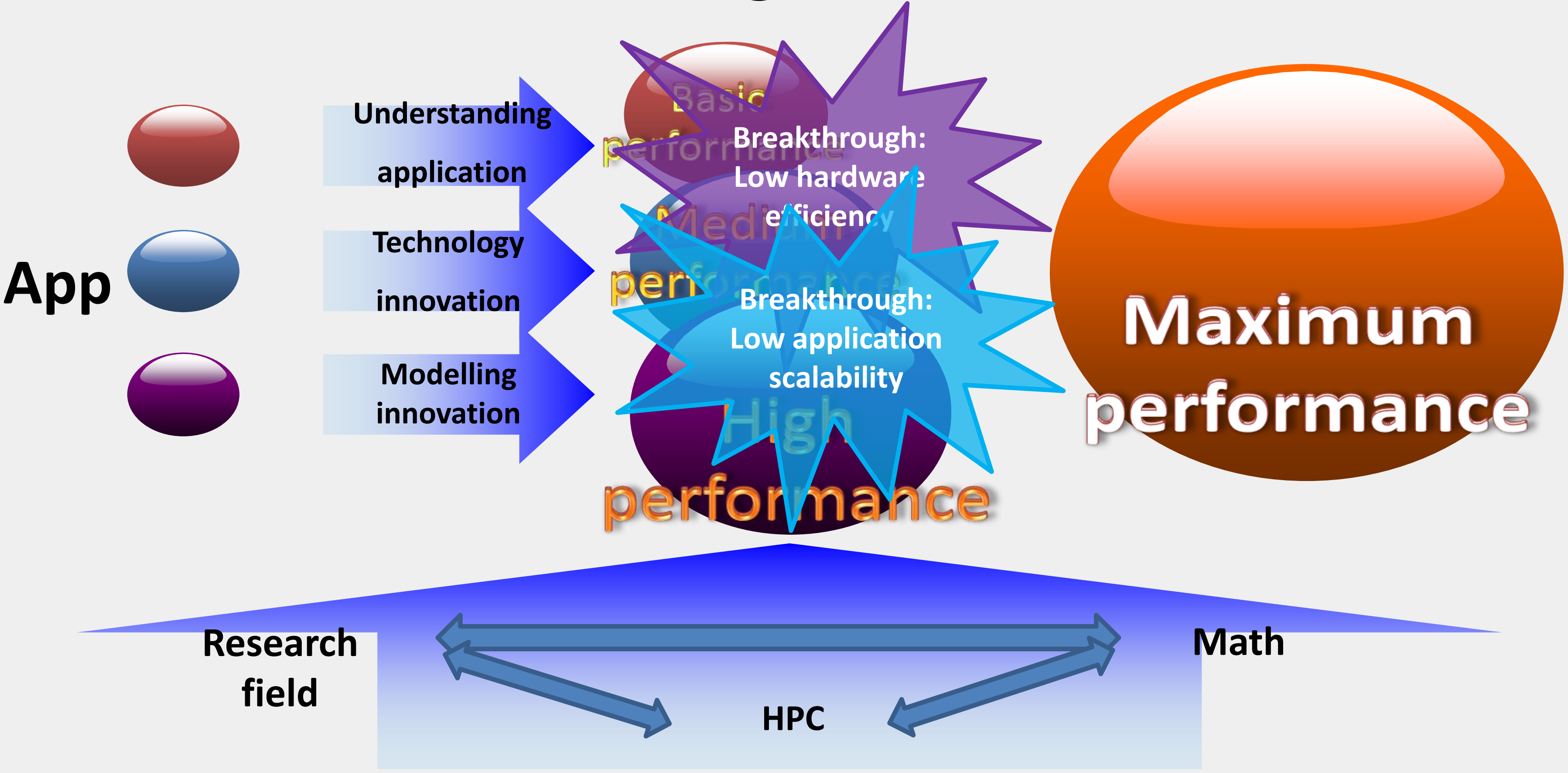
Simulate system~12 electrons



**N=7**

**The optimized coeds can be used to investigate the behavior of spin interaction between electrons and edge states of topological insulators which is significant for the realization of spintronics.**

# Realize the Highest Performance



# Thank You !