Serial and Parallel Performance

CodeSourcery, LLC
September 23, 2003
Design Path

Performance → Portability → Productivity → Parallelism

VSIPL
Specification Status

- **Serial Specification**
  - 216-page draft.
  - Under review by VSIPL Forum.

- **Parallel Specification**
  - 24-page preliminary draft.
  - Initial conceptual review complete.
Serial Performance

• Uses VSIPL reference implementation.
  – Not the fastest implementation…
  – … but the relative performance is important.

• Environment:
  – 2GHz Pentium-M
  – 512KB cache, 512MB RAM
  – GNU/Linux, G++ 3.4
Matrix/Vector

\[ \mathbf{v} \mathbin{+}= \mathbf{m} \mathbf{v} \]

![Graph showing vector length vs. time with two lines representing VSIPL and VSIPL++](image)

www.codesourcery.com
Matrix/Matrix

\[ \text{result} += \tan(\sin(m) + \cos(m)) \]

![Graph showing time vs. matrix size for VSIPL and VSIPL++ performance]
Checked Vector Access

![Graph showing the performance of VSIPL and VSIPL++ for checked vector access. The x-axis represents vector length in powers of 10, and the y-axis represents time. The graph compares VSIPL and VSIPL++ across different vector lengths.](image)
Performance Conclusions

• VSIPL++ has approximately zero overhead.
  – Memory effects actually enable VSIPL++ to outperform VSIPL.
  – Expression-template techniques may also improve performance.

• Exceptions are expensive.
  – We are not sure if this overhead can be eliminated.

• Reference implementation will be directly useful.
  – Vendor-optimized versions will probably be better.
Parallelism

• Target systems:
  – Support 1-64K+ processors.
  – Support MPI, POSIX threads.

• Conceptual model:
  – Single-program multiple-data model.
  – Owner computes.
  – Parallelism requires changing only declarations, not expressions.
Parallel VSIPL++ Model

- view0
- view1
- view2
- view3

- block0
- block1
- block2

- data distribution

- grid function

- map

- user program

- processors

- hardware

www.codesourcery.com
Using Parallelism

• Declaration:

\[
\text{Vector}\langle\text{double},
\begin{array}{l}
\text{Dense}\langle 1, \text{double},
\end{array}
\begin{array}{l}
\text{Map}\langle\text{Block}\rangle
\end{array}
\rangle
\]

\[
\text{v} \ (17, \ 1.0, \ \text{Block}(4))
\]

• Meaning:
  – 17: Vector length.
  – 1.0: Initial value.
  – \text{Block}(4): Block distribution over 4 processors.
FYO4 Objectives

• Specification:
  – Finalize serial and parallel specifications.
  – Get approval from VSIPL Forum.

• Implementation:
  – Finish serial implementation.
  – Draft parallel implementation.

• Measurement:
  – Performance analysis.
Contact Information

• Mark Mitchell
  mark@codesourcery.com

• Jeffrey Oldham
  oldham@codesourcery.com

• Nathan Sidwell
  nathan@codesourcery.com
Serial and Parallel Performance

CodeSourcery, LLC
September 23, 2003