An Accidental Benchmarker

Jack Dongarra
University of Tennessee
Oak Ridge National Laboratory
University of Manchester
Over the Past 50 Years Evolving SW and Alg Tracking Hardware Developments

Features: Performance, Portability, and Accuracy

<table>
<thead>
<tr>
<th>EISPACK (1970's)</th>
<th>Rely on</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Translation of Algol to F66)</td>
<td>Fortran, but row oriented</td>
</tr>
</tbody>
</table>

- **EISPACK** is a software library for numerical computation of eigenvalues and eigenvectors of matrices,
  - Written in FORTRAN.
  - Contains subroutines for calculating the eigenvalues of nine classes of matrices:
    - complex general, complex Hermitian, real general, real symmetric, real symmetric banded,
    - real symmetric tridiagonal, special real tridiagonal, generalized real, and
    - generalized real symmetric matrices.
  - The library drew heavily on Algol algorithms developed by Jim Wilkinson & colleagues.
### Over the Past 50 Years Evolving SW and Alg Tracking Hardware Developments

#### Features: Performance, Portability, and Accuracy

<table>
<thead>
<tr>
<th>EISPACK (1970's)</th>
<th>Level 1 Basic Linear Algebra Subprograms (BLAS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Translation of Algol to F66)</td>
<td>Rely on</td>
</tr>
<tr>
<td></td>
<td>- Fortran, but row oriented</td>
</tr>
<tr>
<td></td>
<td>Standards for: Vector-Vector operations</td>
</tr>
</tbody>
</table>

- **EISPACK** is a software library for numerical computation of eigenvalues and eigenvectors of matrices,
  - Written in FORTRAN.
  - Contains subroutines for calculating the eigenvalues of nine classes of matrices:
    - complex general, complex Hermitian, real general, real symmetric, real symmetric banded,
    - real symmetric tridiagonal, special real tridiagonal, generalized real, and
    - generalized real symmetric matrices.
  - The library drew heavily on Algol algorithms developed by Jim Wilkinson & colleagues.
My First Paper

SOFTWARE—PRACTICE AND EXPERIENCE, VOL. 9, 219–226 (1979)

Unrolling Loops in FORTRAN

J. J. DONGARRA AND A. R. HINDS
Argonne National Laboratory, Argonne, Illinois 60439, U.S.A.

SUMMARY
The technique of ‘unrolling’ to improve the performance of short program loops, resorting to assembly language coding is discussed. A comparison of the benefit ‘unrolling’ on a variety of computers using an assortment of FORTRAN compiled presented.

• Reduces loop overhead
  • Level of unrolling dedicated by the instruction stack size
• Help the compiler to:
  • Facilitates pipelining
  • Increases the concurrence between independent functional units
• Provided ~15% performance improvement

TECHNIQUE
When a loop is unrolled, its contents are replicated one or more times, with appropriate adjustments to array indices and the loop increment. For instance, the DAXPY sequence, which adds a multiple of one vector to a second vector:

DO 10 I = 1,N
Y(I) = Y(I) + A * X(I)
10 CONTINUE

would, unrolled to a depth of four, assume the form

M = N - MOD(N,4)
DO 10 I = 1,M
  Y(I) = Y(I) + A * X(I)
  Y(I + 1) = Y(I + 1) + A * X(I + 1)
  Y(I + 2) = Y(I + 2) + A * X(I + 2)
  Y(I + 3) = Y(I + 3) + A * X(I + 3)
10 CONTINUE

Basic Linear Algebra Subprograms for Fortran Usage

C. L. LAWSON
Jet Propulsion Laboratory
R. J. HANSON
Sandia Laboratories
D. R. KINCAID
The University of Texas at Austin
and
F. T. KROGH
Jet Propulsion Laboratory

A package of 28 low level subprograms for many of the basic operations of numerical linear algebra is presented. The package is intended to be used with Fortran. The operations in the package include dot product, elementary vector operation, Givens transformation, vector copy and swap, vector norm,
Over the Past 50 Years Evolving SW and Alg Tracking Hardware Developments

<table>
<thead>
<tr>
<th>Features: Performance, Portability, and Accuracy</th>
</tr>
</thead>
</table>
| **EISPACK (1970's)**  
  (Translation of Algol to F66) | ![Image]  
  Rely on  
  - Fortran, but row oriented |
| **Level 1 Basic Linear Algebra Subprograms (BLAS)** | ![Image]  
  Standards for: Vector-Vector operations |
| **LINPACK (1980's)**  
  (Vector operations) | ![Image]  
  Rely on  
  - Level-1 BLAS operations  
  - Column oriented |
An Accidental Benchmarker

LINPACK was an NSF Project w/ ANL, UNM, UM, & UCSD
We worked independently and came to Argonne in the summers

Top 23 List from 1977

<table>
<thead>
<tr>
<th>Facility</th>
<th>TIME</th>
<th>UNIT</th>
<th>Computer</th>
<th>Type</th>
<th>Compiler</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCAR</td>
<td>1.049</td>
<td>0.14</td>
<td>CRAY-1</td>
<td>S</td>
<td>CFT, Assembly BLAS</td>
</tr>
<tr>
<td>LASL</td>
<td>1.64</td>
<td>0.43</td>
<td>CDC 7600</td>
<td>S</td>
<td>FIN, Assembly BLAS</td>
</tr>
<tr>
<td>NCAR</td>
<td>1.77</td>
<td>0.36</td>
<td>CRAY-1</td>
<td>S</td>
<td>CFT</td>
</tr>
<tr>
<td>LASL</td>
<td>2.20</td>
<td>0.61</td>
<td>CDC 7600</td>
<td>S</td>
<td>FIN</td>
</tr>
<tr>
<td>Argonne</td>
<td>2.30</td>
<td>0.86</td>
<td>IBM 370/195</td>
<td>D</td>
<td>H</td>
</tr>
<tr>
<td>NCAR</td>
<td>3.09</td>
<td>1.05</td>
<td>CDC 7600</td>
<td>S</td>
<td>Local</td>
</tr>
<tr>
<td>Argonne</td>
<td>3.79</td>
<td>1.33</td>
<td>IBM 3033</td>
<td>D</td>
<td>H</td>
</tr>
<tr>
<td>NASA Langley</td>
<td>4.18</td>
<td>1.42</td>
<td>CDC Cyber 175</td>
<td>S</td>
<td>FIN</td>
</tr>
<tr>
<td>LLL</td>
<td>4.50</td>
<td>1.47</td>
<td>CDC Cyber 175</td>
<td>S</td>
<td>FIN</td>
</tr>
<tr>
<td>LLNL</td>
<td>4.69</td>
<td>1.61</td>
<td>CDC 7600</td>
<td>S</td>
<td>CHAT, No optimize</td>
</tr>
<tr>
<td>SLAC</td>
<td>4.79</td>
<td>1.69</td>
<td>IBM 370/168</td>
<td>D</td>
<td>H Ext., Fast mult.</td>
</tr>
<tr>
<td>Michigan</td>
<td>6.31</td>
<td>1.84</td>
<td>Anadahl 470/V6</td>
<td>D</td>
<td>H</td>
</tr>
<tr>
<td>Toronto</td>
<td>7.90</td>
<td>2.59</td>
<td>IBM 370/165</td>
<td>D</td>
<td>H Ext., Fast mult.</td>
</tr>
<tr>
<td>Northwestern</td>
<td>7.14</td>
<td>4.20</td>
<td>CDC 6600</td>
<td>S</td>
<td>FTN</td>
</tr>
<tr>
<td>Texas</td>
<td>8.93</td>
<td>5.63</td>
<td>CDC 6600</td>
<td>S</td>
<td>RGN</td>
</tr>
<tr>
<td>China Lake</td>
<td>8.53</td>
<td>5.69</td>
<td>Univac 1110</td>
<td>S</td>
<td>V</td>
</tr>
<tr>
<td>Yale</td>
<td>9.52</td>
<td>7.53</td>
<td>DEC KL-20</td>
<td>S</td>
<td>P20</td>
</tr>
<tr>
<td>Bell Labs</td>
<td>9.46</td>
<td>10.1</td>
<td>Honeywell 6080</td>
<td>S</td>
<td>Y</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>9.46</td>
<td>10.1</td>
<td>Univac 1110</td>
<td>S</td>
<td>V</td>
</tr>
<tr>
<td>Iowa State</td>
<td>11.34</td>
<td>10.2</td>
<td>Intel AS/5 mod3</td>
<td>D</td>
<td>H</td>
</tr>
<tr>
<td>U. Ili. Chicago</td>
<td>11.8</td>
<td>10.9</td>
<td>IBM 370/158</td>
<td>D</td>
<td>G1</td>
</tr>
</tbody>
</table>

Appendix B of the Linpack Users’ Guide
Designed to help users estimate the run time for solving systems of equation using the Linpack software.
First benchmark report from 1977;
Cray 1 to DEC PDP-10
The Original Code

Benchmark based on solving $Ax=b$ using LU factorization

Using Level 1 BLAS

Fortran 77

```
subroutine dgefa(a,lda,n,ipvt,info)
  integer lda,n,ipvt(l),info
  double precision a(lda,l)
  double precision t
  integer idamax,j,k,kp1,l,nm1
  info = 0
  nm1 = n - 1
  if (nm1 .lt. 1) go to 70
  do 60 k = 1, nm1
    kp1 = k + 1
    l = idamax(n-k+1,a(k,k),1) + k - 1
    ipvt(k) = l
    if (a(l,k) .eq. 0.0d0) go to 40
    if (l .eq. k) go to 10
    t = a(l,k)
    a(l,k) = a(k,k)
    a(k,k) = t
  10 continue
    t = -1.0d0/a(k,k)
    call dscal(n-k,t,a(k+1,k),l)
    do 30 j = kp1, n
      t = a(l,j)
      if (l .eq. k) go to 20
      a(l,j) = a(k,j)
      a(k,j) = t
    20 continue
      call daxpy(n-k,t,a(k+1,k),l,a(k+1,j),l)
    30 continue
  go to 50
  40 continue
    info = k
  50 continue
  60 continue
  70 continue
  ipvt(n) = n
  if (a(n,n) .eq. 0.0d0) info = n
return
end
```

# of Flops

$$
\sum_{k=1}^{n-1} \sum_{k+1}^{n} \sum_{k+1}^{n} 2 = \frac{2}{3} n^3 + O(n^2)
$$

- **Outer loop**
- **Partial Pivoting**
  - Find the largest element of a column & interchange
- **Middle loop**
- **Scale column**
- **Apply pivot**
- **Inner Loop**
  - $y = \alpha x + y$
$Ax=b$; factor $A$, then solve

Factor $A$ into $L$ and $U$
Then solve the system of equations.
$y = L^{-1} b; x = U^{-1} y$

About $\frac{2}{3} n^3 + O(n^2)$ floating point ops and $\frac{2}{3} n^3 + O(n^2)$ touches in the factorization and $O(n^2)$ for the solves.

Rate of execution measured as:
# of Operations/Time or Floating point operations per second (Flops)

Pivoting needed to
- Avoid dividing by zero or a small number
  - Degeneracy
- Avoid growth in the elements
  - Loss of accuracy
Linpack Benchmark Characteristics

- Portable, runs on any system

- Easy to understand
  One number, rate of execution (Flops/second)

- Algorithm fixed

- Allows for restructuring/optimizing algorithm

- All performance data with the same arithmetic precision, 64-bit floating point.

- Benchmark checks if “correct solution” achieved:
  \[
  \frac{||Ax-b||}{(||A|| ||x||+||b||)}
  \]

- Not intended to measure entire machine performance.

- In the benchmark report, “One further note: The following performance data should not be taken too seriously.”
LU Decomposition Algorithm Has Evolved

Instead of working with a columns of the matrix we work with panels, a collection of columns.

The algorithm proceeds by doing an LU factorization of the panel B.

Then updating $C$, $(U_1 = L_0^{-1}C)$, and then applying all the transformation to the rest of the matrix, $E' = E - L_1U_1$

This results in a rank-k update of $E$ or a matrix multiply (GEMM).

($GEMM: O(n^3)$ ops & $O(n^2)$ references)

Same number of operations and same numerical properties.
Top500 Since 1993

- Since 1978 I maintained a LINPACK Benchmark list.
- Hans Meuer and Erich Strohmaier had a list of fastest computers ranked by peak performance.
- Listing of the 500 most powerful computers in the World.
- Yardstick: Performance for $Ax=b$, dense problem

Maintained and updated twice a year:
  SC‘xy in the States in November
  Meeting in Germany in June
Top500/HPL Benchmark Timeline

- 1974 LINPACK library (vector computers)
- 1977 LINPACK 100 Benchmark (Fortran only)
- 1986 LINPACK 1000 Benchmark (assembly allowed)
- **1988 First Gflop/s system, NEC SX-2**
- 1991 LINPACK Table 3 (HPL rules defined)

- **1993 Top500 starts (LANL CM-5 #1, 60 Gflop/s)**
- 1994 All Top500 systems over 1 Gflop/s
- **1997 First Tflop/s system ASCI Red @ SNL**
- 2001 China has its first computer on Top500
- 2002 #1 Earth Simulator @ JAMSTEC 5 X faster
- 2005 All Top500 systems over 1 Tflop/s

- **2008 First Pflop/s system Roadrunner @ LANL**
- 2010 Tianhe-1A @ Tianjin (2 Pflop/s) 1st time China #1
- 2011 K computer @ RIKEN (8 Pflop/s)
- 2012 Sequoia @ LLNL (1.5 M cores)
- 2014 Tianhe-2 @ NUDT (33 Pflop/s)
- 2016 China and US have = number of systems
- 2016 TaihuLight @ Wuxi (93 Pflop/s, >10 M cores)
- 2018 Summit @ ORNL (122 Pflop/s)
- **2019 All TOP500 systems over 1Pflop/s**
- 2020 Fugaku @ RIKEN (442 Pflop/s)
- 2021 Of the Top500: China=186 & US=123
- **2021 First Exascale System?**
#1 Systems on the Top500 Over the Past 27 Years

<table>
<thead>
<tr>
<th>Top500 List (# of times)</th>
<th>Computer</th>
<th>HPL $r_{\text{max}}$ (Tflop/s)</th>
<th>Procs/Cores</th>
<th>Matrix Size</th>
<th>Hours To BM</th>
<th>MW</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/93 (1)</td>
<td>TMC CM-5/1024 (DOE LANL)</td>
<td>.060</td>
<td>1,024</td>
<td>52,224</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>11/93 (1)</td>
<td>Fujitsu Numerical Wind Tunnel (Nat. Aerospace Lab of Japan)</td>
<td>.124</td>
<td>140</td>
<td>31,920</td>
<td>0.1</td>
<td>1</td>
</tr>
<tr>
<td>6/94 (1)</td>
<td>Intel XP/S140 (DOE SNL)</td>
<td>.143</td>
<td>3,680</td>
<td>55,700</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>11/94-11/95 (3)</td>
<td>Fujitsu Numerical Wind Tunnel (Nat. Aerospace Lab of Japan)</td>
<td>.170</td>
<td>140</td>
<td>42,000</td>
<td>0.1</td>
<td>1</td>
</tr>
<tr>
<td>6/96 (1)</td>
<td>Hitachi SR2201/1024 (Univ. of Tokyo)</td>
<td>.220</td>
<td>1,024</td>
<td>138,240</td>
<td>2.2</td>
<td></td>
</tr>
<tr>
<td>11/96 (1)</td>
<td>Hitachi CP-PACS/2048 (Univ of Tsukuba)</td>
<td>.368</td>
<td>2,048</td>
<td>103,680</td>
<td>.6</td>
<td></td>
</tr>
<tr>
<td>6/97-6/00 (7)</td>
<td>Intel ASCI Red (DOE SNL)</td>
<td>2.38</td>
<td>9,632</td>
<td>362,880</td>
<td>3.7</td>
<td>.85</td>
</tr>
<tr>
<td>11/00-11/01 (3)</td>
<td>IBM ASCI White, SP Power3 375 MHz (DOE LLNL)</td>
<td>7.23</td>
<td>8,192</td>
<td>518,096</td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>6/02-6/04 (5)</td>
<td>NEC Earth-Simulator (JAMSTEC)</td>
<td>35.9</td>
<td>5,120</td>
<td>1,000,000</td>
<td>5.2</td>
<td>6.4</td>
</tr>
<tr>
<td>11/04-11/07 (7)</td>
<td>IBM BlueGene/L (DOE LLNL)</td>
<td>478</td>
<td>212,992</td>
<td>1,000,000</td>
<td>0.4</td>
<td>1.4</td>
</tr>
<tr>
<td>6/08-6/09 (3)</td>
<td>IBM Roadrunner –PowerXCell 8i 3.2 Ghz (DOE LANL)</td>
<td>1,105</td>
<td>129,600</td>
<td>2,329,599</td>
<td>2.1</td>
<td>2.3</td>
</tr>
<tr>
<td>11/09–6/10 (2)</td>
<td>Cray Jaguar - XT5-HE 2.6 GHz (DOE ORNL)</td>
<td>1,759</td>
<td>224,162</td>
<td>5,474,272</td>
<td>17</td>
<td>6.9</td>
</tr>
<tr>
<td>11/10 (1)</td>
<td>NUDT Tianhe-1A, X5670 2.93Ghz NVIDIA (NSC Tianjin)</td>
<td>2,566</td>
<td>186,368</td>
<td>3,600,000</td>
<td>3.4</td>
<td>4.0</td>
</tr>
<tr>
<td>6/11–11/11 (2)</td>
<td>Fujitsu K computer, SPARC64 VIIIfx (RIKEN)</td>
<td>10,510</td>
<td>705,024</td>
<td>11,870,208</td>
<td>29</td>
<td>9.9</td>
</tr>
<tr>
<td>5/12 (1)</td>
<td>IBM Sequoia BlueGene/Q (DOE LLNL)</td>
<td>16,324</td>
<td>1,572,864</td>
<td>12,645,072</td>
<td>23</td>
<td>7.9</td>
</tr>
<tr>
<td>11/12 (1)</td>
<td>Cray XK7 Titan AMD + NVIDIA Kepler (DOE ORNL)</td>
<td>17,590</td>
<td>560,640</td>
<td>4,423,680</td>
<td>0.9</td>
<td>8.2</td>
</tr>
<tr>
<td>6/13–11/15 (6)</td>
<td>NUDT Tianhe-2 Intel IvyBridge + Xeon Phi (NCSS Guangzhou)</td>
<td>33,862</td>
<td>3,120,000</td>
<td>9,960,000</td>
<td>5.4</td>
<td>17.8</td>
</tr>
<tr>
<td>6/16–11/17 (4)</td>
<td>Sunway Taihulight System (NSCC Wuxi)</td>
<td>93,014</td>
<td>10,549,600</td>
<td>12,288,000</td>
<td>3.7</td>
<td>15.4</td>
</tr>
<tr>
<td>6/18–11/19 (4)</td>
<td>IBM Summit Power9 + Nvidia Volta (DOE ORNL)</td>
<td>148,600</td>
<td>2,414,592</td>
<td>16,473,600</td>
<td>3.3</td>
<td>10.1</td>
</tr>
<tr>
<td>6/20–?</td>
<td>Fujitsu Fugaku ARM A64FX (RIKEN)</td>
<td>442,010</td>
<td>7,630,828</td>
<td>21,288,960</td>
<td>4.4</td>
<td>29.9</td>
</tr>
</tbody>
</table>
State of Supercomputing in 2021

- Pflops (> $10^{15}$ Flop/s) computing fully established with all 500 systems.
- Three technology architecture possibilities or “swim lanes” are thriving.
  - Commodity (e.g. Intel)
  - Commodity + accelerator (e.g. GPUs) (145 systems; 138 NVIDIA, 3 Intel Phi + 4)
  - Lightweight cores (e.g. IBM BG, TaihuLight)
- China: Top consumer and producer overall.
- Interest in supercomputing is now worldwide, and growing in many new markets (~50% of Top500 computers are in industry).
- Intel processors largest share, 86% followed by AMD, 10%.
- Exascale ($10^{18}$ Flop/s) projects exist in many countries and regions.
PERFORMANCE DEVELOPMENT OF HPC OVER THE LAST 28 YEARS FROM THE TOP500

- Thinking Machine CM-5 with 1024 processors at Los Alamos Nat Lab used for nuclear weapons design.

My Laptop: 166 Gflop/s

Less incentives to upgrade systems leads to older systems.
PERFORMANCE DEVELOPMENT

- Achieved: 1994
- Achieved: 1996
- Achieved: 1998
- Achieved: 2000
- Achieved: 2002
- Achieved: 2004
- Achieved: 2006
- Achieved: 2008
- Achieved: 2010
- Achieved: 2012
- Achieved: 2014
- Achieved: 2016
- Achieved: 2018
- Achieved: 2020
- Achieved: 2021

- ASCI Red
- Los Alamos NL

- RoadRunner

- China says 2021
- U.S. says 2021-2022
### June 2021: The TOP 10 Systems (42% of the Total Performance of Top500)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Site</th>
<th>Computer</th>
<th>Country</th>
<th>Cores</th>
<th>Rmax [Pflops]</th>
<th>% of Peak</th>
<th>Power [MW]</th>
<th>GFlops/Watt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RIKEN Center for Computational Science</td>
<td>Fugaku, ARM A64FX (48C, 2.2 GHz), Tofu D Interconnect</td>
<td>Japan</td>
<td>7,299,072</td>
<td>442.</td>
<td>82</td>
<td>29.9</td>
<td>14.8</td>
</tr>
<tr>
<td>2</td>
<td>DOE / OS Oak Ridge Nat Lab</td>
<td>Summit, IBM Power 9 (22C, 3.0 GHz), <strong>NVIDIA GV100 (80C)</strong>, Mellanox EDR</td>
<td>USA</td>
<td>2,397,824</td>
<td>149.</td>
<td>74</td>
<td>10.1</td>
<td>14.7</td>
</tr>
<tr>
<td>3</td>
<td>DOE / NNSA L Livermore Nat Lab</td>
<td>Sierra, IBM Power 9 (22C, 3.1 GHz), <strong>NVIDIA GV100 (80C)</strong>, Mellanox EDR</td>
<td>USA</td>
<td>1,572,480</td>
<td>94.6</td>
<td>75</td>
<td>7.44</td>
<td>12.7</td>
</tr>
<tr>
<td>4</td>
<td>National Super Computer Center in Wuxi</td>
<td>Sunway TaihuLight, <strong>SW26010 (260C)</strong> + Custom</td>
<td>China</td>
<td>10,649,000</td>
<td>93.0</td>
<td>74</td>
<td>15.4</td>
<td>6.05</td>
</tr>
<tr>
<td>5</td>
<td>DOE / OS NERSC - LBNL</td>
<td>Perlmutter HPE Cray EX235n, <strong>AMD EPYC 64C 2.45GHz</strong>, <strong>NVIDIA A100</strong>, Slingshot-10</td>
<td>USA</td>
<td>706,304</td>
<td>64.6</td>
<td>69</td>
<td>2.53</td>
<td>25.5</td>
</tr>
<tr>
<td>6</td>
<td>NVIDIA Corporation</td>
<td>Selene NVIDIA DGX A100, <strong>AMD EPYC 7742 (64C, 2.25GHz)</strong>, <strong>NVIDIA A100 (108C)</strong>, Mellanox HDR Infiniband</td>
<td>USA</td>
<td>555,520</td>
<td>63.4</td>
<td>80</td>
<td>2.64</td>
<td>23.9</td>
</tr>
<tr>
<td>7</td>
<td>National Super Computer Center in Guangzhou</td>
<td>Tianhe-2A NUDT, Xeon (12C) + <strong>MATRIX-2000 (128C)</strong> + Custom</td>
<td>China</td>
<td>4,981,760</td>
<td>61.4</td>
<td>61</td>
<td>18.5</td>
<td>3.32</td>
</tr>
<tr>
<td>8</td>
<td>JUWELS Booster Module</td>
<td>Bull Sequana XH2000, <strong>AMD EPYC 7402 (24C, 2.8GHz)</strong>, <strong>NVIDIA A100 (108C)</strong>, Mellanox HDR InfiniBand/ParTec ParaStation ClusterSuite</td>
<td>Germany</td>
<td>448,280</td>
<td>44.1</td>
<td>62</td>
<td>1.76</td>
<td>25.0</td>
</tr>
<tr>
<td>9</td>
<td>Eni S.p.A in Italy</td>
<td>HPC5, Dell EMC PowerEdge C4140, Xeon (24C, 2.1 GHz) + <strong>NVIDIA V100 (80C)</strong>, Mellanox HDR</td>
<td>Italy</td>
<td>669,760</td>
<td>35.5</td>
<td>69</td>
<td>2.25</td>
<td>15.8</td>
</tr>
<tr>
<td>10</td>
<td>Texas Advanced Computing Center / U of Texas</td>
<td>Frontera, Dell C6420, Xeon Platinum, 8280 (28C, 2.7 GHz), Mellanox HDR</td>
<td>USA</td>
<td>448,448</td>
<td>23.5</td>
<td>61</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Countries Share

Chart Title

<table>
<thead>
<tr>
<th>Country</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>China</td>
<td>186</td>
</tr>
<tr>
<td>Japan</td>
<td>123</td>
</tr>
<tr>
<td>Germany</td>
<td>35</td>
</tr>
<tr>
<td>UK</td>
<td>16</td>
</tr>
<tr>
<td>Russia</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>
Plot of Top500 from June 2021
# 1 Fugaku’s Fujitsu A64fx Processor is…

- A Many-Core ARM CPU...
  - 48 compute cores + 2 or 4 assistant (OS) cores
  - New core design
  - Near Xeon-Class Integer performance core
  - ARM V8 --- 64bit ARM ecosystem
  - Interconnect Tofu-D
  - 3.4 TFLOP/s Peak 64-bit performance

...but also an accelerated GPU-like processor

- SVE 512 bit x 2 vector extensions (ARM & Fujitsu)
  - Integer (1, 2, 4, 8 bytes) + Float (16, 32, 64 bytes)
- Cache + memory localization (sector cache)
- HBM2 on package memory - Massive Mem BW (Bytes/DPF ~0.4)
  - Streaming memory access, strided access, scatter/gather etc.
- Intra-chip barrier synch. and other memory enhancing features

Fugaku Total System Config & Performance

- **Total # Nodes:** 158,976 nodes (1 CPU/node)
  - 384 nodes/rack x 396 (full) racks = 152,064 nodes and
  - 192 nodes/rack x 36 (half) racks = 6,912 nodes

- **Theoretical Peak Compute Performances**
  - **Normal Mode** (CPU Frequency 2GHz)
    - **64 bit** Double Precision FP: 488 Petaflops
    - **32 bit** Single Precision FP: 977 Petaflops
    - **16 bit** Half Precision FP (AI training): 1.95 Exaflops
    - **8 bit Integer** (AI Inference): 3.90 Exaops
  - **Theoretical Peak Memory BW:** 163 Petabytes/s

Fugaku represents 16% of all the other Top500 systems.

### System Performance
- Peak performance of 200 Pflop/s for modeling & simulation
- Peak performance of 3.3 Eflop/s for 16 bit floating point used in for data analytics, ML, and artificial intelligence

### Each node has
- 2 IBM POWER9 processors
  - Each w/22 cores
  - 2.3% performance of system
- 6 NVIDIA Tesla V100 GPUs
  - Each w/80 SMs
  - 97.7% performance of system
- 608 GB of fast memory
- 1.6 TB of NVMe memory

### The system includes
- 4608 nodes
  - 27,648 GPUs
  - Street value $10K each
- Dual-rail Mellanox EDR InfiniBand network
- 250 PB IBM Spectrum Scale file system transferring data at 2.5 TB/s
HPCG Results; The Other Benchmark

• High Performance Conjugate Gradients (HPCG).
• Solves $Ax=b$, $A$ large, sparse, $b$ known, $x$ computed.
• An optimized implementation of PCG contains essential computational and communication patterns that are prevalent in a variety of methods for discretization and numerical solution of PDEs

• Patterns:
  • Dense and sparse computations.
  • Dense and sparse collectives.
  • Multi-scale execution of kernels via MG (truncated) V cycle.
  • Data-driven parallelism (unstructured sparse triangular solves).
• Strong verification (via spectral properties of PCG).
HPCG Details

3D Laplacian discretization

\[ L[u] \equiv \nabla^2 u = f \]

Preconditioned Conjugate Gradient solver

\[
\begin{align*}
p_0 & \leftarrow x_0, \quad r_0 \leftarrow b - Ap_0 \\
\text{for } i = 1, 2, \text{ to max_iterations } & \text{ do} \\
z_i & \leftarrow M^{-1}r_{i-1} \\
\text{if } i = 1 & \text{ then} \quad \text{Multigrid and Gauss-Seidel} \\
p_i & \leftarrow z_i \\
\alpha_i & \leftarrow \text{dot_prod}(r_{i-1}, z_i) \\
\text{else} & \\
\alpha_i & \leftarrow \text{dot_prod}(r_{i-1}, z_i) \\
\beta_i & \leftarrow \alpha_i / \alpha_{i-1} \\
p_i & \leftarrow \beta_i p_{i-1} + z_i \\
\text{end if} \\
\alpha_i & \leftarrow \text{dot_prod}(r_{i-1}, z_i) / \text{dot_prod}(p_i, Ap_i) \\
x_{i+1} & \leftarrow x_i + \alpha_i p_i \\
r_i & \leftarrow r_{i-1} - \alpha_i Ap_i \\
\text{if } \|r_i\|_2 < \text{tolerance} & \text{ then} \\
\text{STOP} \\
\text{end if} \\
\text{end for}
\]
## HPCG Top10, June 2021

<table>
<thead>
<tr>
<th>Rank</th>
<th>Site</th>
<th>Computer</th>
<th>Cores</th>
<th>HPL Rmax (Pflop/s)</th>
<th>TOP500 Rank</th>
<th>HPCG (Pflop/s)</th>
<th>Fraction of Peak</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RIKEN Center for Computational Science, Japan</td>
<td>Fugaku, Fujitsu A64FX 48C 2.2GHz, Tofu D, Fujitsu</td>
<td>7,630,848</td>
<td>442.0</td>
<td>1</td>
<td>16.0</td>
<td>3.0%</td>
</tr>
<tr>
<td>2</td>
<td>DOE/SC/ORNL USA</td>
<td>Summit, AC922, IBM POWER9 22C 3.7GHz, Dual-rail Mellanox FDR, NVIDIA Volta V100, IBM</td>
<td>2,414,592</td>
<td>148.6</td>
<td>2</td>
<td>2.93</td>
<td>1.5%</td>
</tr>
<tr>
<td>3</td>
<td>DOE/SC/LBNL USA</td>
<td>Perlmutter, HPE Cray EX235n, AMD EPYC 7763 64C 2.45GHz, NVIDIA A100 SXM4 40 GB, Slingshot-10</td>
<td>761,856</td>
<td>64.6</td>
<td>5</td>
<td>1.91</td>
<td>2.0%</td>
</tr>
<tr>
<td>4</td>
<td>DOE/NNSA/LLNL USA</td>
<td>Sierra, S922LC, IBM POWER9 20C 3.1 GHz, Mellanox EDR, NVIDIA Volta V100, IBM</td>
<td>1,572,480</td>
<td>94.6</td>
<td>3</td>
<td>1.80</td>
<td>1.4%</td>
</tr>
<tr>
<td>5</td>
<td>NVIDIA USA</td>
<td>Selene, DGX SuperPOD, AMD EPYC 7742 64C 2.25 GHz, Mellanox HDR, NVIDIA Ampere A100</td>
<td>555,520</td>
<td>63.5</td>
<td>6</td>
<td>1.62</td>
<td>2.0%</td>
</tr>
<tr>
<td>6</td>
<td>Forschungszentrum Juelich (FZJ), Germany</td>
<td>JUWELS Booster Module, Bull Sequana XH2000, AMD EPYC 7402 24C 2.8GHz, Mellanox HDR InfiniBand, NVIDIA Ampere A100</td>
<td>449,280</td>
<td>44.1</td>
<td>8</td>
<td>1.28</td>
<td>1.8%</td>
</tr>
<tr>
<td>7</td>
<td>Saudi Aramco, Saudi Arabia</td>
<td>Dammam-7, Cray CS-Storm, Xeon Gold 6248 20C 2.5GHz, InfiniBand HDR 100, NVIDIA Volta V100, HPE</td>
<td>672,520</td>
<td>22.4</td>
<td>11</td>
<td>0.88</td>
<td>1.6%</td>
</tr>
<tr>
<td>8</td>
<td>Eni S.p.A., Italy</td>
<td>HPCS, PowerEdge, C4140, Xeon Gold 6252 24C 2.1 GHz, Mellanox HDR, NVIDIA Volta V100, Dell</td>
<td>669,760</td>
<td>35.5</td>
<td>9</td>
<td>0.86</td>
<td>1.7%</td>
</tr>
<tr>
<td>9</td>
<td>Information Technology Center, The University of Tokyo, Japan</td>
<td>Wisteria/BDEC-01 (Odysse), PRIMEHPC FX1000, A64FX 48C 2.2GHz, Tofu D</td>
<td>368,640</td>
<td>22.1</td>
<td>13</td>
<td>0.82</td>
<td>3.2%</td>
</tr>
<tr>
<td>10</td>
<td>Japan Agency for Marine-Earth Science and Technology</td>
<td>Earth Simulator -SX-Aurora TSUBASA, A412-8, Vector Engine Type20B 8C 1.6GHz, Infiniband HDR200</td>
<td>43,776</td>
<td>0.01</td>
<td>41</td>
<td>0.75</td>
<td>5.6%</td>
</tr>
</tbody>
</table>
Comparison between Peak and HPL for June 2021
Comparison between Peak, HPL, and HPCG for June 2021

![Graph showing comparison between Peak, HPL, and HPCG for June 2021.](image)

- **Tflop/s**
- **Rpeak**
- **Rmax**
- **HPCG**
Modern Hardware: Lower Precision for Deep Learning

- Hardware (company)
  - GPU Tensor Cores (NVIDIA)
  - TPU MXU (Google)
  - Zion (Facebook)
  - DaVinci (Huawei)
  - Dot-product engine (HPE)
  - Eyeriss (Amazon)
  - Wafer Scale Engine (Cerebras)
  - Nervana (Intel)
  - Deep Learning Boost (Intel AI)
  - Graph Core
  - ...

- Lower-precision benchmarks
  - Baidu
  - Dawn
  - mlperf
  - Deep500
  - ...
  - HPL-AI
  - 60+
WHY MIXED PRECISION? (Less is Faster)

- There are many reasons to consider mixed precision in our algorithms...
  - Less Communication
    - Reduce memory traffic
    - Reduce network traffic
  - Reduce memory footprint
  - More Flop per second
    - Reduced energy consumption
    - Reduced time to compute
  - Accelerated hardware in current architecture.
  - Suitable numerical properties for some algorithms & problems.

### Mixed Precision: Hardware Motivation

<table>
<thead>
<tr>
<th></th>
<th>IBM Cell Broadband Engine</th>
<th>Apple ARM Cortex-A9</th>
<th>NVIDIA Kepler K10, K20, K40, K80</th>
<th>NVIDIA Volta/Turing</th>
<th>NVIDIA Volta/Turing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>14x</td>
<td>7x</td>
<td>3x</td>
<td>2x</td>
<td>16x</td>
</tr>
<tr>
<td></td>
<td>32 bits / 64 bits</td>
<td>32 bits / 64 bits</td>
<td>32 bits / 64 bits</td>
<td>32 bits / 64 bits</td>
<td>16 bits / 64 bits</td>
</tr>
</tbody>
</table>
HPL-AI Benchmark Utilizing 16-bit Arithmetic

1. Generate random linear system $Ax=b$
2. Represent the matrix $A$ in low precision (16-bit floating point)
3. Factor $A$ in lower precision into $LU$ by Gaussian elimination
4. Compute approximate solution with $LU$ factors in low precision
5. Perform up to 50 iterations of refinement, e.g., GMRES to get accuracy up to 64-bit floating point
6. Use $LU$ factors for preconditioning
7. Validate the answer is correct: scaled residual small
   \[
   \frac{||Ax - b||}{||A||||x|| + ||b||} \times \frac{1}{n\epsilon} \leq O(10)
   \]
8. Compute performance rate as
   \[
   \frac{2}{3} \times \frac{n^3}{\text{time}}
   \]

Iterative refinement for dense systems, $Ax = b$, can work this way.
$L \ U = \text{lu}(A)$
$x = U \backslash (L \backslash b)$
GMRes preconditioned by the LU to solve $Ax=b$

L U = lu$(A)$
$x = U \backslash (L \backslash b)$

lower precision $\text{O}(n^3)$
lower precision $\text{O}(n^2)$
FP64 precision $\text{O}(n^2)$
## HPL-AI Top 10 for June 2021

<table>
<thead>
<tr>
<th>Rank</th>
<th>Site</th>
<th>Computer</th>
<th>Cores</th>
<th>HPL Rmax (Eflop/s)</th>
<th>TOP500 Rank</th>
<th>HPL-AI (Eflop/s)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RIKEN Center for Computational Science, Japan</td>
<td><strong>Fugaku</strong>, Fujitsu A64FX, Tofu D</td>
<td>7,630,848</td>
<td>0.442</td>
<td>1</td>
<td>2.0</td>
<td>4.5</td>
</tr>
<tr>
<td>2</td>
<td>DOE/SC/ORNl USA</td>
<td><strong>Summit</strong>, AC922 IBM POWER9, IB Dual-rail FDR, NVIDIA V100</td>
<td>2,414,592</td>
<td>0.149</td>
<td>2</td>
<td>1.15</td>
<td>7.7</td>
</tr>
<tr>
<td>3</td>
<td>NVIDIA USA</td>
<td><strong>Selene</strong>, DGX SuperPOD, AMD EPYC 7742 64C 2.25 GHz, Mellanox HDR, NVIDIA A100</td>
<td>555,520</td>
<td>0.063</td>
<td>6</td>
<td>0.63</td>
<td>9.9</td>
</tr>
<tr>
<td>4</td>
<td>DOE/SC/LBNL/NERSC USA</td>
<td><strong>Perlmutter</strong>, HPE Cray EX235n, AMD EPYC 7763 64C 2.45 GHz, Slingshot-10, NVIDIA A100</td>
<td>761,856</td>
<td>0.065</td>
<td>5</td>
<td>0.59</td>
<td>9.1</td>
</tr>
<tr>
<td>5</td>
<td>Forschungszentrum Julich (FZJ) Germany</td>
<td><strong>JUWELS Booster Module</strong>, Bull Sequana XH2000, AMD EPYC 7402 24C 2.8GHz, Mellanox HDR InfiniBand, NVIDIA A100, Atos</td>
<td>449,280</td>
<td>0.044</td>
<td>8</td>
<td>0.47</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>University of Florida USA</td>
<td><strong>HiPerGator</strong>, NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Infiniband HDR</td>
<td>138,880</td>
<td>0.017</td>
<td>23</td>
<td>0.17</td>
<td>9.9</td>
</tr>
<tr>
<td>7</td>
<td>Information Technology Center, The University of Tokyo, Japan</td>
<td><strong>Wisteria/BDEC-01 (Odyssey)</strong>, PRIMEHPC FX1000, A64FX 48C 2.2GHz, Tofu D, Fujitsu</td>
<td>368,640</td>
<td>0.022</td>
<td>13</td>
<td>0.10</td>
<td>4.5</td>
</tr>
<tr>
<td>8</td>
<td>National Supercomputer Centre (NSC), Sweden</td>
<td><strong>Berzelius</strong>, NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, A100, Infiniband HDR, Atos</td>
<td>59,520</td>
<td>0.005</td>
<td>84</td>
<td>0.05</td>
<td>9.9</td>
</tr>
<tr>
<td>9</td>
<td>Information Technology Center, Nagoya University, Japan</td>
<td><strong>Flow Type II subsystem</strong>, PRIMERGY CX2570 M5, Xeon Gold 6230 20C 2.1GHz, NVIDIA Tesla V100 SXM2, Infiniband EDR</td>
<td>79,560</td>
<td>0.0049</td>
<td>87</td>
<td>0.03</td>
<td>4.3</td>
</tr>
<tr>
<td>10</td>
<td>#CloudMTS Russia</td>
<td><strong>MTS GROM</strong>, NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, A100 40GB, Infiniband</td>
<td>19,840</td>
<td>0.0023</td>
<td>245</td>
<td>0.015</td>
<td>7</td>
</tr>
</tbody>
</table>
Comparison between HPL-AI, Peak, HPL, and HPCG for June 2021
The Take Away

• HPC Hardware is Constantly Changing
  • Scalar
  • Vector
  • Distributed
  • Accelerated
  • Mixed precision

• Three computer revolutions
  • High performance computing
  • Deep learning
  • Edge & AI

• Algorithm / Software advances follows hardware.
  • And there is “plenty of room at the top”